The
University
Of
Sheffield.

Data Mining For Lead Optimisation

by

Jessica Emily Stacey

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

This work was sponsored by

gsk

And

EPSRC

The University of Sheffield

Faculty of Social Sciences

Information School

September 2021

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Val Gillet from the University of Sheffield for your support and guidance over the past four years. Also, for reading the many iterations of my chapter drafts, but we have finally done it! Another special thanks must go to my Industrial supervisor, Dr Stephen Pickett from GlaxoSmithKline (GSK) for providing your vast amount of knowledge in our discussions and support over the course of my PhD and especially whilst on my virtual placement. I would also like to acknowledge all the members of the Molecular Design Group at GSK who have listened to several presentations and provided insightful feedback on my work, especially Dr Baptiste Canault.

I would also like to express my thanks to all members of the Sheffield Chemoinformatics group, past and present, for all the stimulating conversations that have been had in meetings and around the whiteboard. Particular thanks to Professor Peter Willet, Dr Antonio de la Vega de Leon, Dr James Wallace, Dr Matthew Seddon, Dr Christina Founti, Dr Gianmarco Ghiandoni, James Webster, Mortiz Walter, James Middleton, Hanz Tantiangco and Nahal Shahidan. I must not forget a special thanks to the admin staff in the Information School department that always have answers to my questions and keep the department running smoothly.

I would also like to thank all my family and friends, who have provided constant love, encouragement and support but a special thanks must go to; my mum who is always proud of me and who is my hero; my sister, brother-in-law, niece and nephew, Charlie, Jon, Ella and Riley who are always there for me and provide fun and a good source of laughs; my grandparents, aunties and uncles and cousins, you have all been amazing.

Finally, I would like to thank Rhys, whose love and support has definitely got me to this point now. You have stood by me and read many drafts without complaint. I cannot imagine life without you.

I would like to dedicate this work to my dad, who unfortunately, passed away when I was a young girl. This has given me a drive to always make sure that you are proud of me.

This work would not have been possible without the financial support of the Engineering and Physical Sciences Research Council (EPSRC) and GSK.

Abstract

Providing working treatments safely, quickly and cost effectively are major factors in pharmaceutical companies. One stage within the drug discovery process is the lead optimisation (LO) stage. A LO project is an iterative process that alters a known active core to improve essential properties. These properties should be balanced to create an optimum drug candidate that can be progressed through the drug discovery process. These iterative refinements are typically represented using Markush structures and structure-activity relationship (SAR) tables. When used in unison, they can highlight core scaffolds and the varying surrounding substituents, which could be of particular interest. The use of these representations can highlight the impact small changes in R-groups can have on the property of interest. Unfortunately, however, they cannot provide the same for the core. This is because if there is a small modification to a core structure an entirely new Markush structure and SAR table are generated. Therefore, it becomes difficult to compare the SAR of closely related molecules with similar but non identical cores.

A main aim of this thesis has been to develop a new visualisation tool using reduced graphs (RGs) that allows closely related molecules to be represented. The emphasis in the tool is on functional groups that can form intermolecular interactions instead of chemical substructures.

Alongside the visualisation two scores have been developed. The first score, the exploration score, explains how much new chemical space and information would be added to the dataset when a new molecule is synthesised. The second score, the exploitation score, attempts to explain how important the core structure of a new molecule is to the biological activity. This takes into account the existing activity information. The new visualisation and scores can be used to aid decision making processes in LO projects when considering which molecules to synthesise next, by providing rationale based on different substructures within the core structures.

The final stage of this project is to generate new molecular structures based upon RG node alterations. A node substructure for a specified molecule can be substituted for another as long as it remains the same node type. This allows scaffold hopping to easily occur as the same RG structure is retained. These new molecules can have their exploration and exploitation score calculated to aid the decision on which molecule to synthesise in the next iteration of the LO project.

Table of Contents

List of Figures

List of Tables

Table of Common Acronyms

ADMET	**A**bsorption, **D**istribution, **M**etabolism, **E**xcretion and **T**oxicity
ASB	**A**nalog **S**eries **B**ased
BH	**B**all **H**all
CH	**C**alinski-**H**arabasz
DB	**D**avies **B**ouldin
DT	**D**ecision **T**ree
ECFP	**E**xtended **C**onnectivity **F**inger**p**rint
FW	**F**ree-**W**ilson
LO	**L**ead **O**ptimisation
GBM	**G**radient **B**oost **M**odel
GUI	**G**raphical **U**ser **I**nterface
IC50	Half maximal inhibitory concentration
InChi	**In**ternational **Ch**emical **I**dentifier
IUPAC	**I**nternational **U**nion of **P**ure and **A**pplied **C**hemistry
MCS	**M**aximum **C**ommon **S**ubstructure
MMP	**M**atched **M**olecular **P**airs
MW	**M**olecular **W**eight
PCA	**P**rincipal **C**omponent **A**nalysis
PLS	**P**artial **L**east **S**quares
PSA	**P**olar **S**urface **A**rea
QSAR	**Q**uantitative **S**tructure-**A**ctivity **R**elationship
RECAP	**RE**trosynthetic **C**ombinatorial **A**nalysis **P**rocedure
RF	**R**andom **F**orest
RG	**R**educed **G**raph
SMARTS	**SM**iles **AR**bitrary **T**arget **S**pecification
SMILES	**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**ystem
SVM	**S**upport **V**ector **M**achine

Preface

A massive issue within the pharmaceutical industry is that the drug discovery process is long and expensive. It has been determined that on average it takes twelve years and in excess of one billion pounds to get a new drug to market (Thomas, 2016). The drug discovery process is also extremely complex as the drug properties have to be balanced whilst ensuring there are no off target interactions that could cause harmful side effects. To improve the drug discovery process computational methods have been introduced and this area has more recently been referred to as chemoinformatics. Chemoinformatics is built on the similarity property principle, meaning that molecules that have a similar chemical structure should have similar chemical properties (G. Maggiora, Vogt, Stumpfe, & Bajorath, 2014).

The initial use of computational methods for handling chemical structures was in the 1950s, when techniques for searching databases of chemical structures were developed (Ray & Kirsch, 1957). Then a few years later the first quantitative structure-activity relationship (QSARs) method was established (Hansch & Fujita, 1963). With the advancement of computational technologies more sophisticated techniques have been developed. These included new data mining techniques, visualisations and machine learning methods, providing greater insights into chemical data and structure-activity relationship (SAR) information and suggestions of new molecules to create. For all of these methods, molecular descriptors are needed as inputs. The Sheffield Chemoinformatics Research Group has developed a novel molecular descriptor called the reduced graph which is based upon functional groups that have potential to form interactions with biological receptors. This descriptor is the main focus of thesis.

This thesis aims to overcome the issues associated with Markush structures and SAR tables and generate a new representation that summarises data generated within a lead optimisation (LO) dataset. Methods are then developed that use the information that can be extracted from the visualisation. The substructural fragments from the new representation are taken into account, so that when examining potential new molecules to synthesise, a score for the new chemical space being added to is given along with the potential significance of these fragments. All of the work done within this thesis attempts to aid the decision process for medicinal chemists, whilst generating new ideas for molecules to synthesise in the next iteration of the LO series.

Chapter 1 provides an introduction to chemoinformatics by introducing basic concepts that form the basis of this thesis. It will discuss molecular representations, molecular descriptors and how they can be used in different chemoinformatics techniques. A brief introduction to how machine learning can be applied to chemical data is then given. Different types of visualisations that have also been used to display chemical information are discussed. The final concept introduced is molecule generation.

In Chapter 2, a method is developed to identify RG cores to represent molecules in lead optimisation series that can be used to illustrate the relationship between several molecules that share similar scaffolds. RGs are found for all the molecules and several clustering techniques are investigated to identify the best method to precluster a dataset prior to the application of the RG core extraction technique. Chapter 3 then evaluates the RG cores that have been generated from both the clustered dataset and the dataset as a whole. This comparison provides an understanding of which method creates the best RG cores to describe the relationship between molecules.

Chapter 4 establishes how the RG cores can be mapped back onto the RGs of the molecules within the LO dataset as there are several instances where a molecule's RG can have multiple occurrences of the RG core. The optimal overlap with the other molecules within the dataset is found. Chapter 5 then describes how these RG mappings are used in the new visualisation, that has been developed to incorporate the SAR data across similar scaffolds, through the use of the RG cores.

Chapter 6 and Chapter 7 introduce the work undertaken to generate the exploration and exploitation scores. Chapter 6 investigates how to best produce a score that indicates the level of exploration a new molecule would achieve. The score looks at how much chemical space has currently been explored and how much new information a new molecule would provide. Chapter 7 then investigates creating an exploitation score for a potential new molecule. This indicates how significant each substructure within the RG core structure is to the biological activity. Therefore, a high score would potentially indicate an active area of chemical space that could be of interest to investigate, whereas a low score would potentially indicate an inactive area of chemical space. For both scores a hold out set from LO datasets are scored to understand their performances of how useful the scores are.

Chapter 8 describes the development of a new approach to molecular generation through the use of RGs. Three different approaches have been constructed, a single node alteration, multiple node alterations and a full enumeration. The molecules generated from each method undergo a validation step to indicate how useful this technique is to generate new molecules and whether it can identify molecules within the LO dataset by introducing a hold out set.

Finally, Chapter 9 summarises all of the work and brings together all conclusions, outlines some limitations and provides suggestions for future work.

1 Introduction to Chemoinformatics

Chemoinformatics is a relatively new discipline, although many of the methods that it encompasses have been well researched for many years with chemoinformatics seen by some as a new name for an old problem (Hann & Green, 1999). It has been an instrumental part of several industries, particularly providing significant advancements to the drug discovery process. With the advancements in computing and technology, chemoinformatics is a fast paced field with lots of literature being produced.

This chapter reviews chemoinformatics techniques that provide background to the research carried out in the thesis. The first principle to be reviewed is molecular representations. The molecular representation problem is notorious within chemoinformatics. Subsequently, it involves finding the best molecular descriptor that provides the most information and is most suited to the job. Therefore, several molecular descriptors are reviewed, followed by some important chemoinformatics applications, similarity searching, and reduced graph (RG) applications. The remainder of the chapter will review literature from a variety of procedures that utilise the molecular descriptors: machine learning, visualisations and molecular generation.

1.1 Molecular Representations

Different methods of molecular representation have been adopted to allow chemists and computing systems to understand the molecules that are being investigated. The molecular representations fall under two categories, human readable and computer readable. Both will be discussed.

1.1.1 Human Readable

In order to be able to communicate about molecules, chemists need some way of representing them. Hence different representations have been established. Human readable representations have been around a considerably longer time than computer readable ones. The most common way chemists ensure there is universal understanding is through naming systems, chemical formulas and chemical diagrams.

There are two well-established naming systems: International Union of Pure and Applied Chemistry (IUPAC) and Chemical Abstracts Service (CAS) (Chowdary, Sri, Prasanna, Sudhakar, & Sarathi, 2014; Mills, Cvitas, Homann, Kallay, & Kuchitsu, 1993; *Naming and Indexing of Chemical Substances for Chemical Abstracts TM 2007 Edition A publication of Chemical Abstracts Service*, 2008). Both have different rubrics and conventions, however, they both systematically identify the chemical nomenclature. This means that they are based on the functional groups and their positioning within a molecule. Figure 1-1 shows the IUPAC and CAS name for Aspirin. Additionally, when a drug is identified and brought onto the market it is given a commercial name in order to make it easier for the consumer market, for example Aspirin.

Drug Name: Aspirin
IUPAC: 2-Acetoxybenzoic acid
CAS: Acetylsalicylic acid (ASA)
Chemical Formula: $C_9H_8O_4$
Extended Chemical Formula: $CH_3COOC_6H_4COOH$

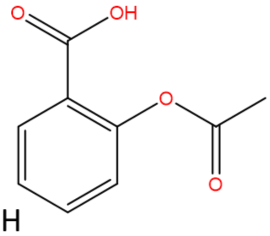


Figure 1-1: Human readable representations for Aspirin

Another way in which chemists communicate molecules is by using chemical formulae. A chemical formula identifies the atom types in the molecule along with the count. A chemical formula is very generic as it can represent several different molecules that contain the same types and numbers of atoms. To overcome this, an extended chemical formula is used. This provides information on the connectivity of the atoms which reduces the number of molecular possibilities down to one.

The final approach is a two-dimensional (2D) structural drawing. This allows a more visual representation and allows more clarity than the two chemical formulas. Chemists follow several different chemical conventions so that there is a basic chemical understanding. For example, the bonds are arranged to best represent their actual structure and hybridisation states on a page (Brecher, 2008). Therefore, this is a suspected 2D representation of what the molecule looks like from using the rules of chemistry.

2D structural drawings can be used to represent multiple molecules simultaneously. For example, chemists working in the lead optimisation stage of drug discovery often use a generic structure in order to display multiple molecules that are related to one another in one simple representation. One way this is achieved is through Markush structures. Markush structures have two main uses. The first is in patent applications so that a multitude of similarly related compounds can be classed/ protected within the patent. The second is used to develop and explain chemical libraries. The part of the molecule that is common to them all, remains constant and is, therefore, only displayed as a 2D structural drawing once. The groups that vary are also recorded, however these are not part of the main structure (Warr, 2011).

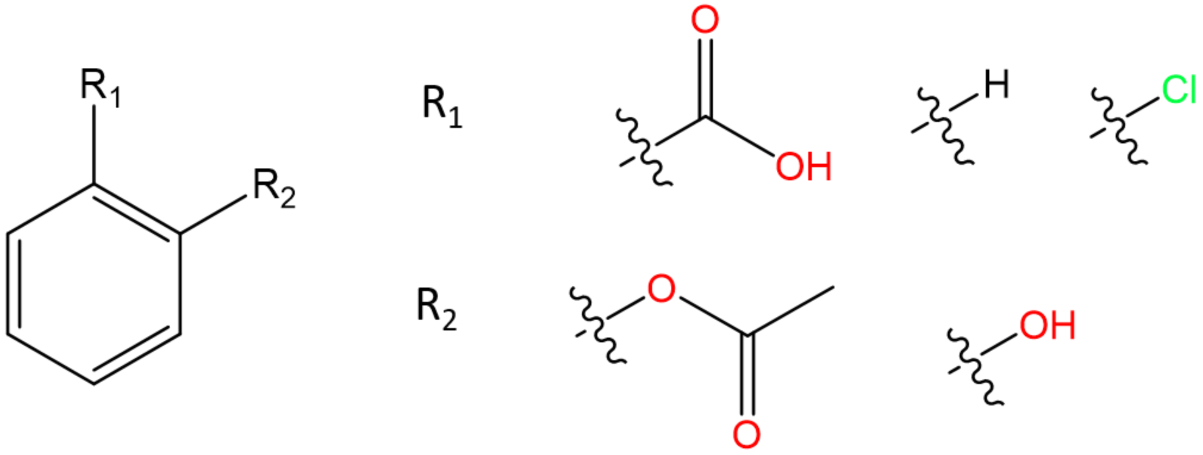


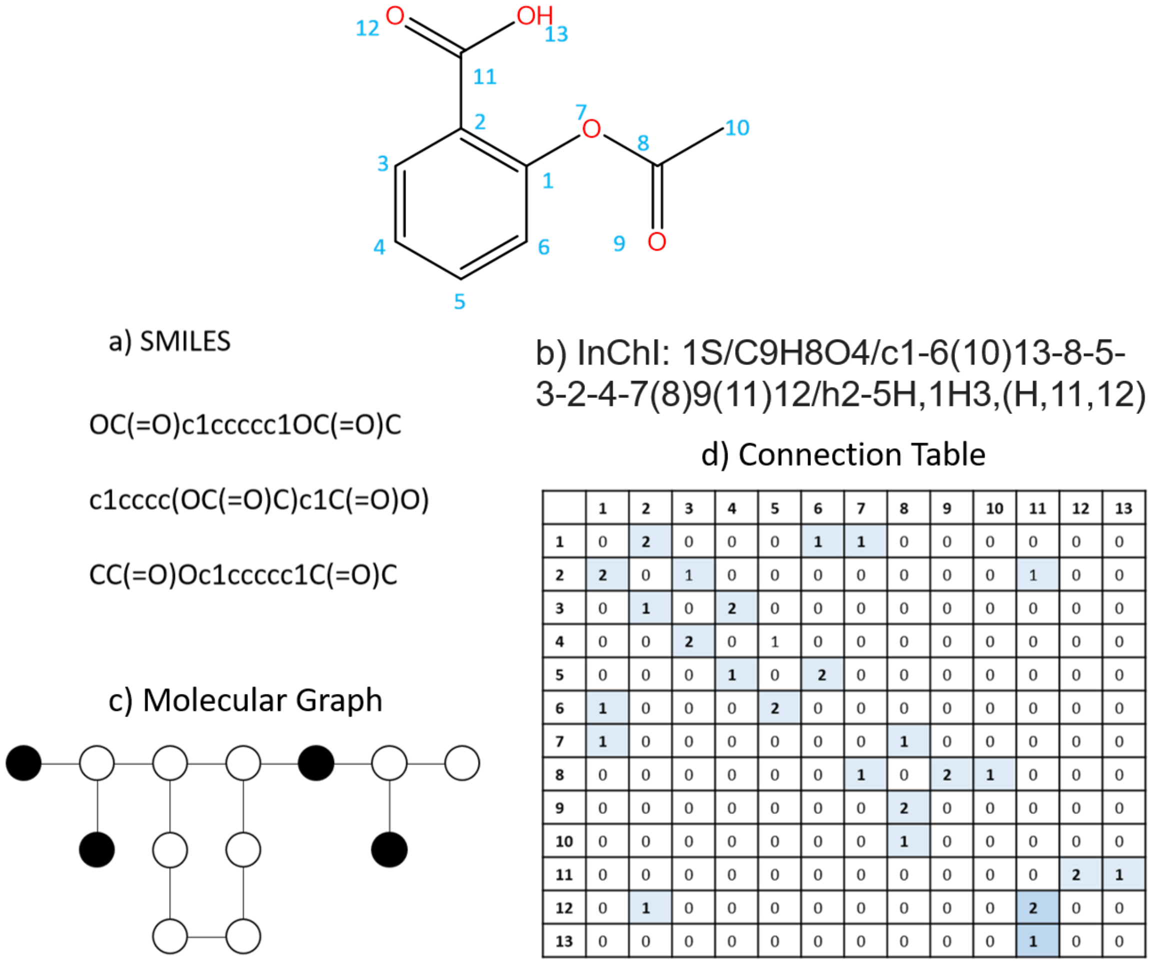
Figure 1-2: Simple example of a Markush structure

Figure 1-2 demonstrates a simple Markush structure. On the left hand side, the core of the structure can be seen and the different R groups can be seen on the right hand side. This example represents six different molecules which can be constructed by combining the different R groups to the core. For some complex Markush structures the R groups may also contain R groups.

1.1.2 Computer Readable

For computer processing, chemical compounds have to be represented in machine readable form. The representation problem is a fundamental part of chemoinformatics, trying to find the best way to represent chemical compounds for a particular task.

There are many types of molecular representations that are commonly used to describe the atoms and bonds within a structure while also making an easily storable representation. Molecular representation types can be grouped by dimensions. One-dimensional representations are linear notations that are built from alphanumeric characters to characterise a molecule. Two-dimensional representations can be seen like a graph as they indicate which atoms are present and the connections of the atom within the molecule. Two-dimensional representations may also contain x and y coordinates to allow them to be drawn on a page. Three-dimensional representations contain the same information as two-dimensional representations, however, they provide additional information regarding the geometry of the molecule. Three-dimensional representations are more accurate than the previous two, nevertheless, this added accuracy makes them much more computationally expensive to compute. Here, the four main representation of 2D structures will be discussed: SMILES and SMARTS, InChI keys, chemical graphs and connection tables.



a) SMILES

OC(=O)c1ccccc1OC(=O)C

c1cccc(OC(=O)C)c1C(=O)O)

CC(=O)Oc1ccccc1C(=O)C

b) InChI: 1S/C9H8O4/c1-6(10)13-8-5-3-2-4-7(8)9(11)12/h2-5H,1H3,(H,11,12)

d) Connection Table

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	2	0	0	0	1	1	0	0	0	0	0	0
2	2	0	1	0	0	0	0	0	0	0	1	0	0
3	0	1	0	2	0	0	0	0	0	0	0	0	0
4	0	0	2	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	2	0	0	0	0	0	0	0
6	1	0	0	0	2	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	1	0	2	1	0	0	0
9	0	0	0	0	0	0	0	2	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	2	1
12	0	1	0	0	0	0	0	0	0	0	2	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	0

Figure 1-3: Aspirin computer readable representations. a) SMILES b) InChI Key c) Connection Table d) Molecular graph

Simplified Molecular Input Line Entry Specification (SMILES) is the simplest and most popular way that a molecular structure can be represented and can be described as a linear notation of a 2D structural drawing (A. R. Leach & Gillet, 2007; O'Boyle, 2012; Weininger, 1988). It contains information about which atoms are bonded and the bond types as well as the element types and whether an atom is aromatic or aliphatic. SMILES can also include charges and chirality of atoms. SMILES have become popular due to their ease of use and low computational and memory costs. A drawback of SMILES is that a molecule can be represented in many different ways (see Figure 1-3a). One way to overcome this is to apply a canonicalization algorithm to the SMILES representation so that it always gives the same unique SMILES for the molecule (A. R. Leach & Gillet, 2007). SMILES arbitrary target specification (SMARTS) is an extension of the SMILES notation that is used to represent specific substructures. SMARTS are typically used in searching for substructures within molecules. The main fundamentals remain the same between SMILES and SMARTS, however, SMARTS allows the inclusion of more symbols. Two examples are that a wildcard atom, *, can be specified which means that this atom can be any element type, and an exclamation mark, !, can be used to specific not a particular element or atom type (James, Weininger, & Delany, 2020).

The International Chemical Identifier (InChI) is an alternative linear representation method that can contain more information, however, this information is still restricted to two-dimensional characteristics. There is only one standard InChI for each molecule. Just like the SMILES representation, InChI allows the identification of different stereoisomers, easily allowing different structural information such as E/Z, alkene bond or R/S chiral centres to be distinguished. The InChI canonicalization algorithm generates a unique standard InChI for a molecule which can then be hashed to create the InChI key (Heller, McNaught, Pletnev, Stein, & Tchekhovskoi, 2015; O'Boyle, 2012).

There are also two dimensional representations. The 2D structural drawings can be adapted to be represented on a computer, this is becomes known as a chemical graph. The molecule is represented as a graph composed of nodes and edges, as can be seen in Figure 1-3c. The nodes are the circles and characterise the atoms of the molecule, which can have properties linked to them. The edges are the lines which signifies the bonds. An extra level of sophistication can be added by colouring the nodes and bonds to differentiate between the different types. Chemical graphs can be represented as connection tables in order for the computer to interpret them (A. R. Leach & Gillet, 2007).

A connection table is a simple table representation that attempts to replicate the connections that are made within a molecule. The atoms are enumerated and the bond order between each of the atoms is recorded within the table (for example, 1=single, 2=double, 3=triple), see Figure 1-3d. In most connection tables, hydrogen atoms are ignored. The level of complexity of the table was further developed in 1992 by Dalby et al. by adding more information into the connection table such as the hybridisation states, and xyz coordinates of each atom (Dalby et al., 1992; A. R. Leach & Gillet, 2007).

1.2 Molecular Descriptors

Many chemoinformatics applications require molecular descriptors to be defined which are experimentally measured or theoretically derived properties of a molecule. The molecular descriptors attempt to encapsulate as much chemical knowledge about the molecule as possible. This knowledge can include molecular properties such as the size, the shape, the symmetry, the stereochemistry, the branching, sterics, the cyclicity, and the hydrophobicity. Molecular descriptors can be derived from several different principles. These principles consist of graph theory, topology, mathematics and multiple chemistry theories, such as chemical properties and potential bindings. These descriptors can then be used in different computational methods for more analysis and understanding. This section will discuss some of the main molecular descriptors that are commonly used in chemoinformatics.

1.2.1 Simple Counts and Physicochemical Properties

The simplest molecule descriptor is to use counts of features that are of interest. For example, these features can include counts of atom types, ring systems, rotatable bonds, hydrogen bond donors and hydrogen bond acceptors.

Physicochemical properties can be calculated and also be used as a molecular descriptor. These can either be computed values or experimental values. Physicochemical properties that are typically used are molecular weight, lipophilicity (logP), solubility, polar surface area and molecular flexibility. These properties can be calculated either on an atomic level or fragment level.

Both of these descriptors do not provide much information about the molecule itself due to their simplicity so they are often used in combination with other descriptors or can be used more as a way of filtering molecules (A. R. Leach & Gillet, 2007).

1.2.2 2D Fingerprints

Fingerprints were originally generated for substructure searches and then subsequently used for similarity searching (Willett, 1987). Fingerprints are very popular descriptors due to their ease of use and quick computational time. They are also preferred over graphical representations as they are much more efficient, can quickly capture complex relationships even with large amounts of data and contain similar information to a chemical graph. However, this is highly dependent on the fingerprint method used as some can be very basic whereas others can be more complex (Hasan, Bonde, Buchan, & Hall, 2012; Wollenhaupt & Baumann, 2014). 2D molecular fingerprints do not need to be canonicalised, therefore, for a fingerprint method the fingerprint for a molecule will always be the same.

There are three fingerprint types commonly used: structural keys, hashed fingerprint, and extended connectivity fingerprints (R. D. Brown & Martin, 1996; Zahoránszky-kőhalmi, Bologa, & Oprea, 2016). A molecular fingerprint is a binary key that indicates whether certain fragments appear within a chemical structure or not. Structural keys are based on substructural fragments which can be atom-centred or path-based. These can be dictionary-based where each bit in the fingerprint corresponds to a fragment in the dictionary. In hashed fingerprints, the fragments are generated algorithmically and are based on paths of atoms to a predefined length which form the inputs to a hashing algorithm to generate a number of bits to set in the fingerprint. A feature of hashed fingerprints is that they are not easy to interpret as the hashing algorithm makes it difficult to identify the fragment without knowing the specific hashing algorithm (Stepniewska-Dziubinska, Zielenkiewicz, & Siedlecki, 2017).

Extended connectivity fingerprints (ECFP) were designed in an attempt to capture relationships between molecular fragments and molecular activity. ECFPs are also referred to as circular fingerprints and take into account a central atom and the surrounding atoms at increasing neighbourhoods. Each atom is initially given its own atom identifier, then depending on the chosen diameter, the neighbouring atom identifiers are combined into an array in an iterative manner. This is performed for each atom. The arrays are then hashed so that there is one single identifier for each generated array, that is, for each atom. Therefore, the amount of information contained within the ECFP is dependent on the diameter assigned. Like the hashed fingerprints described above, ECFPs can be generated on the fly and do not depend on predefined structural keys. Therefore, they can cover a wide range of functional groups including unusual or novel groups. Due to their advantages, they are frequently used for activity modelling and similarity searching. Another advantage is that they automatically capture stereochemistry and allow rings that have ortho, meta or para substituents to be distinguished.

Bit folding can occur which reduces the fingerprint down to a particular size. Hashing (and bit folding) can lead to bit collisions which is where two different substructures are characterised as the same identifier (Rogers & Hahn, 2010).

1.2.3 Reduced Graphs

A reduced graph (RG) is a compressed representation of a molecular structure that is focused upon functional groups that could form interactions with a biological receptor. Groups of connected atoms are reduced to individual pharmacophore-type nodes, for example, an aromatic ring is reduced to one node, see Figure 1-4. The nodes in the reduced representation are connected according to the original structure to form a reduced graph. There are several ways in which this compression can be represented. One of the main ways to represent them is to use SMILES. This is achieved by mapping the different node types to atom types that are outside the usual set used for organic molecules.

RGs were originally created to describe the chemical space that is covered by a Markush structure as defined in a patent application (Gillet et al., 1987; Gillet, Holliday, & Willett, 2015). They were then applied to similarity searching with the aim of highlighting the types of interactions that functional groups could make rather than specific functional groups. The focus on the types of functional groups rather than particular atoms within molecules can lead to molecules with different 2D structures being identified as similar (Birchall, Gillet, Harper, & Pickett, 2008; Gillet, Willett, & Bradshaw, 2003; Harper, Bravi, Pickett, Hussain, & Green, 2004).

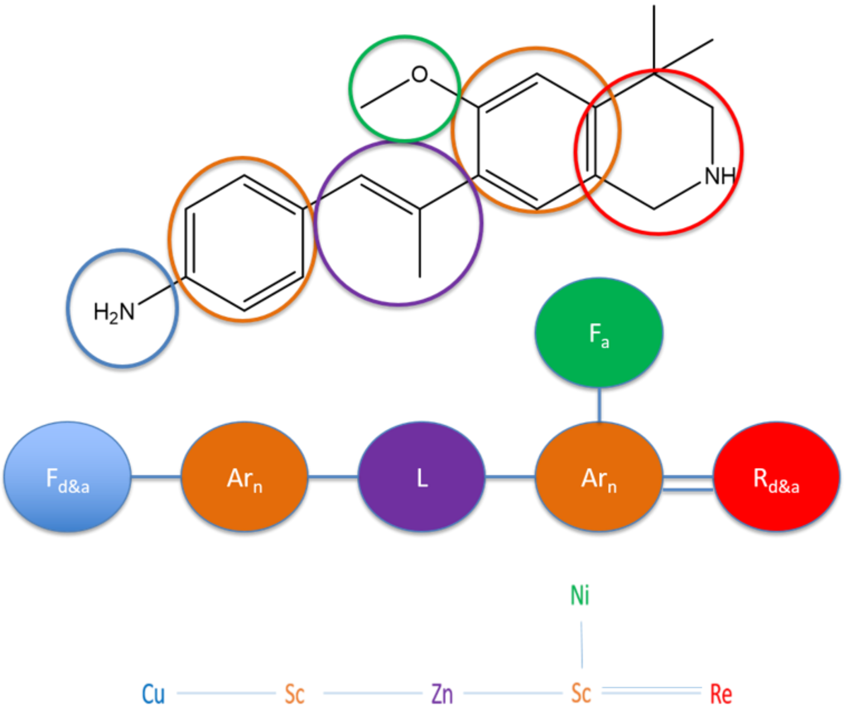


Figure 1-4: An example of a reduced graph representation (Birchall & Gillet, 2010)

Different types of graph reduction can be applied and different amounts of information can be associated with the nodes. Figure 1-4 is the most basic example of a RG, the hydrogen bond acceptor and/ or donors have been found and are defined as feature nodes, F. For ring nodes, whether the ring is aromatic, Ar, or aliphatic, R, is also found. Any atoms that have not been assigned and connect nodes are then assigned as linkers, L. Other definitions include also encoding positive and negative ionisable features as node types (Gillet et al., 2003; Harper, Bravi, et al., 2004).

There have also been similar approaches developed with the same principles of bringing together atoms with similar binding potentials. Rarey et al. established the feature tree (Rarey & Dixon, 1998). The feature tree is a tree of nodes and edges that replicates the functional features and connections of a chemical graph. The difference between a feature tree and a RG is that the feature tree does not contain cycles, whereas the RG can. Feature trees have also been used for similarity searching applications.

Stiefl et al. generated the extended RG (ErG) (Stiefl, Watson, Baumann, & Zaliani, 2006). The fundamental approach is similar to the RG with the key features of a molecule that could be important in forming interactions with a receptor being emphasised. However, the ErG is more complex as it tries to conserve the size and shape of the original molecule. This is achieved in several steps. The first is that the positions of the nodes, by retaining the distances between the node with features. Therefore, the linker node is redundant. The second is how ring nodes are defined. For each ring a centroid is established and any atoms that are potential features are also defined as a node. Then for any atom that has a substation site, including fused rings, are retained and connected to the ring centroid node. If any ring atoms have not been defined or connected they are then removed. Stiefl et al. generated ErG into its own FP that can be useful in virtual screening experiments (Stiefl & Zaliani, 2006).

Another approach that attempts to use similar principles and exploit the observed pharmacophoric features is DeCAF (Discrimination, Comparison, Alignment tool for 2D Pharmacophores). DeCAF generates a pharmacophore model representation, seen in Figure 1-5, where five key pharmacophoric features are highlighted, hydrophobic, aromatic, hydrogen bond donor, hydrogen bond acceptor and ring atoms. Each atom is highlighted depending on these pharmacophoric features and the ring systems are then combined to one node. Each feature has an assigned weight which can be altered if the importance of a feature is considered to be more significant. When expressing multiple molecules, it can indicate the frequency of each feature. The edges between the nodes indicates the distances in bond in the chemical graph. Unlike RG and ErG, DeCAF are kept and utilised in their graphical forms (Stepniewska-Dziubinska et al., 2017).

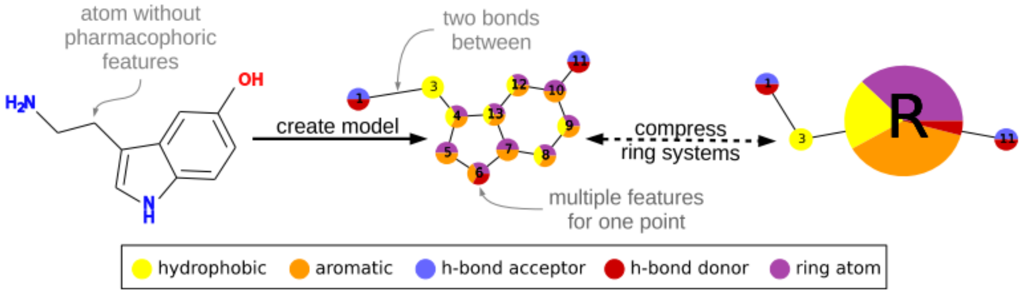


Figure 1-5: DeCAF pharmacophore model of a single molecule (Stepniewska-Dziubinska et al., 2017)

Some of the ways in which these different molecular descriptors are utilised in chemoinformatics applications are now described.

1.3 Chemoinformatics Applications

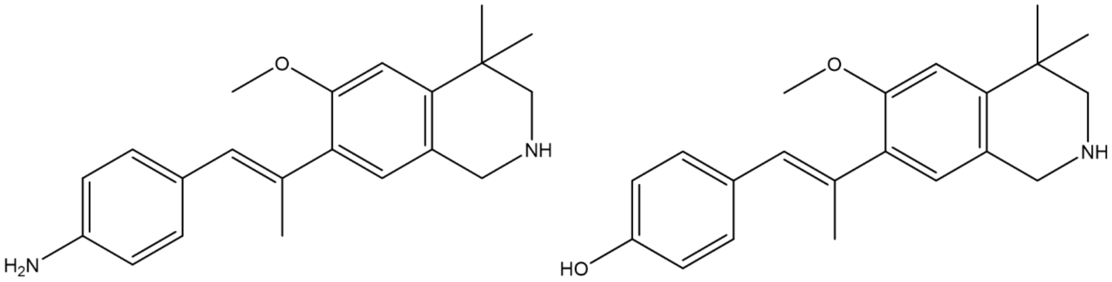
The molecular descriptors are implemented in a way in which they can be useful in various areas of chemoinformatics. Several of these areas are described below.

1.3.1 Similarity

Molecular similarity plays an important role in being able to establish common patterns that emerge within a chemical series due to the identification of a relationship between molecular structure and activity. The rationale for similarity searching is that molecules that have similar structures are also similar in biological activity (Willett, 2005). Similarity searching can be used with many different molecular descriptors, for example, fingerprints, chemical graphs and reduced graphs, with 2D fingerprints being most common but to them having shown a large amount of success with very little computational cost (Arif, Holliday, & Willett, 2010). Some of the main methods are described below.

1.3.1.1 Fingerprint Similarity

The similarity of two fingerprints can easily be calculated. As a fingerprint shows the present or absence of particular fragments as 1's and 0's, the number of bits that are the same can be calculated. An example of a fingerprint comparison is shown in Figure 1-6, when the bits are the same a value of one is given if the values are different a value of zero is given. These values are then totalled up and input into the Tanimoto similarity coefficient, which is explained later in this section.



Molecule 1

Molecule 2

	Bit 1	Bit 2	…	Bit 26	…	Bit 746	…	Bit 889	…	Bit 1920	…	Bit 2047	Bit 2048	
Molecule 1	0	0		1		0		1		1		0	0	
Molecule 2	0	0		1		1		0		1		0	0	
Comparison	1	1	…	1		0	…	0		1		1	1	Total = 2039

Figure 1-6: An example of a fingerprint comparison

2D fingerprints have been demonstrated to be effective in similarity searching especially for molecules that are close analogues. Conversely, the main drawback with 2D fingerprints is that they do not allow compounds to be found that share the same biological activity but have different 2D structures (Gillet et al., 2003). This also indicates that they are not good at scaffold-hopping which is one of the desired abilities of molecular representations as it allows the exploration of previously unexplored chemical space (Stepniewska-Dziubinska et al., 2017).

Brown et al. found that the most effective 2D descriptor for clustering aimed at finding compounds that share the same activity was the structural key representation even though this encodes less structural information, compared to the hashed 2D fingerprint. It was suggested that the hashed fingerprints were not as effective as expected due to the hash key encoding a large amount of information within a small number of bits (R. D. Brown & Martin, 1996). Arif et al. investigated the use of weighting systems in the similarity scoring between two two-dimensional fingerprint representations. The first approach established the frequency of each fragment within the set and weighted them accordingly, the more it appears the greater the importance to the system activity. Contrastingly, the second approach interprets that if a fragment is frequently hit then this means that the fragment is not important in the molecular activity. Both methods varied drastically depending on the definition of the weighting, which is very dataset dependent, and therefore a wide variety of results can be obtained which is not desirable (Arif, Holliday, & Willett, 2009; Gillet et al., 2015).

1.3.1.2 Maximum Common Substructure (MCS)

A maximum common substructure (MCS) is a set of common atoms and bonds between two molecules. From a graph-theoretical point of view, the MCS of two molecules is defined as the maximum common edge subgraph (MCES) or maximum common induced subgraph (MCIS) of two graphs. MCES is a subgraph that contains the largest number of edges that are common between both molecules. Whereas MCIS is a subgraph that contains the largest number of vertices, see Figure 1-7. The identification of the MCS allows a degree of overlap to be established between two molecules which can be used as a measure of their similarity (Duesbury, Holliday, & Willett, 2017).

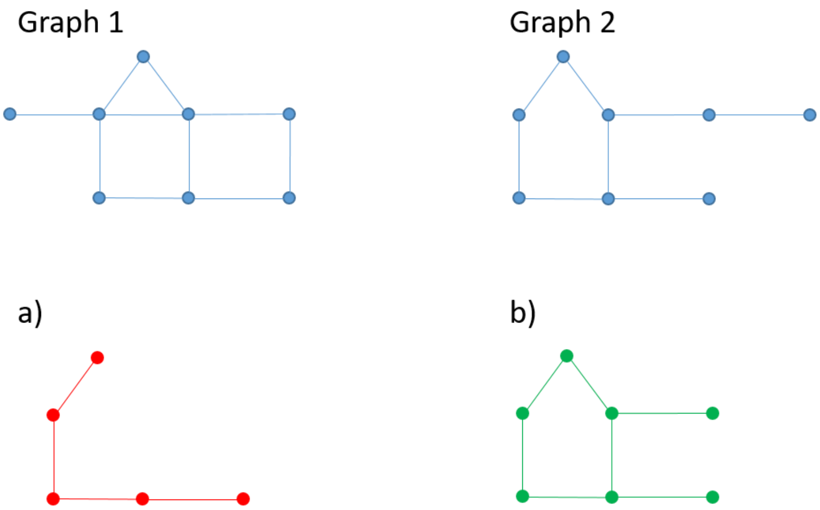


Figure 1-7: MCSs of Graph 1 and Graph 2 a) MCIS b) MCES

In the first definitions of the MCS, the algorithms were designed to find the maximum common connected substructure between two molecules. For example, in Figure 1-8 all of the atoms and bonds highlighted in green would form the MCS. The rationale for this approach is that the environment of an atom affects its properties, for example, a nitrogen atom has very different properties in pyridine as it does within piperidine.

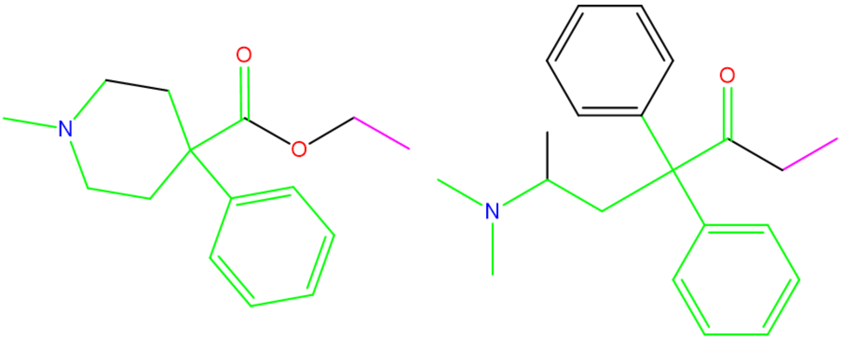


Figure 1-8: An example of an MCS produced from RASCAL (Raymond, Gardiner, & Willett, 2002). *All coloured bonds are part of the MCS*

Finding the MCS between two graphs is computationally expensive for large datasets (Willett, 2005; Wollenhaupt & Baumann, 2014). A clique is a set of nodes that are plotted on a graph that are connected and all nodes are connected to one another. A maximal clique is the largest clique within a graph which cannot be a subset of another clique (Ostergard, 2002). Barrow and Burstall were the first to attempt to overcome the computational costs of MCS detection by converting it to a maximal clique problem (Barrow & Burstall, 1975). Stepniewska-Dziubinska et al. further adapted the Barrow and Burstall MCS algorithm and combined it with Bron-Kerbosch's algorithm for finding maximal cliques. The maximal clique is the clique with the highest similarity score, signifying that it the best alignment of all the models. They also included the topological distance of the bond as a constraint

(Stepniewska-Dziubinska et al., 2017). This methodology uses a similar algorithm to Raymond et al.'s program RApid Similarity CALculation (RASCAL) that calculated the similarity between two molecules. RASCAL is a fast similarity finding algorithm designed to analyse large-scale data sets. This algorithm provides chemical similarity searching on graphs instead of fingerprints. This is done by finding the maximum common induced subgraph (MCIS) and the maximum common edge subgraph (MCES). The RASCAL algorithm was compared to the MC1 algorithm (Wood, 1997), Ostergard maximum clique problem (Ostergard, 2002) and a maximal clique detection algorithm (Bessonov, 1985). The RASCAL algorithm outperformed these methods due to it being more efficient and stable as it had a quicker run time. It also allowed the classification of lower minimum similarity index (MSI) thresholds which is crucial as this allows potential scaffold-hopping as they have a bigger structural difference (Raymond et al., 2002).

1.3.1.3 Tanimoto Coefficient

The Tanimoto coefficient is used to quantify the similarity between two molecules (Willett, 2005; Zahoránszky-kőhalmi et al., 2016). The Tanimoto coefficient equation is as follows:

$$T_{m_A m_B} = \frac{|\mathrm{A}| \cap |\mathrm{B}|}{|\mathrm{A}| \cup |\mathrm{B}|} \qquad (1.1)$$

Where the Tanimoto similarity is the number of features common to molecules A and B divided by the number of features in A or B (Zahoránszky-kőhalmi et al., 2016). Therefore, for the example within Figure 1-6, only the comparison bits set to "1" are counted in the Tanimoto, 2039, and the number of features within A, molecule 1, are 2048 as this is how many bits are within the fingerprint. So the Tanimoto similarity coefficient for the ECFP fingerprints is 0.996.

The MCS is the largest subgraph that is common to molecules and can be converted to a similarity value through the use of a similarity coefficient in a similar way to quantifying similarity based on molecular fingerprints. The Tanimoto coefficient equation has been adapted for this purpose (G. M. Maggiora & Shanmugasundaram, 2004).

$$T_{m_A m_B} \frac{MCS}{A + B - MCS} \qquad (1.2)$$

Where A is the number of atoms within molecule A, B is the number of atoms within molecule B and MCS is the number of atoms within the MCS between molecules A and B.

Even though the Tanimoto coefficient is still the most widely used similarity coefficient in chemoinformatics, several people have investigated different similarity coefficients, such as Forbes and Russell-Rao coefficients. These were seen to give similar results if not superior to the Tanimoto coefficient in certain circumstances (Arif et al., 2009; Willett, 2005).

1.3.2 RG Applications

RGs have many useful applications in chemoinformatics. The RG has been shown to be useful in similarity searching. There are three different approaches that can be used to calculate the similarity between two molecules based on their RGs. These are by generating fingerprints from the RGs, by comparing the graph directly or using edit distance methods. As the RG is represented as a SMILES then a path-based fingerprint can be generated (Birchall & Gillet, 2010; Gillet et al., 2003). Another type of fingerprint that has been investigated is node-pair descriptors, which have been shown to have a greater success in comparing the RGs (Barker, Gardiner, Gillet, Kitts, & Morris, 2003; Birchall & Gillet, 2010).

The RGs can also be compared directly using graph matching techniques, such as the MCS. RGs are smaller graphs than chemical graphs, therefore, this allows their comparison to be more computationally efficient. There are several research groups who have looked into similarity searching based on RGs and achieved some level of success (Harper, Bravi, et al., 2004).

Gunera et al. also attempted to create a method for similarity searching that involves RGs as a combination of graphs and fingerprints. However, this methodology incorporated the use of colour on the nodes to be able to interpret more data. This research aimed to identify bioisosteres, which is where the functional groups within a molecule are swapped but the physicochemical properties remain unchanged, i.e. they remain biologically active, this is called scaffold hopping (Gunera & Kolb, 2015).

Harper et al. investigated similarity using edit distances and RGs. Two different types of distances were defined: a simple edit distance and weighted edit distance. A simple edit distance is where nodes can only be inserted, deleted or mutated in order to make two RGs match. In contrast, a weighted edit distance associates different weights to the insertion or deletion of different functional groups. They showed that RGs can be used in parallel with other methods to identify molecules that before would not have been found using just one method (Harper, Bravi, et al., 2004).

Birchall et al. further developed this work by creating a genetic algorithm that assigned the weighting of the edit distances. This was found to outperform and differ significantly to Harper et al., with a higher recall rate across a range of different activity classes. These weightings were seen to be more effective as they had been trained over several different activity groups (Birchall, Gillet, Harper, & Pickett, 2006).

The RG has also been used in various clustering algorithms. Harper et. al first used RGs to cluster high-throughput screening data. The molecules were clustered dependent on several descriptors, either FP or RG of the molecules, and their respective similarity techniques, for RGs this is either substructure searching, FP, or edit distance. Molecules that had similar descriptors were then brought together in a cluster (Harper, Bravi, et al., 2004).

Gardiner et al.'s research aimed to find a better way of representing the main relationship in a cluster. Gardiner et al. initially clustered a database, then for each cluster, they found an RG that is the MCES for as many molecules within a cluster. This method was successful in identifying an RG for a cluster that is representation of the majority of the molecules and which can be mapped back to a chemical graph to provide information on subgraphs that are important to the activity. (Gardiner, Gillet, Willett, & Cosgrove, 2007).

RGs have been used to visualise SAR. One method is by Wollenhaupt et al. in their generation of inSARa program. An RG algorithm is first applied to the molecules, after which the MCS is found. As the RG dramatically reduces the size of the molecular representation, computing the MCS becomes a less computationally expensive task. This technique can be applied to large-scale problems such as large-scale SAR analysis. In this technique, a root MCS is established in order to generate a hierarchical network structure. An important key finding from Wollenhaupt et al.'s research is that it is important to not define the size of the root MCS too small. In their results, an RG-MCS greater than or equal to three could only found for 50% of the data. Therefore 50% of the data have an MCS of less than three nodes and this could just be from random MCS that happened to have two things in common. It was suggested that to become clustered the minimum number of MCS nodes should be four to help rule out random similarity or molecular features that appear in most molecules which does not provide any information about the SAR, for example, linker nodes (Wollenhaupt & Baumann, 2014).

1.4 Machine Learning

A huge part of chemoinformatics and data mining is using the data to extract knowledge. This is also particularly important in the lead optimisation (LO) stage of the drug discovery process. Several techniques have been developed to prioritise molecules to make based upon properties which can be learnt from the previous data. The properties that are of particular importance are the biological potency and the absorption, distribution, metabolism, excretion and toxicity (ADMET) properties.

Machine Learning has become a popular technique in chemoinformatics due to its ability to allow the computer to learn information. Machine learning is typically classified into two types: supervised and unsupervised (Witten, Frank, Hall, & Pal, 2016). There have been several successful uses of supervised and unsupervised learning in chemoinformatics, some of which are reviewed below.

1.4.1 Supervised Learning

Supervised learning gets its name as the algorithm learns under supervision. Supervised learning requires labelled training data and the aim is to learn the relationship between the input label and the set of variables that are used to present the objects, that is, how one maps onto the other. The resulting model is then able to predict the corresponding label for a new molecule from its known variables (Clarke et al., 2008; Witten et al., 2016). Quantitative structure-activity relationships (QSAR) is an example of how supervised machine learning can be applied to a chemoinformatics problem and is described below, along with several widely used supervised learning algorithms.

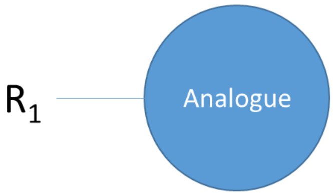
1.4.1.1 Quantitative Structure-Activity Relationship

Quantitative structure-activity relationships (QSAR) is a mathematical technique used to derive a relationship between structural features of molecules and a measured property, particularly their activities. QSAR models are either regression or classification models that are used to link features (X) of molecules to their potencies (Y). In a regression model Y is an activity value and in a classification model Y is an activity category. Molecules that have previously been unseen by the model can then have predictions made using the QSAR model. QSAR was first developed in 1962 by Hansch and Fujita who predicted the reactivity and molecular lipophilicity of phenoxyacetic acids by modelling on benzoic acid substituents (Hansch, Maloney, Fujita, & Muir, 1962). QSAR modelling can be used throughout the drug discovery projects as it allows predictions to be made on the activity of a molecule or its ADMET properties to establish a more optimised molecule. Molecular fingerprint-based representations are the most commonly used descriptors in QSAR (Wollenhaupt & Baumann, 2014).

QSAR models are developed using a training set of molecules. The model can then be applied to unseen molecules to make predictions (Cherkasov et al., 2014). A QSAR model can be built via two different methods: internal cross-validation, where a percentage of the training set is taken out before the model is built and these molecules become the test set. This is done several times and an average is taken of the model performance. Or an external validation can occur with a pre-established set of compounds that are not contained within the training data, as long as the molecules are within the domain of applicability (Dearden, Cronin, & Kaiser, 2009).

1.4.1.1.1 Free-Wilson Analysis

Free and Wilson is a common approach in medicinal chemistry due to its simplicity and interpretability. A Free-Wilson analysis estimates the bioactivity of the molecules by summing the activities of the substructural fragments that constitute the molecule (Wilson & Free, 1964). The molecules that are used to create the model only need to be provided with the property of interest; they do not require the chemical analogue/ scaffold and substituents (R-groups) to be predetermined. An example of how the Free-Wilson analysis is calculated for one R group is demonstrated in Figure 1-9. The equation can then be adapted for more R groups around a scaffold, the subsequent summed R groups would be added to the end of the equation.



$$BA = \mu + a_1X_1 + a_2X_2 + \cdots + a_iX_i$$

$$BA = \mu + \sum_{i=1}^{R_1} a_iX_i$$

New Molecule	R_1 Derivatives			
	X_1	X_2	…	X_i
1	0	1	…	0
2	1	0	…	0
3	0	0	…	1
…	…	…	…	…

Figure 1-9: Free-Wilson analysis example for one R group. Where μ is the contribution of the parent analogue, a_i is the activity contribution of R group substructural fragment i and Xi is whether the R group i is present, 1, or absent 0

One of the drawbacks of this methodology is that it can only provide predictions for R-groups that have previously been seen and therefore does not allow any exploration around certain substitution sites. Additionally, predictions for new molecules can only occur if the new molecule identifies with any of the scaffolds extracted within the dataset. Free-Wilson approach is limited to the local analysis of homogeneous datasets, however, other methods have been developed to overcome this limitation.

1.4.1.1.2 Interpretation of QSAR

An important aspect in chemoinformatics is to understand the relationships in QSAR models. There have been several approaches to attempt to make them interpretable and provide information on which atoms or group of atoms are important to the property of interest.

The first is matched molecular pair analysis (MMPA) that identifies SAR patterns through defining matched molecular pairs (MMP) that are present. A MMP is a "well-defined structural change" between two molecules, Figure 1-10. Kenny and Sadowski first established this concept (Kenny & Sadowski, 2005). SARs can be identified as chemical pair transformations that can cause a change in ADMET properties or target binding. MMPs are identified and differences in their measured ADMET data are used to link properties to structures. An example of an MMP is changing a methyl group to a fluorine atom (Griffen, Leach, Robb, & Warner, 2011; A. G. Leach et al., 2006).

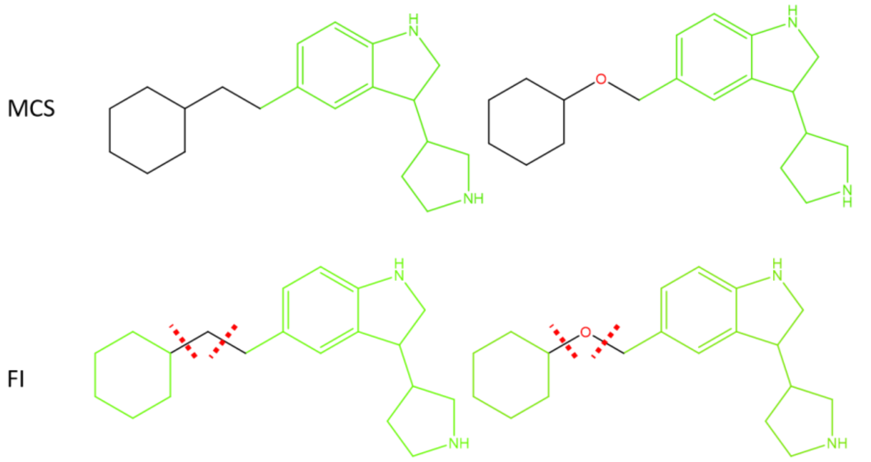


Figure 1-10: Comparison of MCS and FI MMP methods, MMP highlighted in green

The most common way of finding an MMP is via the fragment-index (FI) method, which can be divided into several steps. The first step is to fragment each molecule by cleaving bonds. Hussain and Rea first described this method by cleaving every acyclic single bond between two heavy atoms. The bonds are broken one at a time and the resulting fragments are stored as SMILES, one is the core, that is the fixed fragment, while the other is the fragment that changes (Figure 1-10) (Hussain & Rea, 2010). An alternative approach is based on finding the MCS. The main drawback with MMPs is that it depends on the implementation as using the two different techniques can result in different MMPs, see Figure 1-10.

Polishchuk et al. developed a method that masks a fragment of interest and compares the predictions of this masked molecule and the original unmasked molecule, to provide information about the masked atom(s). Two different descriptors were used and four different models to generate a consensus score (P. G. Polishchuk, Kuźmin, Artemenko, & Muratov, 2013; P. Polishchuk et al., 2016). Further work, was also done by Polishchuk that took the environment of the atom(s) into account too (Matveieva, Cronin, & Polishchuk, 2019). This method will be further described in Chapter 7.

Sheridan has recently used similar principles to generate a method that colours the atoms according to how much they contribute to the activity (Sheridan, 2019). Sheridan uses a combination of molecular descriptors, in frequency form, and ML models to establish a consensus prediction from all combinations. To understand the importance of each atom a recalculated prediction is compared to the original prediction, when the atom is changed to a sodium, Na, atom. However, Sheridan showed that this is not always possible, as these different combinations are not always in agreement with the importance of each atom or even the ordering of the importance of the atoms in the molecule.

1.4.1.2 Modelling Algorithms

1.4.1.2.1 Decision Tree

Decision trees are a form of machine learning technique that can also be used to predict discrete properties/ labels (Wawer, Lounkine, Wassermann, & Bajorath, 2010). Decision trees have been used across a wide range of chemical endpoints to successfully predict or interpret relationships or both (Agrafiotis, Shemanarev, Connolly, Farnum, & Lobanov, 2007).

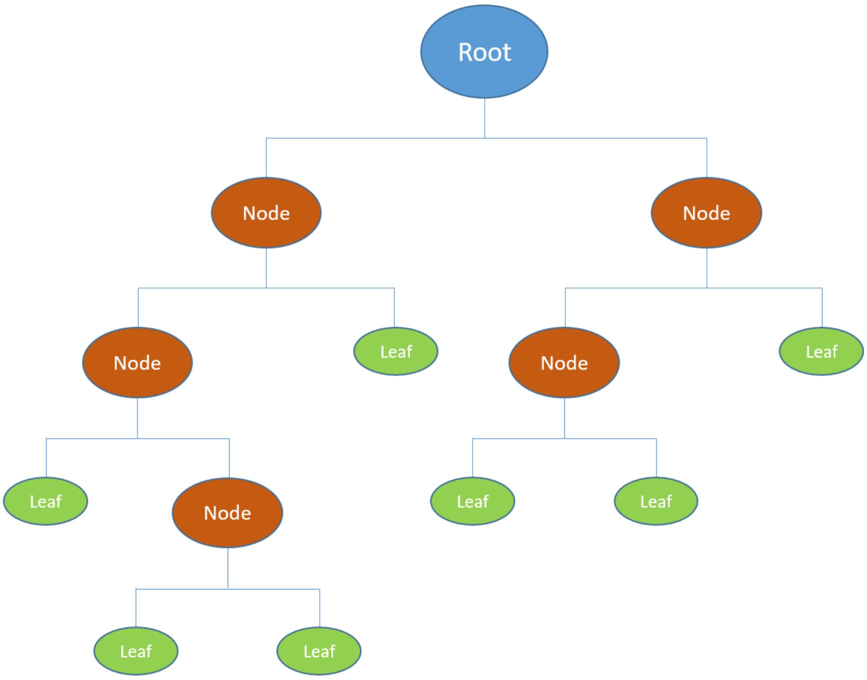


Figure 1-11: Decision Tree

An example of a decision tree is shown in Figure 1-11. The root node represents the whole dataset, which is split into nodes based upon some criteria or property. The process is repeated until eventually the outcome is a leaf which is the prediction/ label. In order to split nodes different metrics can be used to establish the best split to keep alike molecules together. Common metrics that are used are Gini impurity, information gain (entropy), and variance reduction.

1.4.1.2.2 Random Forest

The decision tree is prone to overfitting the data whereby it learns the training data but is not accurate in predicting new data. One way in which the decision tree can be enhanced is to create a collection of decision trees into a random forest (RF) (Ho, 1995, 1998). RFs can help to overcome low predictivity and instability of a model. RFs are stable models and are resistant to noise within the data. This is because it uses a combination of decision tree models, a prediction is based on the predictions made across multiple trees and each tree is built using a subset of descriptors/ data. An example of how a RF works is shown in Figure 1-12. A RF is built following three rules. The first is that each tree is generated from a random sample of the training set; by repeating this a number of times, the whole of the training set will be sampled. The second is that when splitting nodes only a defined number of random features are used. The third is that the trees cannot be pruned, i.e. be reduced (P. Polishchuk, 2017).

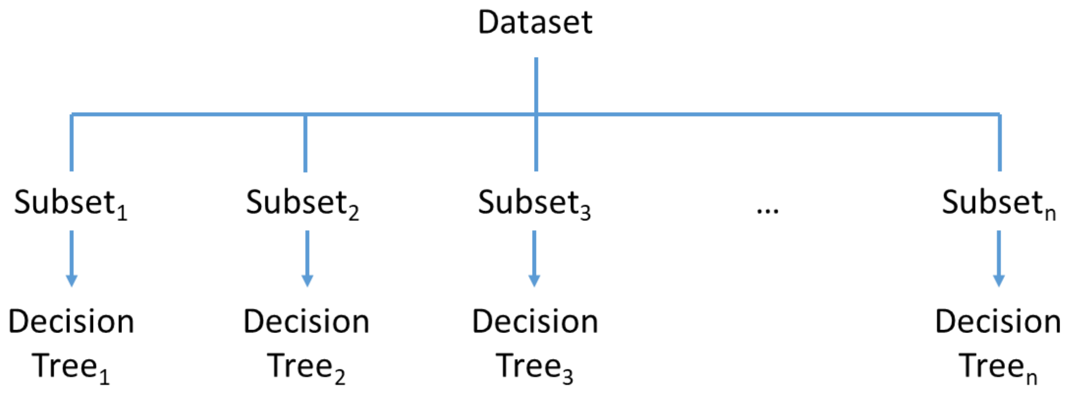


Figure 1-12: An example of how a random forest works

1.4.1.2.3 Gradient Boost Model

Gradient boost model (GBM) is another ensemble approach based on decision trees that has been used in chemoinformatics for regression and classification problems (P. Polishchuk et al., 2016; Sheridan, Wang, Liaw, Ma, & Gifford, 2016). GBM sequentially builds weak models, where a weak model is one that is considered to be slightly better than random meaning that it has poor accuracy. This, ultimately, improves the predictivity power since it has less bias and variance as each model learns from the previous one. An illustrated example of how a boosted system works is displayed in Figure 1-13. When moving on to the next decision tree larger weights are given to the incorrectly predicted observations. The next decision tree amplifies these incorrectly predicted observations to minimise the loss function and reduce the errors.

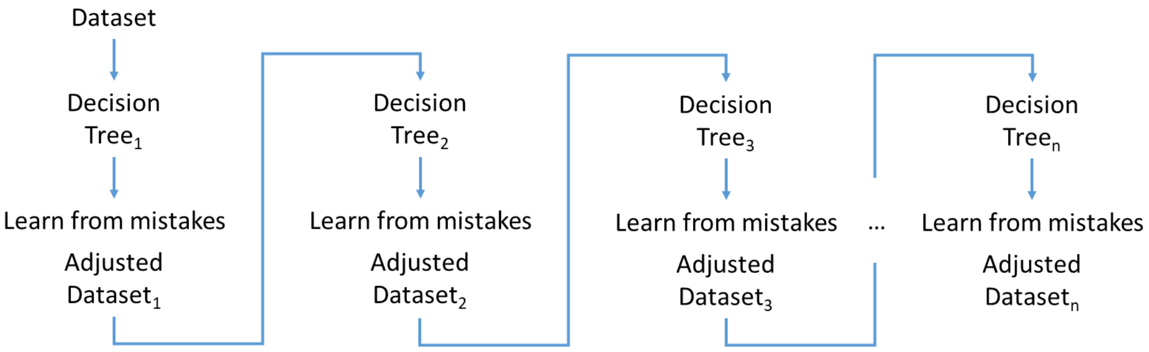


Figure 1-13: An example of a boosted system

1.4.1.2.4 Support Vector Machine

The support vector machine (SVM) is a supervised technique that identifies a hyperplane in regression and classification problems. They were initially introduced to the area of chemoinformatics for binary classification (Burbidge, Trotter, Buxton, & Holden, 2001) and were later used on regression problems (Alvarsson et al., 2016).

SVM uses a kernel function to create a hyperplane, also known as a decision boundary, that for a classification problem separates the data into two different classes. The data points on either side of the hyperplane are defined as the support vectors. The optimum position of the hyperplane is when the distance between the support vectors is maximised, which is known as the margin, and which minimises the classification error (H. Chen et al., 2013). An example of a classification problem and how a SVM model works can be seen in Figure 1-14. This has two classes, 1 labelled in red and 2 labelled in blue which contain two features, x and y.

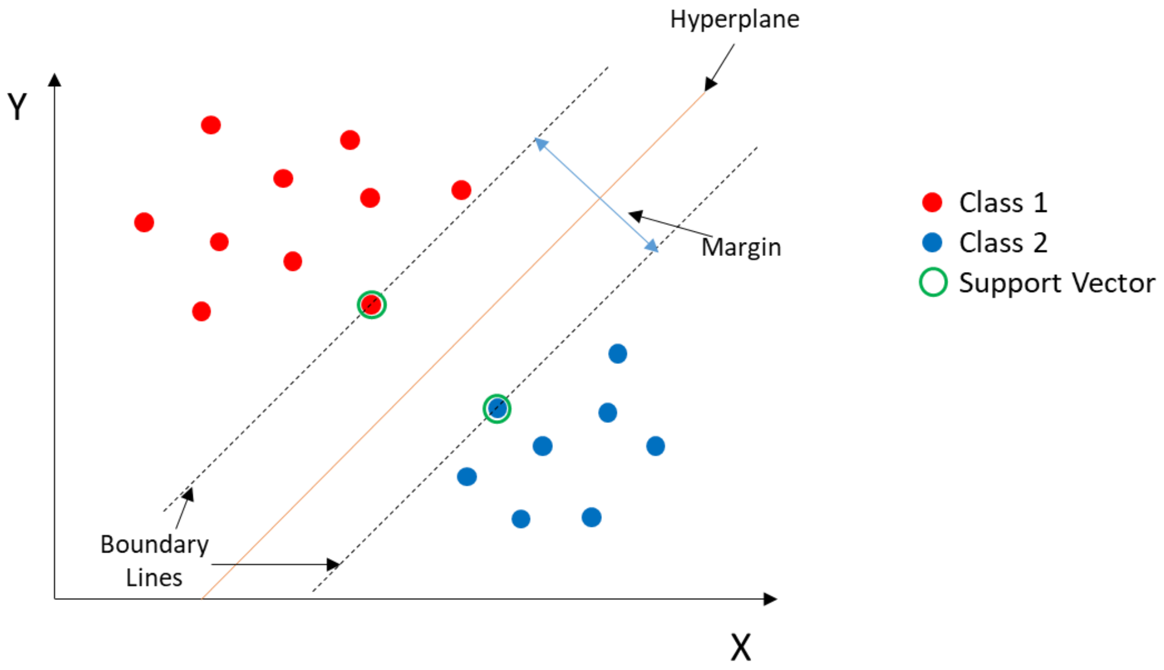


Figure 1-14: Illustration of how an SVM model works

Vapnik further developed the SVM method so that it could be useful for regression problems (Vapnik, 1995). For a regression model, instead of the hyperplane acting as a boundary between classes, it instead helps to establish a prediction value for the property of interest. Within a support vector regression (SVR) model an ε-insensitive region is established, this region controls the noise of the data within the model, Figure 1-15. Ideally all data points are within this region, however, this is not always possible to have a relevant worthwhile model, so the data points that lie outside this region have an associated error attached, ξ_i, which increases as the distances from the ε-insensitive region increases (Ivanciuc, 2007). A hyperplane can then be fitted using these data points allowing an estimate y value to be provided for any given x value.

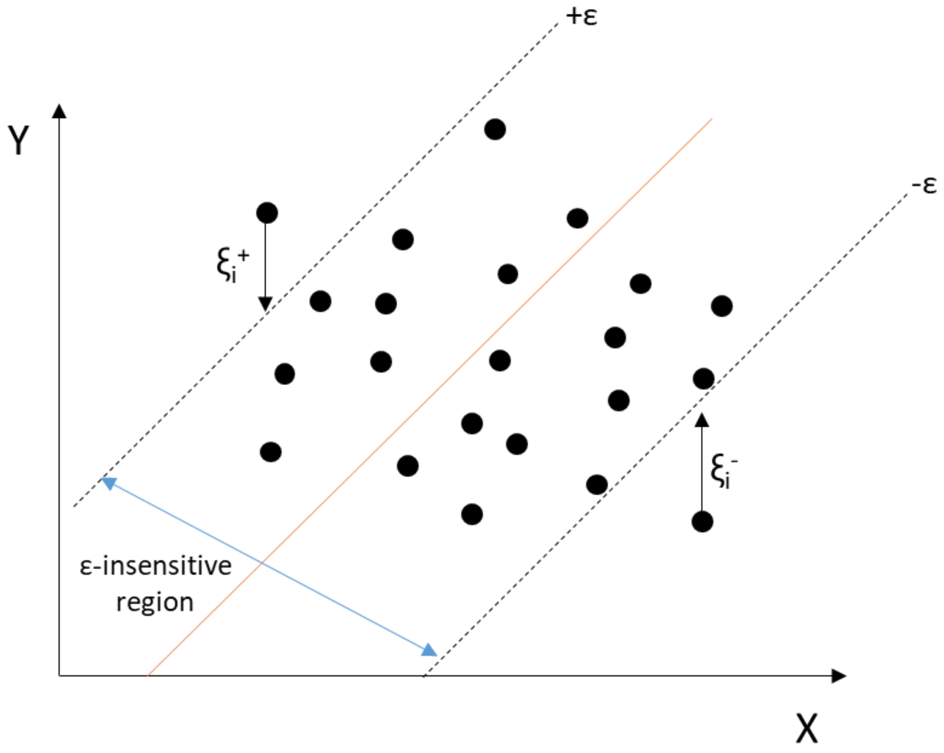


Figure 1-15: SVR example

Not all examples are as straightforward as in Figure 1-14 and a clear hyperplane may not be identifiable. A hyperplane may be more easily identifiable when the data is projected into higher dimensions which is done through a kernel function. A common kernel function is the radial basis function (RBF). RBF is a Gaussian function φ:

$$\varphi = e^{\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)} \tag{1.3}$$

Where x and y are vector points and σ controls the shape of the hyperplane. The squared distance between these two points is calculated. γ, $\frac{1}{2\sigma^2}$, scales the amount of influence two points have on each other (Ivanciuc, 2007). It is commonly used as it works in infinite dimensions and is efficient.

1.4.1.2.5 Partial Least Squares

Partial least squares (PLS) regression is based on the upon principal component analysis (PCA) and multiple linear regression. PLS is a technique that can deal with the large number of independent variables that are typically used to represent molecules. A PLS model provides a technique for correlating information in one data matrix, X, to the information within another matrix, Y, where X are the independent variables and Y is the dependent variable(s). Whereas, PCA is just the analysis of one data matrix, i.e., X, which can then be visualised, see below.

The first step of a PLS is to first reduce the number of independent variables to a smaller set of uncorrelated components. A linear regression can then be performed on these components to allow the dependent variable, the property of interest, to be estimated. These two steps can be explained via the following equations:

$$X = TP^T \tag{1.4}$$

$$\hat{Y} = TBC^T \tag{1.5}$$

X is the original variable matrix which is made up from of a set of T scores and P their associated loadings and X-residuals, E. The set of T scores can then be combined with the regression weights, B, and the weight matrix of X, C, to provide an estimate of the Y value, Ŷ (Abdi, 2010).

The way in which the PLS is developed means that the first components encode the most variation. As a result, PLS can reduce high dimensional data to a significantly smaller number of variables, which contains the largest amount of information for that original X matrix. This large reduction allows them to be useful predictors for chemical properties (B. Chen, Zhang, Bond, & Gan, 2015; P. Polishchuk et al., 2016).

1.4.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning is applied to unlabelled data. The main use of unsupervised learning in chemoinformatics is to model molecular properties and biological activity in order to learn more about the data. Examples of this are principal component analysis (PCA) and clustering (Clarke et al., 2008).

1.4.2.1 Principal Component Analysis

PCA is a dimensionality reduction technique based on using linear combinations of the descriptors. However, with the reduction of the dimensions, there is an expected loss of information (Medina-Franco, Martínez-Mayorga, Giulianotti, Houghten, & Pinilla, 2008; Wawer et al., 2010).

The principal components are calculated from an original matrix of n rows, where n is the number of molecules, and p columns, where p is the number of descriptors, through a variance-covariance matrix. The variance-covariance matrix is of size of n x n. The eigenvectors represent the directions of lines, which can be drawn through the data points. The eigenvalues are the amount that the data points vary from this eigenvector. The eigenvector with the largest eigenvalue is known as the principal component, this means that it is the one with the largest variance. The number of eigenvectors and values that are considered following a PCA analysis determines the number of dimensions used to represent chemical space. Therefore, if the model only has two dimensions then the two largest eigenvalues are used. The principal components selected are all orthogonal to each other (A. R. Leach & Gillet, 2007; Medina-Franco et al., 2008). As variables with high dimensions can be reduced to two or three variables, principal components, the PCA can be visualised in a plot form, see Figure 1-18. A disadvantage of this visualisation technique is that the principal components are linear combinations of the original descriptors which can be hard to interpret and that it is not always possible to condense information into a small number of key components to allow patterns to be identified (Bro & Smilde, 2014).

1.4.2.2 Clustering

Clustering allows objects to be divided into different groups or clusters, where objects in the same group are the similar and objects in different groups are different. From a chemoinformatician's point of view, this may enable relationships between chemotypes and biological activity to be identified (Gillet et al., 2015; Wawer et al., 2010). There are three main steps to clustering: the first step is to establish each molecule's set of features; the second step is to calculate pairwise similarities between the molecules; the third and final step is to cluster together similar molecules (Willett, 2005).

There are two main types of clustering methods which are hierarchical and non-hierarchical methods. Both clustering methods generate distinct or crisp clusters; therefore, none of the structures can appear in multiple clusters, other than the nested clusters within a hierarchy of clusters (R. D. Brown & Martin, 1996). There is another type of clustering known as fuzzy clustering that allows the objects to belong to more than one cluster. This arose in popularity due to its unique ability to be able to have an item in multiple clusters, which in some instances can be useful (Bunin, Siesel, Morales, & Bajorath, 2007).

There have been several areas in the chemoinformatics area that use clustering, such as: SAR, virtual screening (Gupta & Zhou, 2021), high-throughput screening (Harper, Bravi, et al., 2004; Reymond & Awale, 2012), visualisations (Gütlein, Karwath, & Kramer, 2012), scaffold analysis (Mok & Brown, 2017) and molecular docking (Makeneni, Thieker, & Woods, 2018).

1.4.2.2.1 Clustering Techniques

1.4.2.2.1.1 Agglomerative Clustering

Agglomerative clustering is a type of hierarchical clustering that builds clusters from the bottom up. Therefore, each molecule begins on its own and the molecules come together based upon a certain criterion. The merging of clusters is repeated until all molecules are in a single cluster. There are three main implementations of agglomerative hierarchical clustering: group-average, single-link and complete-link. In each case, the two clusters are combined that have the smallest index. The three indexes are illustrated in Figure 1-16 by different colours. The four resulting clusters are demonstrated with a dashed line. The group-average is the distance between the cluster's averages, shown by the red arrows. The single-link index is the minimum distance between any two molecules where one is taken from each cluster, shown by the green arrows. The complete-link is the maximum distance between any two molecules where one is taken from each cluster, shown in by the yellows arrows. A clustering level is chosen based on either a distance criterion or a specific number of clusters (R. D. Brown & Martin, 1996; Murtagh, 1985).

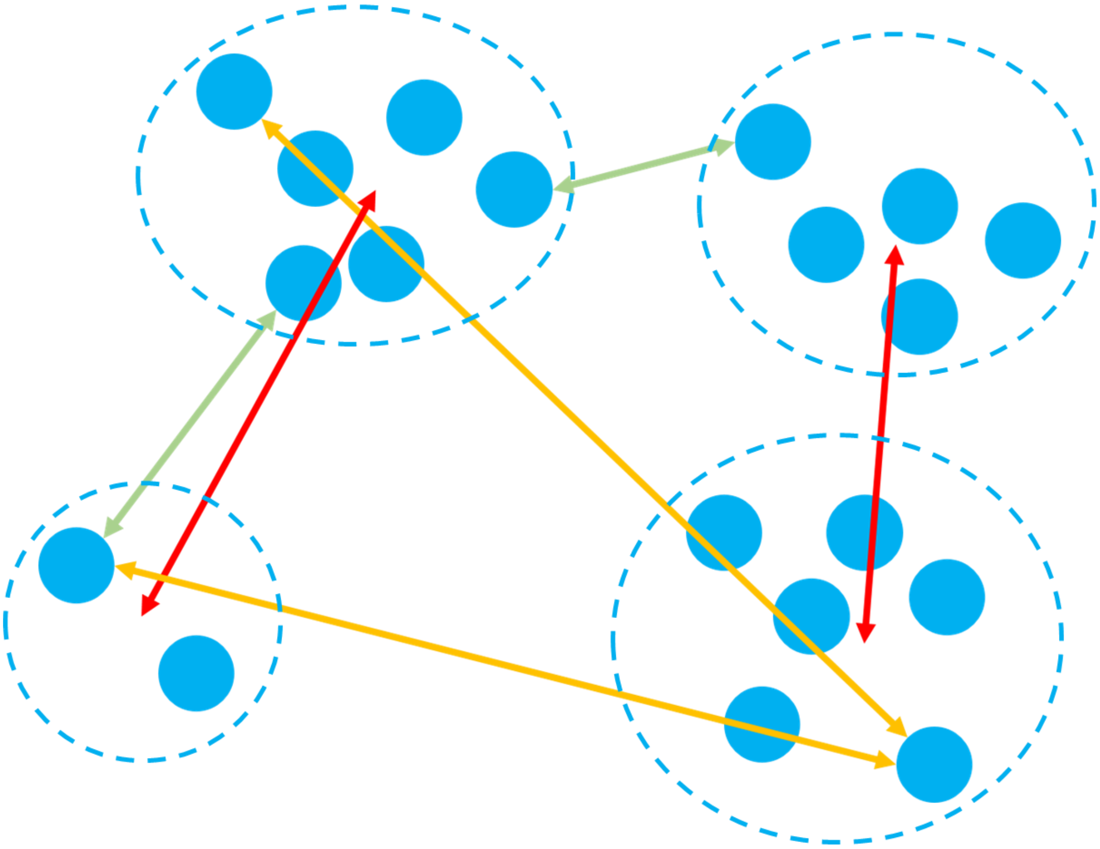


Figure 1-16: An example of agglomerative clustering, the red arrows indicates the average distance algorithm, the green arrows indicates the single-link algorithm and the yellow arrows indicates the complete-link algorithm.

1.4.2.2.1.2 Butina Clustering

Butina clustering, also known as Taylor-Butina as it was first described by Taylor et. al (Taylor, 1995), consists of two main steps. The first step is to identify potential centroid molecules. These are the molecules that the clusters will be built around and will, therefore, be the centres of the clusters. For each molecule, the number of near neighbours is calculated using a specific Tanimoto threshold. The molecules are then ordered so that the molecule with the highest number of neighbours is first and the molecule with the fewest is last. This ordering eliminates issues of order dependencies that are seen with some other clustering techniques. The first molecule is then selected as the centroid of the first cluster and all the molecules that are above the similarity threshold become part of this cluster. These molecules are removed from the list to ensure that a molecule is only assigned to a single cluster. The next remaining molecule is then chosen from the list and the process is repeated until all molecules are within a cluster or are a cluster on their own (Butina, 1999). An example of this clustering technique can be seen in Figure 1-17 where the red circles represent centroid molecules, the lighter blue larger circles represent the Tanimoto threshold, and the small darker blue circles represent molecules that fall within the Tanimoto threshold of the respective centroid molecule.

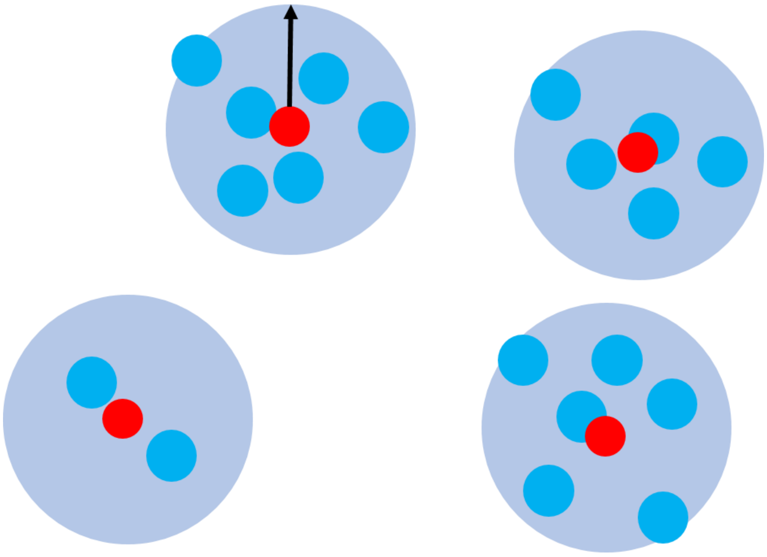


Figure 1-17: Example of Butina clustering

1.4.2.2.1.3 K-Means Clustering

The K-means algorithm also uses centroids, as for Butina clustering, however, the number of centroids (and therefore clusters) is pre-defined as k. The centroids are initially chosen at random and each molecule is assigned to its nearest centroid. For each cluster, a new centroid is calculated as the centre of the compounds within the cluster. Each molecule in the dataset is then reassigned to its nearest centroid. The process is repeated until there is no change of molecules to clusters (Bora & Gupta, 2014).

1.4.2.2.2 Cluster Validity

Different clustering techniques typically produce different clusterings of molecules depending on the input parameters and the algorithm used. Therefore, once clusters have been established, it is important to check how valid the clusters actually are. Cluster validity is determined by calculating various scores that aim to quantify how good the clustering is. There are two criteria that are typically used. These are the compactness of the clusters and the separation between clusters. There are three types of indexes: external, internal and relative. External indexes are based on how well matched the clusters are to known specified classes or labels. This is useful if the ideal clustering of the data is known. However, if this is unknown, which is usually the case, then this index cannot be used without self-assigning all the data points. Internal indexes are based on the data itself and do not make reference to any pre-existing cluster information. These indexes are typically based on calculating inter- and intra-cluster distances. Relative indexes compare multiple clusters, via either an external or internal index (Agrawal, Garg, & Patel, 2015). There are three different ways of calculating the inter-cluster distances. The first is the single linkage method which uses the same principle as the agglomerative clustering single linkage and is the minimum distance between clusters, that is, the distance between the closest two data points. The second is complete linkage which is the maximum distance between clusters, that is, the distance between the furthest two data points. The third is the distance between the centroids of each of the clusters. Single linkage and complete linkage involve pairwise similarity calculations and can, therefore, have large computational costs (Azuaje & Bolshakova, 2002; Halkidi, 2001).

1.4.2.2.2.1 Silhouette Average

Silhouette average provides information on the cohesion and compactness of the clusters while also providing information on the separation. A silhouette average is a value between -1 and +1 with the closer the value is to +1 the better the clustering. A, *s*, is calculated for each data point/molecule as shown in Equation 1.6.

$$s(i) = \begin{cases} 1 - \frac{a_i}{b_i}, & a_i < b_i \\ 0, & a_i = b_i \\ \frac{b_i}{a_i} - 1, & a_i \geq b_i \end{cases} \tag{1.6}$$

$$silhouette\ average = \frac{1}{m}\sum_{i=1}^{m} s(i) \tag{1.7}$$

Where a_i is the average distance of point *i* to all points within its own cluster, b_i is the average distance of *i* to all points in the nearest neighbouring cluster and m is the number of data points. The closer the silhouette score is to one the better matched a data point is to its own cluster, which represents good cohesion, and the more poorly it matches to the other clusters, representing good separation. When a data point is in a cluster on its own, i.e., a singleton, then that cluster's silhouette score becomes zero as this is the most neutral score to give it as zero sits directly in the middle between -1 and +1 (Rousseeuw, 1987).

1.4.2.2.2.2 Dunn Index

The Dunn index is also based on inter- and intra-cluster distances.

$$D = \min_{i=1 \ldots n_c} \left\{ \min_{j=i+1 \ldots n_c} \left(\frac{d(c_i, c_j)}{\max\limits_{k=1 \ldots n_c} \left(diam(c_k) \right)} \right) \right\}, \tag{1.8}$$

$$where\ d(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x, y)\}$$

$$and\ diam(c_k) = \max_{x, y \in c_i} \{d(x, y)\}$$

Where *d(c$_i$, c$_j$)* is the distance between clusters c_i and c_j and *max{d(x,y)}* equates to the maximum distance between point *x* and *y* that are within the same cluster, c_k. The more compact and more separated clusters are the higher the Dunn index (Dunn, 1974; Halkidi, 2001). A disadvantage of the Dunn index is that it can be computationally expensive for large datasets and it is sensitive to noise.

1.4.2.2.2.3 Davies Bouldin Index

Davies Bouldin (DB) index is the average of the ratios between the intra-cluster distance and the separation between clusters for each cluster. This means that the Davies Bouldin must be positive. There are several properties that should be conserved. The main one is that the ratio between *i^th^* and the *j^th^* cluster must be greater than or equal to zero, whilst also maintaining that the ratio R_{ij} equals R_{ji}.

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \quad d_{ij} = d(v_i, v_j) \quad s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i) \tag{1.9}$$

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i, where\ R_i = \max_{j=1 \dots n_c, i \neq j} (R_{ij}), \qquad i = 1 \dots n_c \tag{1.10}$$

Where s_i is the intra-cluster distance, d_{ij} is the inter-cluster distance and n_c is the number of clusters. This index generally suggests that the lower the value of DB then the better the cluster separation and the more compact the clusters are (Davies & Bouldin, 1979).

1.4.2.2.2.4 Calinski-Harabasz

The Calinski-Harabasz (CH) index is the ratio of the average intercluster distance and the average intracluster distance.

$$CH = \frac{between\ sum\ of\ squares\ /(n_c - 1)}{within\ sum\ of\ squares\ /(n_o - n_c)} \tag{1.11}$$

$$between\ sum\ of\ squares = \sum_{k=1}^{K} n_k \|z_k - z\|^2 \tag{1.12}$$

$$within\ sum\ of\ squares = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \|x_i - z_k\|^2 \tag{1.13}$$

Where n_o is the number of data points, n_c is the number of clusters, n_k is the number of points within cluster *k*, z_k is the centroid of cluster *k*, *z* is the centroid of the overall points and x_i is a data point within cluster *k*. Unfortunately, due to the nature of intra-cluster sum of squares, it examines all possible grouping of points which means that it can become very computationally expensive (Calinski & Harabasz, 1974). A larger CH index is preferred as this demonstrates a larger cluster separation relative to the cluster compactness (Jauhiainen & Kärkkäinen, 2017).

1.4.2.2.2.5 Kelley Index

The Kelley index is different from the other indexes as this index penalises clusters that contain a large number of singletons as these are not desired.

$$(n-2)\left(\frac{\bar{d}_{wl}-\min(\bar{d}_w)}{\max(\bar{d}_w)-\min(\bar{d}_w)}\right)+1+k_l \tag{1.14}$$

Where *n* is the number of points, k_l is the number of cluster, $\bar{d}_{wl}$ the mean of the intra-cluster distances and min($\bar{d}_w$) and max($\bar{d}_w$) are the minimum and maximum of the intra-cluster distances. The singletons are excluded from this calculation, therefore, this equation penalises where there are a large number of singletons. The better the clustering the lower the value of the Kelley index (Kelley, Gardner, & Sutcliffe, 1996).

1.4.2.2.2.6 Ball-Hall Index

Ball-Hall (BH) index is one of the indexes that only takes into account the intra-cluster distances.

$$BH=\frac{1}{K}\sum\nolimits_{k=1}^{K}\frac{1}{n_k}within\ sum\ of\ squares^{\{k\}} \tag{1.15}$$

Where *K* is the number of clusters and n_k is the number of data point in the k^{th} cluster. The desired level is known as the elbow. This means if all BH points are plotted against number of clusters. Then the turning point in the graph is the number of clusters that should be chosen (Ball & Hall, 1965).

1.5 Visualisation of Chemical Space

Many different visualisation techniques have been used to help medicinal chemists easily interpret and identify relationships between chemical structures and activities. They can also be important in suggesting chemical space to further explore or exploit and display a large amount of data without any bias. This can either be to focus on one compound or to explore a larger set of compounds that have the potential to be biologically active (Kayastha, Kunimoto, Horvath, Varnek, & Bajorath, 2017; Wawer et al., 2010; Wollenhaupt & Baumann, 2014). Additionally, visualisation is crucial as it allows complex problems to be simplified to aid the decision-making process. It can provide information on which molecules to prioritise, while also providing information on how diverse a chemical series is, along with key SAR analysis with ADMET properties. Unfortunately, due to the complex nature of chemical space, which is frequently multi-dimensional, it is challenging to represent it in fewer dimensions to make it human readable (Medina-Franco et al., 2008).

Four common uses of visualisations were identified by (Stumpfe & Bajorath, 2016) are shown in Figure 1-18. Firstly, is plotting the compounds in a low dimensional space. Second, they can be used to organise data according to common substructures to, for example, indicate the significance of R-groups around a scaffold. Third, they can be used to indicate how data can be organised through the clustering and partitioning. Finally, they can be used to relate chemical structures to properties of interest, typically active values.

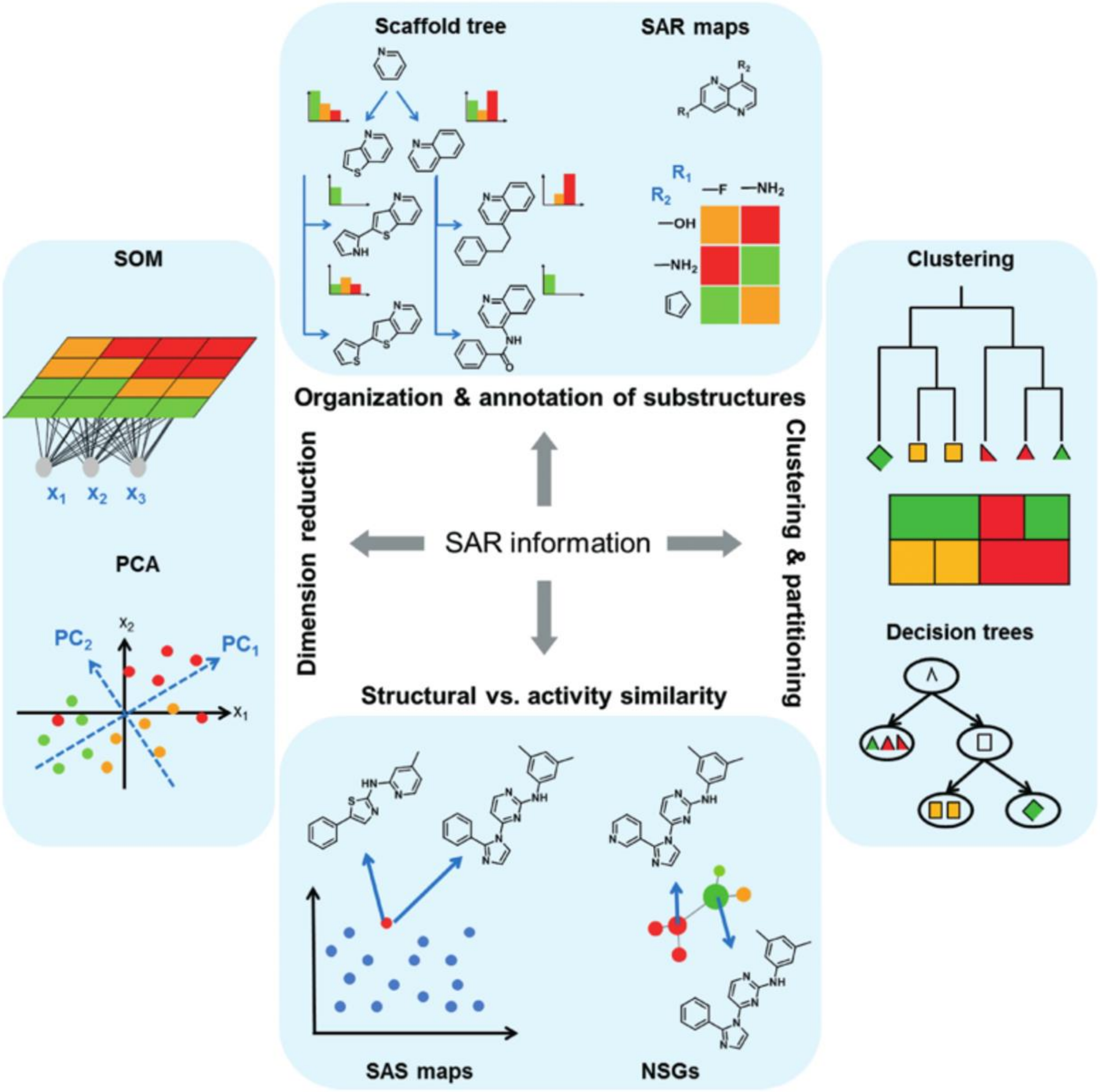


Figure 1-18: Different visualisation approaches (Stumpfe & Bajorath, 2016)

1.5.1 Generation of Structure-Activity Relationship Tables

A unique way for chemists to display their data is in a structure-activity relationship (SAR) table. SAR tables are a way of uniformly representing chemical data within a table. This is done so the medicinal chemists can easily interpret and analyse the chemical space that has been explored.

1.5.1.1 Traditional SAR tables

A traditional SAR table is also known as a R-group table. Each row represents a chemical structure and the columns represent different R-groups with the corresponding chemical substitution in the cell. The final column represents the biological activity value of the molecule. These tables have been used for many years as chemists find them easy to comprehend. However, they can be time-consuming to generate and also difficult to fully and critically analyse (Agrafiotis, Shemanarev, et al., 2007).

Traditional SAR tables were generated manually however they are then subjective and different chemists may organise the data differently using a different core scaffold and different definitions of the R groups. This could have a large impact if the tables are analysed automatically or are fed into another system as the different variations could lead to different results. This type of R-group table is also not able to deal with large datasets (Wollenhaupt & Baumann, 2014). Different methods are being developed to overcome these disadvantages.

1.5.1.2 SAR Matrices

SAR matrices were developed in 2007 by Agrafiotis et al. with the main aim to help medicinal chemists see relationships within their data algorithmically without the need of a chemist's input. A SAR matrix is a table where the columns are different R groups and the rows are different cores such that each cell is a unique molecule that can be coloured according to its biological activity, or physicochemical property that is being investigated. This can easily allow the medicinal chemist to identify areas of chemical space that have not been investigated, see Figure 1-19 (Agrafiotis, Shemanarev, et al., 2007). These matrices are generated through a mixture of MMP analysis and clustering. One cluster can easily be seen in Figure 1-19, the highlighted molecule is the combination of the core (row) and the R-group (column) (Stumpfe & Bajorath, 2016). Further work has been done recently, which constructs the SAR matrix whilst also being useful in molecular design as it provides predictions for neighbouring unexplored blank molecules (Yoshimori & Bajorath, 2020).

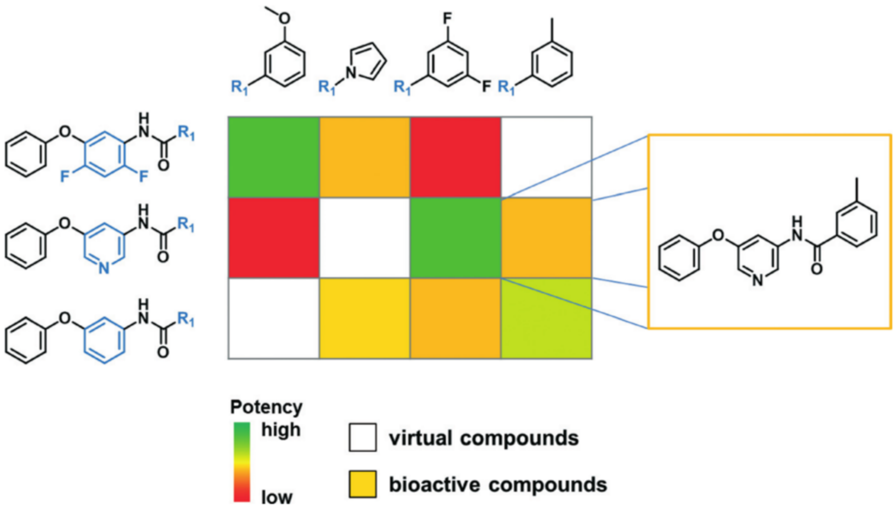
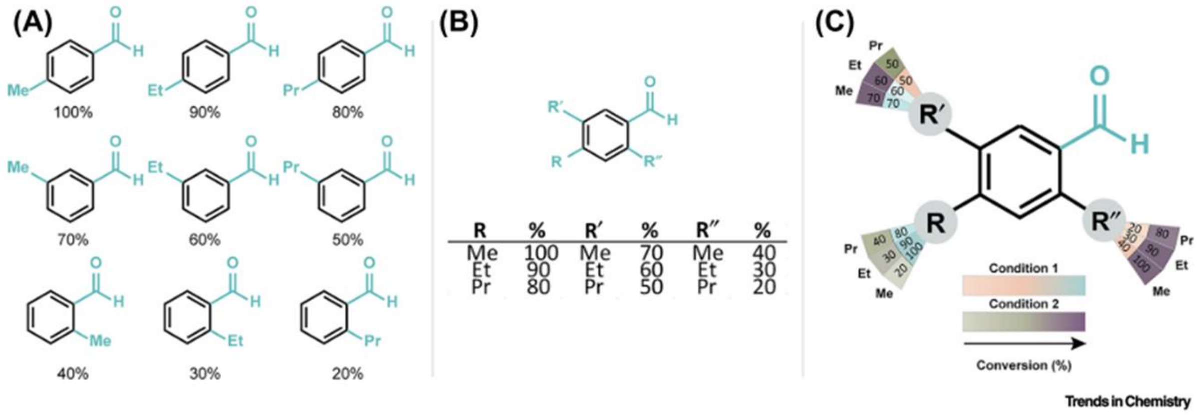


Figure 1-19: SAR matrix (Stumpfe & Bajorath, 2016)

1.5.1.3 Radial Scope Plot

Recently, there has been a new visualisation to display structure-data relationships, the radial scope plot illustrated in Figure 1-20c. This attempts to combine a scaffold structure and a coloured heatmap of properties of the R-groups (Rodríguez Benítez, Dürr, & Narayan, 2020). Heatmaps will be discussed further later. These radial scope plots attempt to demonstrate the R-groups that impact the conditions that are of interest.



R	%	R′	%	R″	%
Me	100	Me	70	Me	40
Et	90	Et	60	Et	30
Pr	80	Pr	50	Pr	20

Figure 1-20: A) Traditional substrate scope plot, one core with multiple R-groups. B) A table representation of the substrate information. C) Radial scope plot for two different conditions. (Rodríguez Benítez et al., 2020)

1.5.2 Structure-Activity Similarity Maps

Structure-activity similarity (SAS) maps were first introduced by Maggiora et al. in 2001. SAS is a scatterplot representation that represents pairwise structure molecular similarities and molecular activities, see Figure 1-18. This visualisation differs from the SAR matrix as each point in the scatterplot is a molecular pair rather than a singular molecule. Figure 1-21 shows the four major regions that are associated with SAS maps. These regions help to identify where there is a high/low structure similarity and/or a high/low activity difference. Region II is a region with smooth SAR where there is a high molecular similarity and a low activity difference. Region IV is where the activity cliffs are as they contain high molecular similarity and high activity difference. Region I is the scaffold hopping region which is important and this is located across from the activity cliffs. Region III is an uninteresting region (Saldívar-González, Naveja, Palomino-Hernández, & Medina-Franco, 2017).

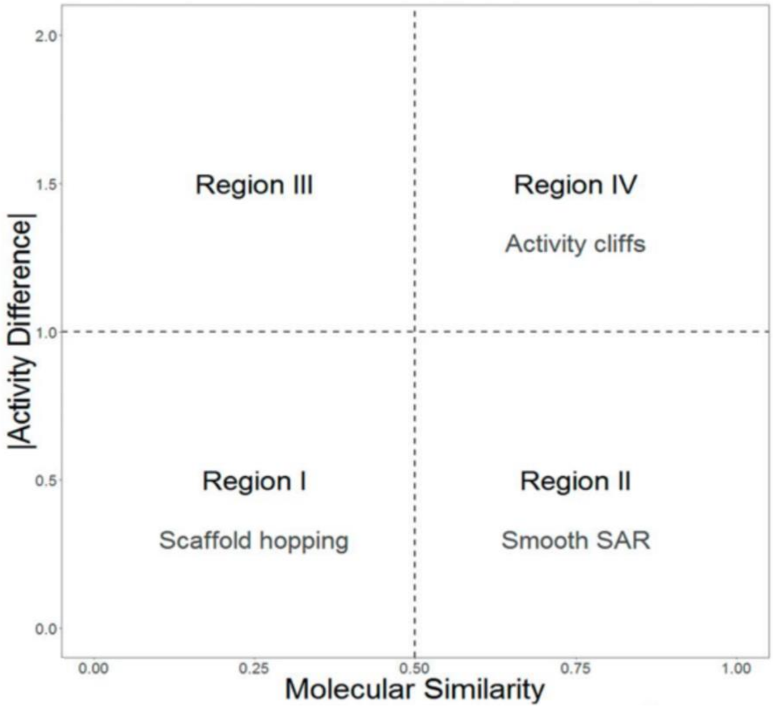


Figure 1-21: SAS map showing the four major regions (Saldívar-González et al., 2017)

The regions in SAS maps are hard to define and vary from task to task. Additionally, SAS maps are limited to small datasets due to the plotting of N^2 data points, where N is the number of molecules. There are several different variations of the SAR maps to help to overcome these issues. One example is density SAS maps which contain a heat map based on the number of points within that region, therefore, giving the frequency of the data points within that area (Saldívar-González et al., 2017).

1.5.3 Networks

Chemical space networks were originally developed in 2014 by Maggiora et al. In a chemical space network, each molecule is represented as a node and the edges connect two molecules if their pairwise similarity is above a certain threshold. The network structure represents the chemical space by capturing the discrete pairwise similarity of molecular structures rather than just the molecules themselves. This means that it does not suffer from the 'curse of dimensionality'. In addition, it allows easy analysis of chemical space through algorithms developed for network analysis. However, there is one universal issue that it fails to address, the visualisation can change depending on how the molecular representation is defined (G. M. Maggiora & Bajorath, 2014)

Networks provide a good interpretation of chemical space. Networks allow us to establish activity cliffs where a slight change in chemical structure, whilst retaining a similar functionality (same RG), can have a huge impact on the biological activity (Wollenhaupt & Baumann, 2014). Like many other visualisation techniques, network maps can be nicely complemented with the use of colour, as it allows a user to quickly and easily establish relationships between structure and activity. Two methods that are of interest are InSARa and SARANEA, as they use networks to display chemical information.

1.5.3.1 InSARa

InSARa is a chemical space network in which the underlying molecular representation is the RG. This is a visualisation technique that exploits MCS. The pairwise MCSs are found which consequently allows a root MCS to be found. The molecules are then matched to which root MCS they belong to. The network consists of a root node as the smallest RG-MCS of the cluster. This is then expanded out to more nodes, which contain subsets of the cluster represented by larger RG-MCS. Then, the molecules are placed onto the network attached to the largest possible RG-MCS of a cluster, see Figure 1-22. This visualisation allows easy identification of activity cliffs, pharmacophoric features of the chosen target and SAR hotspots (Wollenhaupt & Baumann, 2014). If a RG-MCS node has all red coloured attached nodes bar one being green, then this would demonstrate an activity cliff. Whereas a SAR hotspot would be if there were multiple occurrences of all colours at a single RG-MCS node.

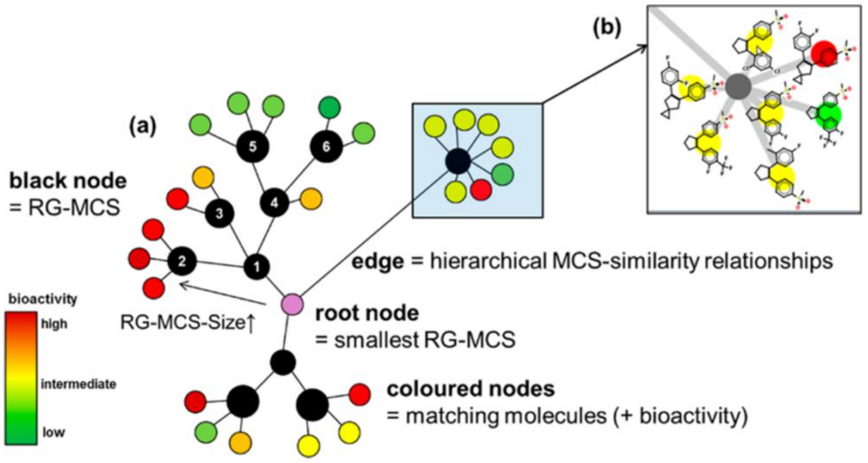


Figure 1-22: a) A prototype of the inSARa network. b) Import the chemical structures using Cytoscape with the chemViz plugin (Wollenhaupt & Baumann, 2014)

1.5.3.2 SARANEA

SARANEA is a network similarity graph that explores a dataset's structure-activity relationship (SAR) and structure-selectivity relationship (SSR). Due to the structure and purpose of the SARANEA visualisation, it gets its name SAR due to the exploration of SAR analysis within this work and ANEA comes from araneae the scientific term for spider's webs which refers to the likeliness of network graphs to spider webs.

A node within the network represents a molecule within the dataset; the node's colour depends on the bioactivity of that molecule. The edges demonstrated between the nodes are if the two connected molecules have a structural similarity above a predefined threshold. Additional information is provided in the form of two numerical scores. A discontinuity score estimates the contribution from an individual molecule to the disagreement with the datasets SAR. A cliff index indicates how far away the molecule is from the activity cliff, therefore, the higher the cliff index, the greater the difference in activity from the activity cliff (Lounkine, Wawer, Wassermann, & Bajorath, 2010).

1.5.4 Generative Topographic Mapping

Generative topographic mapping (GTM) is another visualisation technique that reduces the dimensionality of the input data. It has been adapted for chemoinformatics from computer science. GTM is a similar visualisation technique to SOM. However, with GTMS the input data is not mapped exclusively to one node, instead, the mapping is based on probability distribution function. Therefore, a compound can map to multiple nodes with a probability so that the multiple nodes can share a compound (Stumpfe & Bajorath, 2016).

1.5.5 Radial Clustergrams

Radial clustergrams are similar to dendrograms. Dendrograms are a tree diagram that is used to display hierarchical clustering. This visualisation technique, however, displays the cluster in different layers, from the centre of a circle outwards. Each layer of the tree is seen as a different layer. Radial clusters are considered to be the most effective and efficient way to visualise large amounts of data on a screen (Agrafiotis, Bandyopadhyay, & Farnum, 2007).

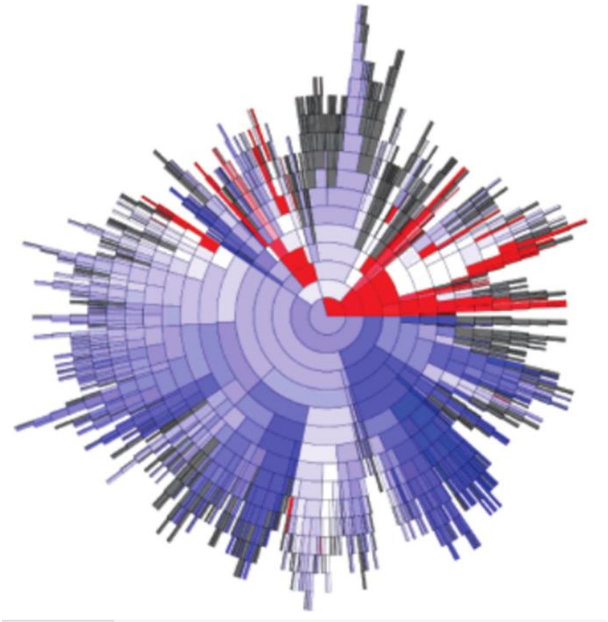


Figure 1-23: A radial clustergram, each strand are trees and the colour corresponds to the properties of the tree (Agrafiotis, Bandyopadhyay, et al., 2007)

Radial clustergrams are different from other methods as they do not display their data in a linear manner. The radial clustergram gets its name from it being a circular representation around a node, see Figure 1-23. Colour coding can be added to it to be able to clearly understand the properties and the molecular structures (Ivanenkov, Savchuk, Ekins, & Balakin, 2009).

1.5.6 Heat Maps

Heat maps are visualisation techniques that are suitable for large datasets and consequently, they have grown in their use recently due to an increase in the size of compound datasets. As traditionally, high quantity and high-density datasets that contain a lot of data points have been hard to visualise. A heat map is a mapping of data points to a grid. The heat in this instance refers to the number of data points within a region, generally the more data points the warmer the colour but this is dependent on how the map is constructed (Auman, Boorman, Wilson, Travlos, & Paules, 2007). This allows high-density data sets that contain a lot of data points to be analysed with colour showing the extent to which an area has been explored. Heat maps were originally used in social science, however, they have started to be used in chemoinformatics where there is a need to put high-density information into a visual representation that allows a proper biological context to be identified such as hot spots and clusters of biological activity. The difference in colours assists in hypothesis-generating or data interpretation (Juneau, 2015).

1.5.7 Scaffold Visualisations

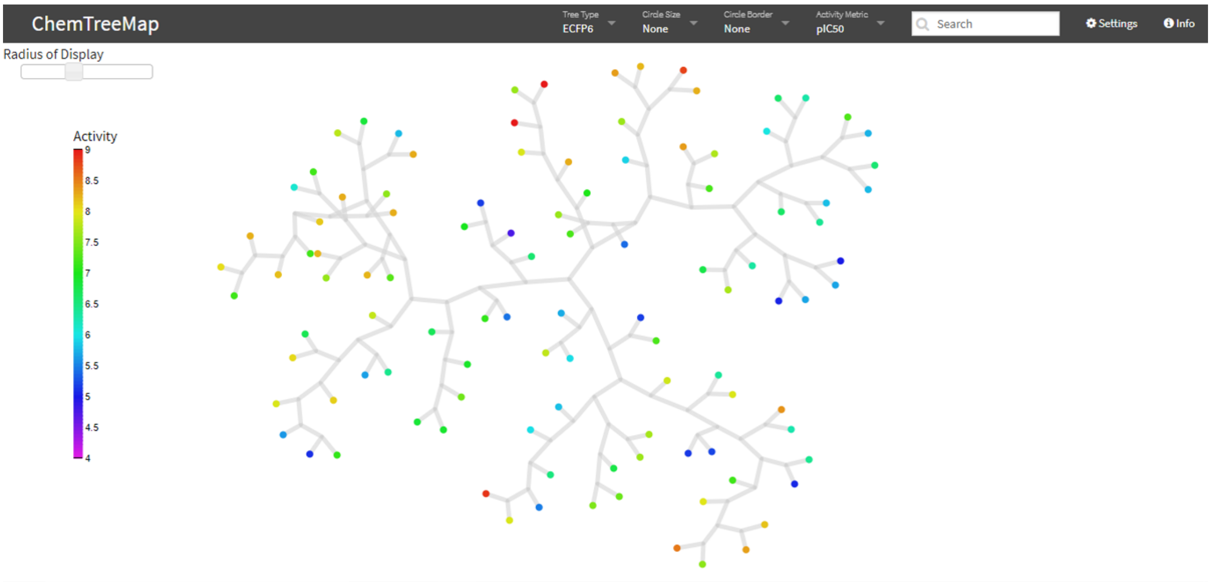
There have been several visualisations that have been designed specifically to demonstrate the scaffolds that are present within a dataset and how molecules can vary around the scaffold. Various scaffold visualisations have been reviewed further.

1.5.7.1 The Scaffold Tree

The scaffold tree was developed in 2007 that creates a hierarchical tree like structure based upon scaffolds. The nodes of the tree are scaffold structures that when progressing down the tree becomes more complex scaffolds. The various scaffolds are generated through a set of chemically derived meaningful rules. For each molecule the terminal chains are removed. Ring structures are then iteratively removed until there is just one ring remaining. If any of the removal iterations would lead to a disconnected structure, then this could not be removed. Each of the scaffolds can be coloured according to the property of interest, generally it indicates the ratio of active compounds for that corresponding scaffold (Schuffenhauer et al., 2007). The scaffold tree has sequentially been expanded and transformed into an open source library scaffold graph (Scott & Edith Chan, 2020). Wetzel's scaffold hunter tool takes advantage of these scaffold trees to create an interactive tool that uses the hierarchical structure along with the structure-activity relationships (Wetzel et al., 2009).

1.5.7.2 ChemTree Map

The ChemTreeMap visualisation tool attempts to display the chemical space that has been explored and the relationship between all of the molecules. As well as providing information about the overall relationships of the molecules, specific information was focused on, such as molecular properties and biological activities. The tree is a hierarchical tree that places similar molecules on the same branches and the length of these branches corresponds to the molecular similarity, Tanimoto similarity. The molecules are represented as extended connectivity fingerprints (ECFP). The size, the colour and the outline of the node are all user-defined based upon their interest and chosen properties (Lu & Carlson, 2016). An example of the ChemTreeMap can be Figure 1-24.



Tree Type ECFP6

Circle Size None

Circle Border None

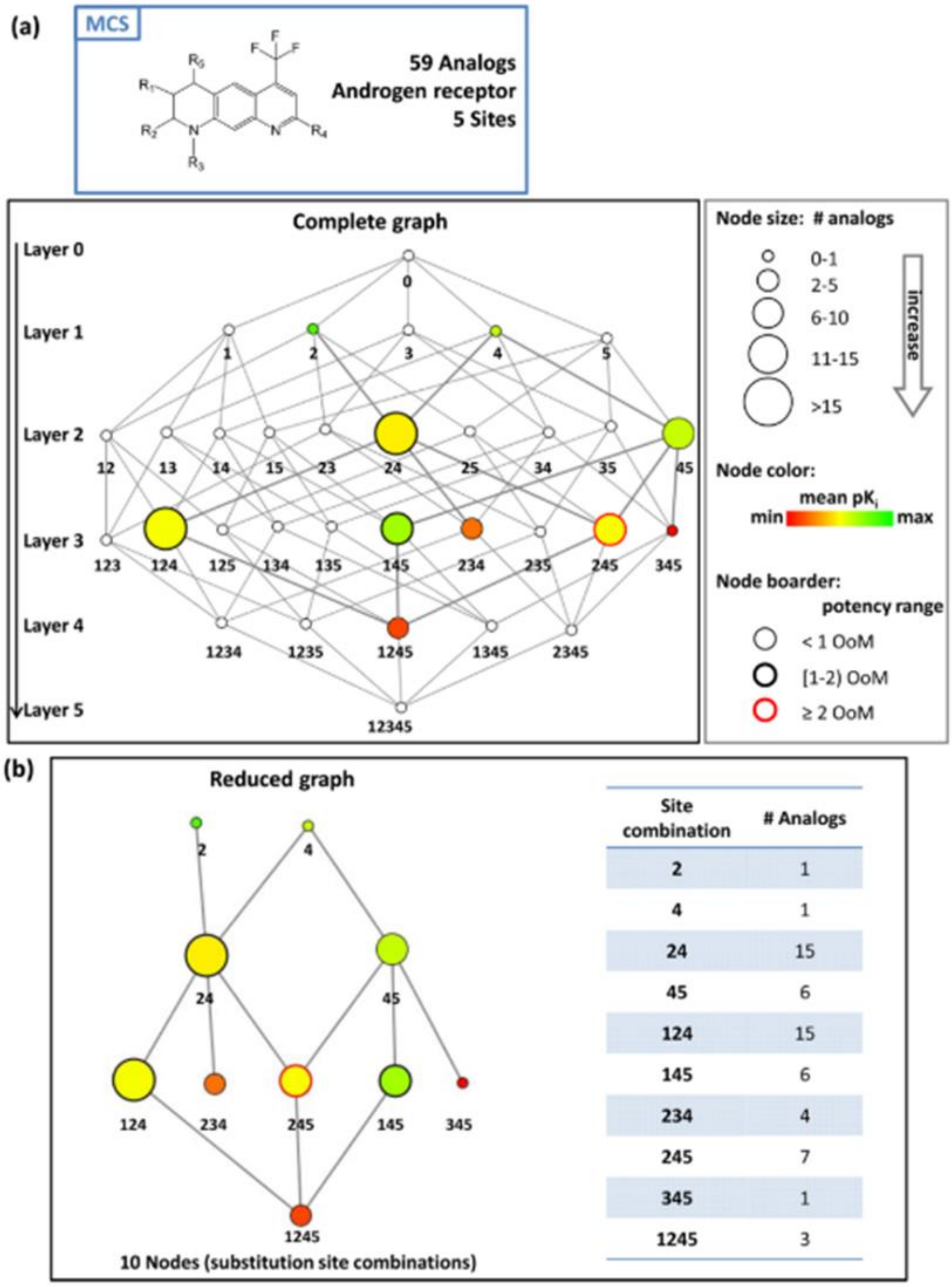
Activity Metric pIC50

Figure 1-24: ChemTreeMap example of Chk1 dataset

1.5.7.3 AnalogExplorer and AnalogExplorer2

AnalogExplorer is a graphical method that examines analogue series within a dataset. The AnalogExplorer tool works by generating Bemis-Murcko scaffolds and then potential MMP adapted scaffolds. By allowing slight adaptions it tries to combine closely related series' so similar SARs can be identified. These adaptations have two strict rules, the MMP transformation maximum size of the exchanged fragment is thirteen non-hydrogen atoms and the size difference between the two scaffolds must be less than eight non-hydrogen atoms. By using MMP transformations, it attempts to display activity cliffs and R-groups responsible for the cliff. The R-groups decomposition then occurred to identify all substitution sites around the scaffold whilst also identifying the R-groups (Zhang, Hu, & Bajorath, 2014). AnalogExplorer2 uses the same principals, however, the methodology is sensitive to the molecules' stereochemistry within the scaffold and R-groups (Hu, Zhang, Vogt, & Bajorath, 2015).

The AnalogExplorer is a directed hierarchical graphical representation, Figure 1-25a. The number of layers equates to the number of R-groups demonstrated. Therefore, it shows all possible substitution sites and site combinations that are currently present within the series. The beginning node is the analogue without any R-groups being added. Each layer indicates a different substitution site. The node size is the number of analogues within that node, the node colour is the mean pKi value and the node border is the potency range demonstrated within that node. Also, the node is labelled with a number and this is the node number that has been assigned to it. When a node does not contain a colour, this indicates that there are no corresponding molecules that only contain R-groups to this level. AnalogExplorer is a methodology that attempts to utilise the reduction in information by creating its own reduced graph via excluding redundant nodes, Figure 1-25b. As once the chemical graph trees have been displayed they are then also turned into the RG counterparts. For both trees, the positioning is determined by DAGLaout algorithm of Java package JUNG (“Java Universal Network/Graph Framework,” 2020).



Site combination	# Analogs
2	1
4	1
24	15
45	6
124	15
145	6
234	4
245	7
345	1
1245	3

Figure 1-25: Prototypic AnalogExplorer for Androgen receptor (Zhang et al., 2014)

1.5.7.4 rdScaffold Network

rdScaffold network is a scaffold network that has been implemented in RDKit. A molecule is iteratively fragmented according to pre-defined rules. The rdScaffold network then represents these different scaffolds where the node is the chemical structure of the scaffold and the edges are labelled with the operation. The operation indicates whether it is an initialisation (remove side chains), fragmentation (according to the rules) or as generic (replacement of non-carbon atoms with carbon atoms) (Kruger, Stiefl, & Landrum, 2020).

1.5.7.5 LASSO

Another scaffold representation is the layered skeleton-scaffold organisation (LASSO) graph. The scaffolds within the LASSO visualisation are of cyclic skeletons that follow Bemis Murcko scaffold rules. Each layer within the representation increases in complexity. A pie chart is represented at each layer for each unique cyclic skeleton, the colours within the pie chart demonstrates the activity values. A demonstration of how the LASSO graph can be generated is in Figure 1-26. Figure 1-26a is a dataset and their corresponding activity values. Figure 1-26b shows the generation of the cyclic skeletons and relating pie charts. Figure 1-26c demonstrates how the information can be layered. In the LASSO graph the chemical structures are replaced with the activity pie charts (Gupta-Ostermann, Hu, & Bajorath, 2012).

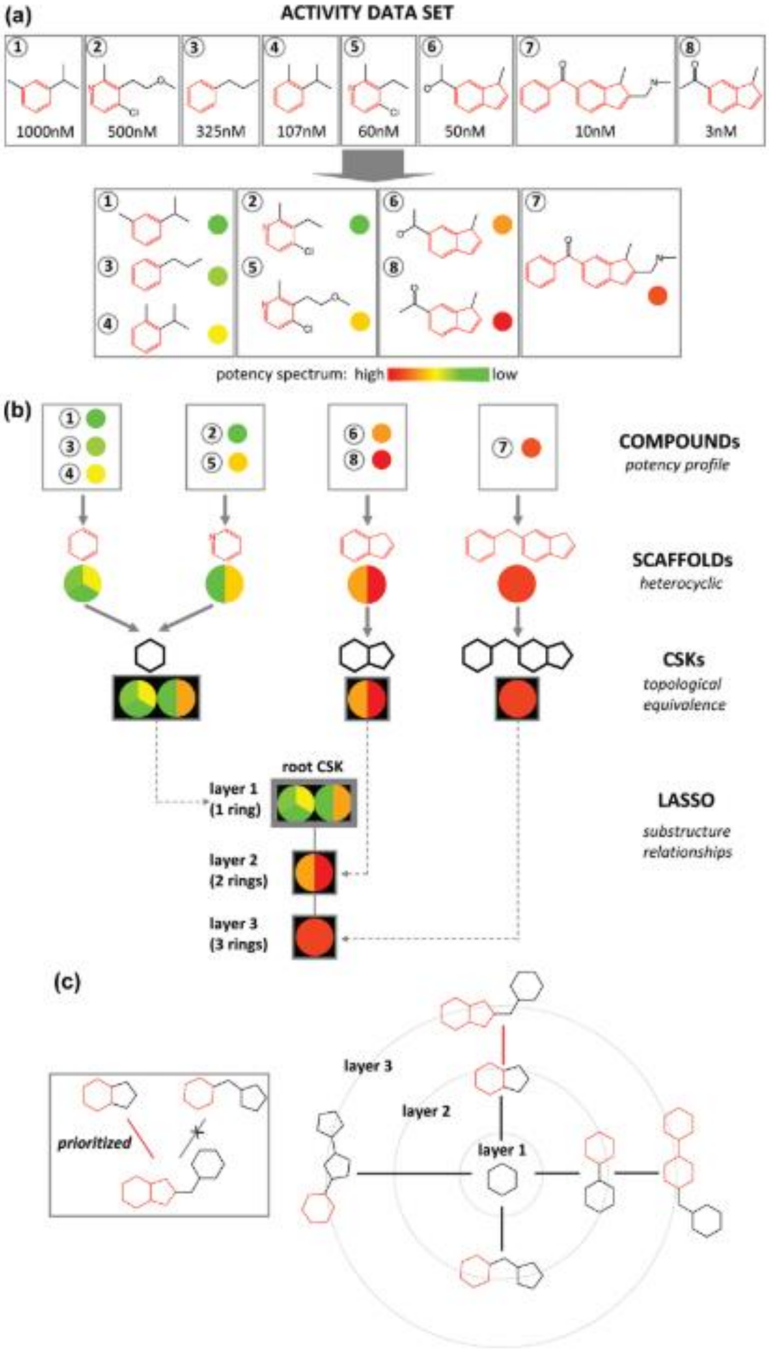


Figure 1-26: Illustration of how the LASSO graph is generated (Gupta-Ostermann et al., 2012)

1.6 Molecular Generation

The final chemoinformatics approach to be examined is molecular generation. This is typically referred to as de novo design. De novo design techniques design new molecules in an automated way. There are typically two approaches, either atom-based or fragment-based. Several of which will be discussed.

1.6.1 Atom-Based

Atom-based approaches build a molecule one atom at a time. Atom-based methods were first introduced to construct molecules to fit the binding site of a receptor. A molecule is grown atom-by-atom to ensure that the molecule can fit within the pocket and form potential hydrogen bonds (Bohacek & McMartin, 1994; Nishibata & Itai, 1991). However, these methods can produce vast number of molecules that are either not valid or do not resemble drug-like molecules. More advanced methods of atom-based approaches have been introduced in recent years due to deep learning and artificial intelligence approaches. These can be anything from recurrent neural networks, autoencoders and generative adversarial networks (N. Brown et al., 2020).

One of particular note, is the method developed by Pogány et. al (Pogány, Arad, Genway, & Pickett, 2019). This is a deep learning method that takes an RG as an input string and returns SMILES strings of molecules that should have the same RG as the input string. This method is a long short-term memory neural network. Therefore, a test set of molecules and corresponding RGs has to be imported into the model so that the model can learn how SMILES can relate to RGs.

As well as building molecules based on RG structures there have also been implementations that are based upon scaffolds. Langevin et. al recently incorporated recurrent neural networks and scaffolds to constrain the molecular generation to just a scaffold. A disconnected SMILES with wild atoms, *, to reserve connection points is input into the model. The model performs a sampling at the wild atoms to perform a scaffold transformation (Langevin, Minoux, Levesque, & Bianciotto, 2020). Another scaffold-based method is a variational autoencoder that allows a scaffold to be inputted and is grown atom or bond at a time (Lim, Hwang, Moon, Kim, & Kim, 2020).

Another recent method looks at exploiting scaffolds and fragmentation rules. A scaffold is decorated in either a multi-step process or single-step process. The decorators were created using MMPs or RECAP rules. One molecule in the training set can contain multiple scaffolds, and therefore, multiple decorations. However, all scaffolds have to retain one ring system and comply with the rule of 3. Additionally, the decorator fragments can only contain one attachment point and the scaffold can have up to four. These are then used to train a scaffold generator model, recurrent neural network, and a decorator model, a bidirectional recurrent neural network. (Arús-Pous et al., 2020).

1.6.2 Fragment-Based

In contrast, fragment-based approaches instead of building molecules atom-by-atom are based upon fragments. These occur through different fragmentation processes and can use fragment libraries. The fragments can be any size.

An early fragment-based de novo design method is the Topliss tree. The Topliss tree is a tree that attempts to guide molecular design to the most active molecule. Topliss trees are usually used to alter an analogue in a step-wise manner based upon the physicochemical properties of new fragments. The Topliss tree was first used to alter the substitution on an aromatic ring (Topliss, 1972). More modern versions based on similar principles to the Topliss tree are molecules generated from MMPs (Dossetter, Griffen, & Leach, 2013; Griffen et al., 2011) or apparent well-known functional group changes (Stewart, Shiroda, & James, 2006).

Several approaches alternatively, have rules that are implemented to fragment a molecule for part of the molecule to then be replace with a different fragment, examples are RECAP (Lewell, Judd, Watson, & Hann, 1998), BREED (Pierce, Rao, & Bemis, 2004) and BRICS (Degen, Wegscheid-Gerlach, Zaliani, & Rarey, 2008). These three methods will all be described further in Chapter 8.

Another method that use uses fragment mutations is Polishchuk's CReM method (P. Polishchuk, 2020). The CReM method looks at fragmenting the molecule and then using a database of interchangeable fragments to replace the fragment. There are three different ways the structure can be generated, through a mutate, grow or link. Where mutate is a replacement, grow is a mutate operation, however, it is the replacement of a hydrogen atom, and link is the replacement of hydrogen atoms on two separate molecules to appropriately link the two separate molecules together.

Fragment-based methods have also been developed with chemical scaffolds in mind. Jin et. al created a method that utilises junction trees and a variational autoencoder. A chemical graph is first turned into a junction tree based upon chemical substructures which have been determined from a training set constructed from building blocks. The junction tree and chemical graph are both fed into latent embeddings for the junction tree to be decoded and generates fragments to reconstruct into a chemical graph (Jin, Barzilay, & Jaakkola, 2018).

1.7 Conclusion

With the rise in cost of the drug discovery process a new era of chemistry has arisen. This has led to intensive developments in different chemoinformatics and data analyst techniques. Therefore, appropriate chemical structural representations are required so that both humans and computers can understand and interpret them. The three most common representation are SMILES, molecular graphs and connection tables.

These representations can be transformed into descriptors for data mining applications through mathematical procedures that allow the characteristics and properties of the chemical structures to be captured. Fingerprints are one of the most common descriptors, of which there are many different types: atom-based, path-based, hashed and ECFP. They all follow the similar concepts of a bit string containing ones and zeros to indicate whether the features are present or absent. However, careful consideration is required when selecting the best type of descriptor to use. One descriptor that is of particular interest in this thesis is the reduced graph due to its ability to compress the representation down to a more compact form that explains the molecules' potential binding abilities.

Molecular descriptors form the input to a variety of data mining and machine learning techniques in which the computer aims to identify patterns in the data. A key aim is to learn information to create models that relate the descriptors to the biological activities of the molecules so a prediction can be made for a previously unseen molecule. Other machine learning methods have the ability to group similar molecules together, where the similarity values can be calculated from either FPs or the maximum common substructures of chemical structures.

There are many different ways in which molecular datasets can be visualised in order to make it easier for chemists to understand and interpret molecular relationships. Some of the visualisation general techniques and scaffold visualisations used in chemoinformatics have been reviewed, however, there are varied approaches to indicating the chemical space that has been explored whilst also showing potential chemical space to work in. This can be used to aid QSAR problems in the drug discovery process, in key stages such as lead optimisation. It has also been demonstrated that there is a need for a method that has the ability to deal with large amounts of data whilst also being interpretable by the end chemist. As good as visualisations are, they are not a replacement for quantitative numbers and analytics and statistics. Visualisation are a tool to further aid decisions and understanding of data.

Finally, within this chapter, several different approaches to molecular generation were examined. Both atom and fragment-based methods that have presented. Some will provide inspiration for a new molecular generation algorithm that has been created and described within this thesis. The next chapter shall focus on attempting to best represent the relationship between a set of molecules within a dataset. Ultimately, this will provide a platform to be able to visualise these relationships and SAR.

2 Using Reduced Graphs to Represent Lead Optimisation Series

2.1 Introduction

The lead optimisation (LO) stage of the drug discovery process is a crucial step to create a compound or compounds with the desired property profile(s). The desired property profiles aim to balance absorption, distribution, metabolism, excretion and toxicity (ADMET) properties while retaining or improving on potency. Initially, a few active analogues are identified and, through an iterative process, substituents on the molecules are modified to build chemical series and ideally identify compounds with the desired properties. Therefore, LO datasets generally contain hundreds of molecules that are built around a small number of scaffolds. An example is illustrated in Figure 2-1, where the scaffold has different substitution points identified on the ring, indicated through the use of arrows. Several different suggestions of substructures at these different substitution points are shown. LO datasets are typically represented through structure-activity relationship (SAR) tables and Markush structures. Markush structures are used to indicate the core scaffold with the substituents shown as R groups and SAR tables indicate the variation in the activity for different substituents at each R group position (Agrafiotis, Shemanarev, Connolly, Farnum, & Lobanov, 2007; Hu, Stumpfe, & Bajorath, 2016). SAR tables and Markush structures have been widely adopted as they are easy to understand and interpret. However, both the core scaffold and the substituents are represented as substructural fragments, which limits the usefulness of the approach. This is because a slight change to the core can lead to a new scaffold being produced, which can then make it more challenging to interpret the SAR across the series. The work undertaken in this chapter establishes a new technique for representing LO series to overcome these limitations. It extends the principles of the SAR table by replacing the substructural fragments by reduced graphs (RGs) which are insensitive to some small changes in substructure.

RGs are used in Chemoinformatics to represent and search Markush structures in chemical patents, to identify SAR and for scaffold-hopping. (Barker et al., 2006; Barker, Gardiner, Gillet, Kitts, & Morris, 2003; Birchall, Gillet, Harper, & Pickett, n.d., 2008; Birchall, Gillet, Willett, Ducrot, & Luttmann, 2009; Gillet, Downs, Holliday, Lynch, & Dethlefsen, 1991; Gillet et al., 1987) RGs are a compressed representation of a molecular structure. The atoms within the structure are compressed into RG nodes that focus on functional groups that have potential to form binding interactions. Different substructures can be collapsed into the same node type. Therefore, the RG representation is a many to one representation, as multiple molecules can produce the same RG, Chapter 1. Thus, RGs highlight parts of molecules that could form interactions with a receptor which are crucial parts of drug molecules. Consequently, when chemical scaffolds are represented as RGs, these can highlight key interactions that are important in the drug binding process and can bring multiple Markush structure cores into a single representation. Another potential benefit of using an RG core as a scaffold is that it can allow the chemist to comprehend areas of chemical space that have been either over or under-explored. The degree to which regions of chemical space have been explored can be identified through the number of specific substructural groups represented by a particular node and the number of examples of each.

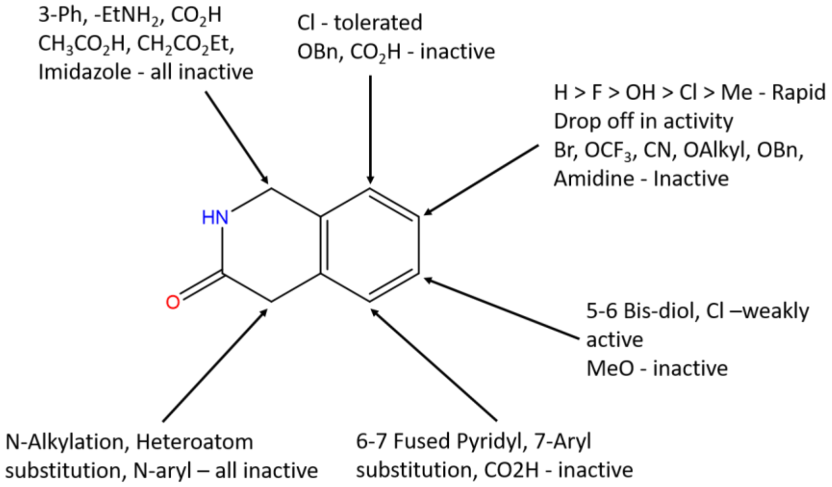


Figure 2-1: An example of how chemists explore different substituents on a core scaffold in lead optimisation

2.2 Methodology

Given a LO dataset, the aim is to organise the data into one or more RG SAR tables which can then be visualised. A RG SAR table is similar to the SAR table, however, all components of the table are represented as RGs rather than as substructural fragments. Figure 2-2 shows an example of what the approach is trying to achieve. This hypothetical LO series consists of molecules represented by three different scaffolds, each of which has been explored in a similar way through different substituents around the rings. Using RGs, the three separate scaffolds are combined to become one RG core which is comprised of three RG nodes. The nodes are defined based on the bonding capabilities of the underlying substructures and whether the node is aromatic or aliphatic and are labelled using different atom symbols outside of the commonly used set; Ga is an acyclic hydrogen bond acceptor (HBA), Ce is an aliphatic hydrogen bond donor and acceptor (HBD-HBA) and No is an aromatic inert node. More information on the definitions of RG nodes will be provided later in this chapter.

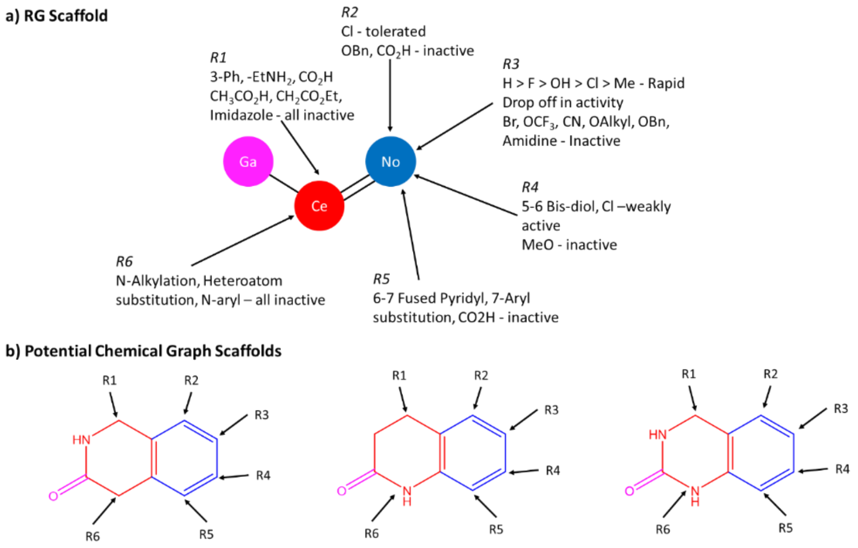
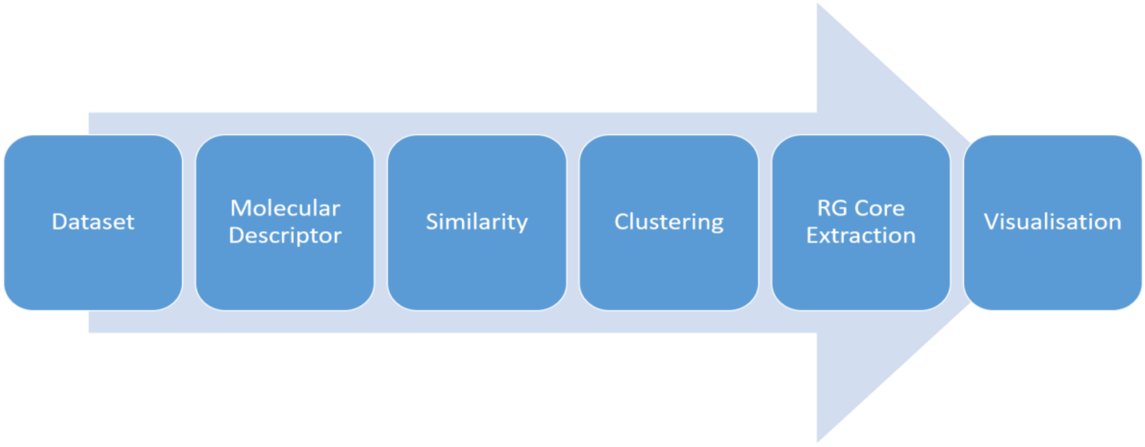


Figure 2-2: An example of how an RG core can be used to represent three closely related scaffolds from a lead optimisation project. Each RG node represents a different type of interaction and they have been coloured on the chemical graph according to the RG node types. a) A RG core, where Ga is an acyclic HBA node, Ce is an aliphatic HBD-HBA and No is an aromatic inert node. b) Chemical graph scaffolds that are represented by the same set of nodes shown in the RG core. The colours used in the chemical graph show how atoms map to the RG nodes.

The generation of RG cores for a dataset and to ultimately visualise them is comprised of several steps. The overall workflow is shown in Figure 2-3. A more detailed workflow where each of the steps has been expanded is shown in Figure 2-4.



Dataset

Molecular Descriptor

Similarity

Clustering

RG Core Extraction

Visualisation

Figure 2-3: Optimised Workflow

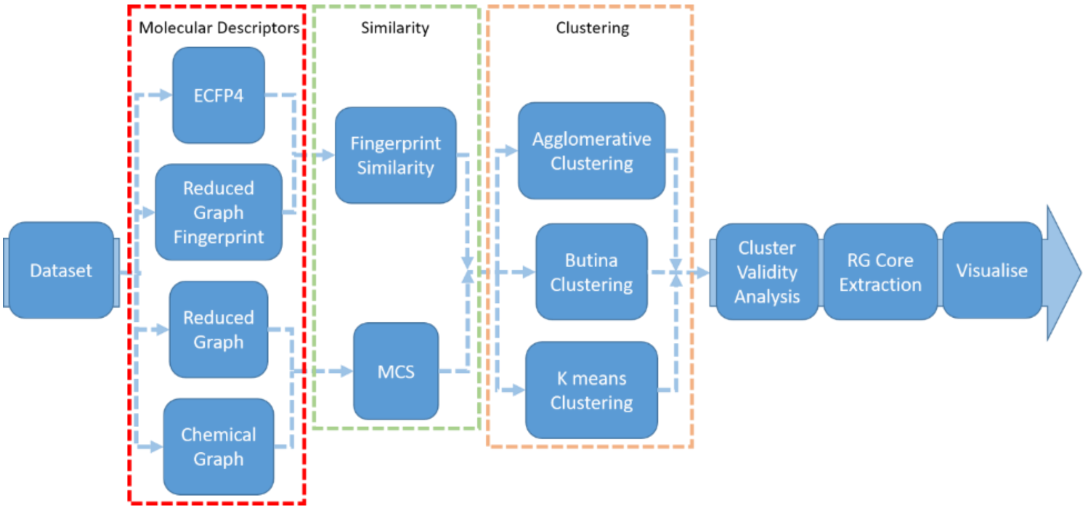


Figure 2-4: Experimental workflow followed to optimise the overall workflow

To extract the RG cores from a dataset the molecules must first be converted to RGs. From here, the relationships between the different RGs are investigated and one or more RG cores that are common to several molecules are obtained. The molecules can then be mapped to their respective core with any additional components represented through R groups, also represented as RG nodes rather than substructures. The extraction of the RG cores is the penultimate step before visualising the RG SAR tables. The final steps in constructing the visualisation tool are described in the next three chapter.

Investigations were performed to find the best way of implementing each step to analyse and visualise the data in an optimum way. The use of a clustering step before extracting the RG cores was investigated to determine if the prior organisation of molecules can achieve a better mapping of molecules to RG cores. The clustering step aims to group compounds so that the chemical relationships are easier to establish. Clustering requires a similarity method which in turn is based on molecular descriptors and different ways of calculating these are investigated. All of these stages are discussed in greater detail below.

2.2.1 Datasets

The workflow has been developed using nine different datasets. Four publicly available datasets were used initially, three of which have been extracted from ChEMBL23 (Gaulton et al., 2012). The three datasets are a P2x7 receptor dataset (ChEMBL4805), a subset of the P2x7 dataset consisting of compounds reported by GSK, and a neurokinin receptor dataset (ChEMBL249). A fourth dataset was included from a Bajorath et al. paper on LO (Vogt, Yonchev, & Bajorath, 2018). A further five datasets were also used in order to compare the methods developed here with a related approach called ChemTreeMap (Lu & Carlson, 2016). The five datasets were extracted from BindingDB, ChemBank and ChEMBL20 and consist of: cyclin-dependant kinase 2 (CDK2); checkpoint kinase 1 (Chk1); cytochrome P450 3A4 (Cyto); clotting factor Xa (FactorXa); and p38α MAP kinase (p38α). All datasets are analysed in this chapter and subsequent chapters.

A cleaning process was applied to all the datasets. The cleaning process involved removing duplicate molecules, removing molecules that RDKit could not read, and filtering out molecules that contained more than 50 heavy atoms. These were too complex to process and led to high computational costs. Table 2-1 shows the number of molecules in the datasets after cleaning.

Table 2-1: The datasets and the number of molecules contained in each

Name of Dataset	Number of Molecules
Bajorath	2549
CDK2	1368
Chk1	106
Cyto	6370
FactorXa	1956
Neurokinin	2475
P2x7	2259
P2x7 Subset	691
p38α	3644

A further pre-processing step was carried out. Datasets Bajorath, P2x7 and P2x7 subset were manually clustered to provide a benchmark for evaluating the clustering methods. Manual clustering is a subjective process as different chemists may cluster compounds in different ways. Therefore, all manual clustering was done by the same chemist (i.e. the author of this thesis). The manual clustering examined the molecules within a dataset to find related molecules. Both the chemical graphs and the reduced graphs were considered. The primary consideration was that molecules should have similar chemical scaffolds, where similarity was determined according to binding potential based on the RG node definitions. An example is shown in Figure 2-5. The coloured substructures show the basis for the clustering; they are the same or vary slightly while retaining similar binding potentials. In Cluster 1, the molecules all have a benzene ring connected to a carbon atom then an amide group that links to an aliphatic ring with hydrogen bond acceptor potential and with a carbonyl substituent. When a molecule was closely related to two clusters, it was assigned to the cluster that it is more closely associated with.

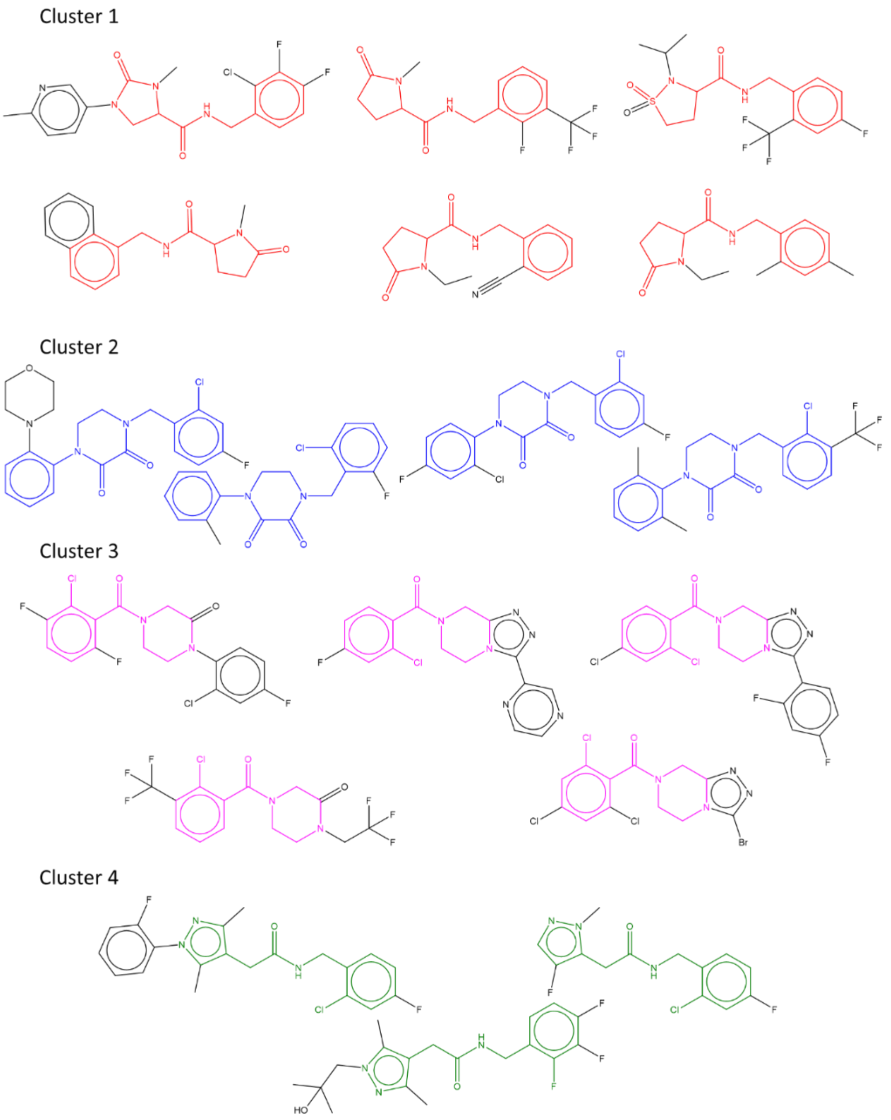


Figure 2-5: A subset of the P2x7 subset dataset which illustrates the manual clustering. The highlighted substructures show the basis for the clustering with the different colours indicating different clusters. Substructures of the same colour are similar with similar binding features.

2.2.2 Reduced Graph Generation

Reduced graphs are fundamental to the approach. They are used to represent the molecules and then to group them according to common RG cores.

To generate the RGs, a program was implemented using python and RDKit. This is based on a published implementation, which has not been publically released (Barker et al., 2003; Gardiner, Gillet, Willett, & Cosgrove, 2007; Gillet et al., 1991, 1987). Users can set their own definitions of a hydrogen bond acceptor (HBA) and a hydrogen bond donor (HBD) in both implementations. However, the new implementation allows more flexible definitions of the resulting RG via several different parameters that can be set depending on the interests of the chemist. Therefore, there can be many different variations of the RG for one molecule.

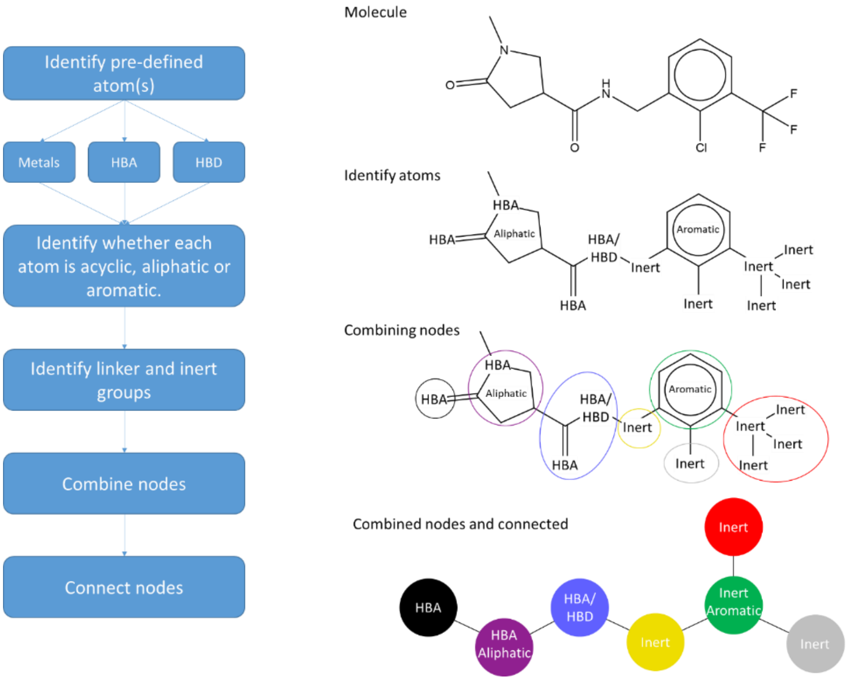

Identify pre-defined atom(s)
Metals
HBA
HBD
Identify whether each atom is acyclic, aliphatic or aromatic.
Identify linker and inert groups
Combine nodes
Connect nodes
Molecule
Identify atoms
Combining nodes
Combined nodes and connected
Aliphatic
Aromatic
HBA/ HBD
Inert
HBA
HBA Aliphatic
Inert Aromatic

Figure 2-6: RG generation workflow alongside an example

Figure 2-6 indicates the workflow undertaken to generate a RG from a molecule. The RG code initially identifies functional groups that are: HBD; HBA; or both hydrogen bond donor and acceptor (HBD-HBA), with the definitions being read in from an input file. The definitions are in SMILES Arbitrary Target Specification (SMARTS) format (Daylight, n.d.). Acyclic functional groups are represented by nodes with the appropriate label. Ring atoms are then identified and individual rings are labelled as aromatic or aliphatic along with their hydrogen bonding characteristics: non-bonding; HBA; HBD or HBD-HBA. The number of ring nodes is determined as the smallest set of smallest rings.

Linker groups are then identified: these are atoms that have not previously been defined as belonging to nodes. By introducing linker nodes, all of the atoms within a molecule are incorporated into nodes. Nodes that are adjoining and of the same type are combined together to form one node. Additionally, HBD or HBA node next to a HBD-HBA would also be collapsed together. An exception to combining adjoining alike atoms are atoms that are within a ring. Atoms are only combined with atoms of the same ring, therefore, a fused ring would be two separate nodes not one.

The nodes are then connected via edges. A pair of nodes is connected by a single edge unless the two nodes represent fused rings when they are connected by two edges. Three additional rules have been explicitly written into the code. The first is the order in which the predefined nodes are established and found. Initially, metals are found if the metal parameter is used, more will be detailed about this later in this section; then the defined atoms for HBD and then the defined HBA atoms. The second is the handling of carbonyl groups; both the carbon and oxygen are considered as a single HBA node unless the carbon atom is within a ring when the oxygen is considered on its own. Another example of an explicit rule is that if a halogen is next to a HBA group, it becomes a part of that node. This rule means that, for example, acyl chlorides form a single node, Figure 2-7.

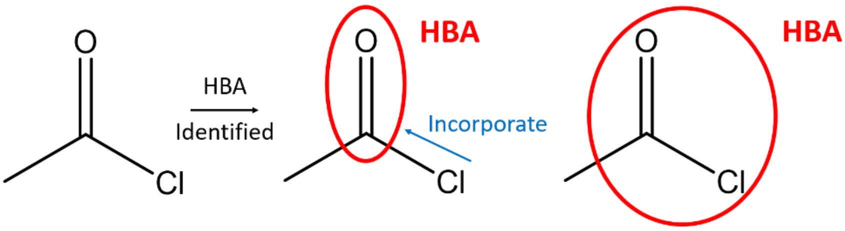


Figure 2-7: Incorporating halogens into HBA nodes

The labels for the different node types are atomic symbols outside of the standard atom set included in organic molecules, as shown in Table 2-2. The RGs are then written as valid SMILES strings, albeit with atom symbols that are not commonly seen in organic molecules. Each node is also annotated with a SMARTS string that represents the corresponding substructure, with the attachment points being labelled as a wild atom. For example, a phenyl ring will be represented by a node labelled as No and annotated by c1ccccc1. Within this implementation, only HBA and HBD definitions were used, along with metal definitions when the metal parameter was used.

Table 2-2: A table showing the reduced graph node definitions

Node Definition	SMILES code
Acyclic inert	Li
Acyclic HBA	Ga
Acyclic HBD	Gd
Acyclic HBD-HBA	Ge
Aromatic inert	No
Aromatic HBA	Na
Aromatic HBD	Nd
Aromatic HBD-HBA	Ne
Aliphatic inert	Co
Aliphatic HBA	Ca
Aliphatic HBD	Cd
Aliphatic HBD-HBA	Ce
Metal	Au
Complex	Hg

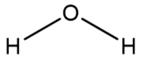
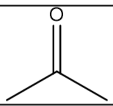
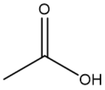
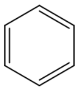
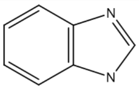
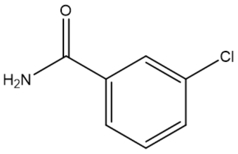
The default RG is illustrated in Table 2-3 for several simple chemical graphs. The default RG is the simplest form used here. The node types consist of aliphatic ring nodes, aromatic ring nodes, acyclic nodes and linker nodes and terminal carbon atoms are recursively removed, so no terminal linker nodes are generated. All nodes except linkers are labelled as either inert (no hydrogen bonding characteristics), HBA, HBD or HBA-HBD.

The first four examples in Table 2-3 consist of just one node. In the first molecule, just the oxygen atom is identified (as hydrogens are not included directly within the default reduced graph) and is assigned as a Ge node, as it is both a hydrogen bond acceptor and a donor. For the second molecule, just the carbonyl is identified. The two carbons that are on either side of the carbonyl are terminal carbon atoms and, as above, are recursively removed. For the third molecule, both of the oxygen atoms are assigned to a single node as they are adjacent. The fourth molecule is a benzene ring and all of the atoms are non-bonding aromatic atoms. These are all part of the same ring and they are therefore combined into one node. The fifth molecule is an example of a fused ring that results in two nodes connected by a double bond. The ring on the left is all carbon and, therefore, represented as an Aromatic inert node, No. The ring on the right contains two nitrogen atoms that are hydrogen bond acceptors, and becomes an Aromatic HBA ring node, Na. The final molecule consists of an Aromatic inert ring node (No) that is connected to an Acyclic HBD-HBA node (Ge) representing the amine group and an Acyclic inert node (Li) which represents the Chlorine atom.

The implementation allows the different types of RGs to be created through use of parameters. This is because different features are known to be important for different biological targets. The parameters consist of four different types which can be set independently as on or off:

- Terminal carbon chain
- Complex
- Double bond
- Metal

Table 2-3: Table showing simple chemical graphs and reduced graphs

Chemical Graph	Reduced Graphs
	Default
	Ge
	Ga
	Ge
	No
	No═Na
	Ge—No—Li

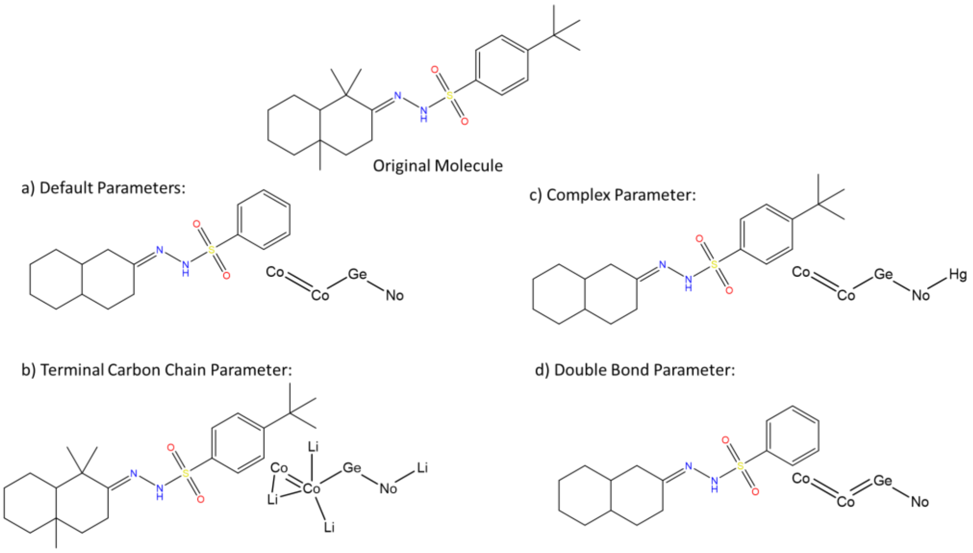


Figure 2-8: Example of the various user parameters and how they can affect the RG definitions

Figure 2-8 and Figure 2-9 show several larger molecules to illustrate how the RGs vary when different parameters are used. Figure 2-8 demonstrates the effect of the first three parameters. The default setting is that none of the four parameters are set. The fused ring within Figure 2-8 results in two Co nodes connected by a double bond. When the terminal carbon chain parameter is set, terminal carbon chains are identified as linker nodes, Figure 2-8 (b). This results in four linker nodes: two are connected to one of the Co nodes; one is connected to both of the Co nodes; and a fourth Li node is attached to the Aromatic inert node, No. In the latter case, the four carbon atoms, trimethyl, are combined into a single Li node. Terminal linker nodes could be of interest to a chemist as they might indicate nodes that play some role in forming hydrophobic interactions between the potential drug and the receptor. The “complex” parameter defines heteroatoms or branched groups not previously identified as nodes, note that this does not include straight-chain carbon groups. Therefore, branched terminal carbon chains fall into this category, but single carbon atoms do not. The results of using the “complex” parameter can be seen in Figure 2-8 (c). The double bond parameter retains the bonding between nodes so that an edge can be labelled as a single, double, or triple bond. Allowing the bonding to be retained also allows any resonance occurring in the molecule to be retained, as demonstrated in Figure 2-8 (d).

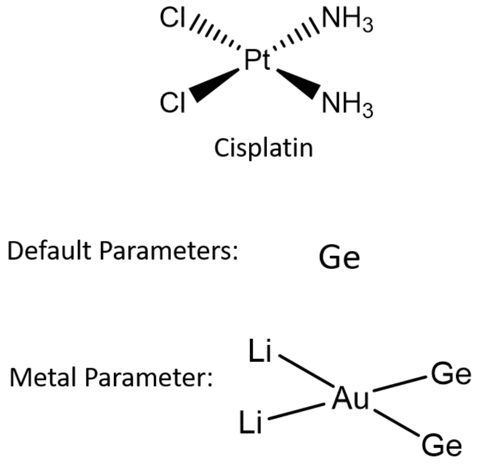


Figure 2-9: Example of the metal parameters being used on cisplatin

Finally, as shown in Figure 2-9, the metal parameter defines all metal atoms as individual metal nodes. Metals atoms are particularly important for biometallic drugs, such as cisplatin, which is a common chemotherapy drug used to target specific cancers. When the default RG settings are used, just one node is found for the molecule in Figure 2-9: an acyclic HBA-HBD node, Ge. This is because the NH_3 functional groups are classified as acyclic hydrogen bond donor and acceptor groups. The platinum is defined as a hydrogen bond acceptor due to the hydrogen bond acceptor definition used. As these atoms are connected they are combined into a single node with all the properties; the chlorine atoms are identified as terminal linker nodes which get removed. When the metal parameter is set, the platinum atom is identified as a metal node, and all of the atoms form separate nodes.

2.2.3 Molecular Descriptors

Four different descriptors were used for the clustering experiments: the chemical graph (CG), RDKit’s Morgan radius 2 fingerprint (M2FP), the reduced graph (RG), and a reduced graph fingerprint (RGFP), respectively. The M2FP and CG descriptors are based on the original all-atom structures and allow similarity to be calculated using a fingerprint and a maximum common subgraph (MCS) approach, respectively. The chemical graphs were then converted to reduced graphs which can be compared at the graph level or as fingerprint representations.

The Morgan radius 2 fingerprint, which is equivalent to an extended connectivity fingerprint (ECFP4), was generated in RDKit by setting a radius of 2 and a bit length of 2048. The chemical graph descriptor was created by entering the molecule as a SMILES representation into an RDKit molecule object (“RDKit: Open-Source Chemoinformatics,” 2018). The RG was generated using the methods described above. The RGFP was generated in RDKit by passing the RG SMILES representation into RDKit and calculating the M2FP for it.

2.2.4 Similarity

Calculation of pairwise similarity is used for both the clustering and the RG core extraction steps. The similarity matrices for the two fingerprint representations (generated from the CG and the RG respectively) are calculated using the Tanimoto coefficient. Calculating a similarity matrix in this manner is a rapid and computationally cheap calculation.

Pairwise similarities for the CGs and the RGs are calculated based on the MCS, which is more computationally expensive than calculating similarity based on fingerprints. There are two different types of MCS: a disconnected MCS (dMCS), and a connected MCS (cMCS), as shown in Figure 2-10. The parts highlighted in red are the atoms and bonds that are involved in the MCS. Figure 2-10a shows the connected cMCS. Figure 2-10b shows the disconnected dMCS. The dMCS is made up of two components which are shown by the blue wavy line and labelled (i) and (ii). Both the disconnected and connected versions were calculated for both the chemical graphs and the reduced graphs. It was thought that the disconnected version might provide useful information that might be missed using the connected version. dMCSs tend to be larger and can detect similarities between molecules where there is a segment in the middle that differs or where they have large common substructures connected via different atoms, as shown in Figure 2-10.

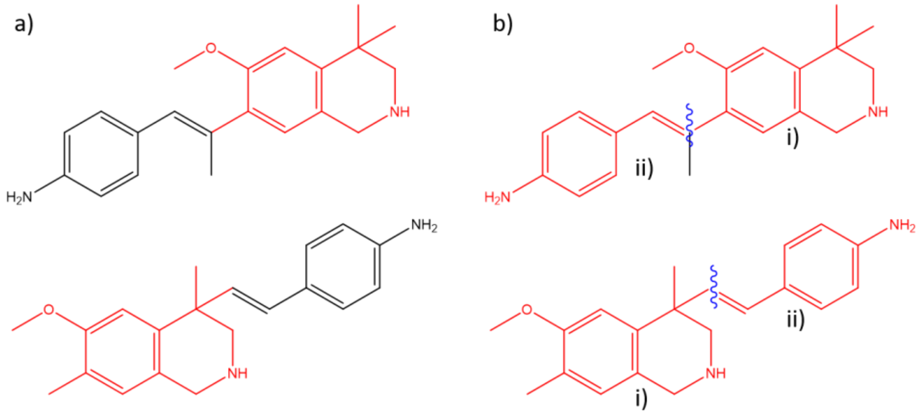


Figure 2-10: Example of a connected (a) and a disconnected (b) MCS

The MCS was calculated using RDKit’s implementation (“RDKit: Open-Source Chemoinformatics,” 2018). The MCS algorithm is configured to compare the edges (bonds) as well as the nodes (atoms) so that, for example, a benzene ring does not match a cyclohexane ring. The MCS algorithm in RDKit is currently only able to find the cMCS. Therefore, an iterative process was implemented that identified the initial MCS and then removed the atoms in the MCS from each molecule. The process was then repeated until no further MCSs could be found. The MCSs found in each iteration were then combined to form a dMCS. The dMCS process is demonstrated in Figure 2-11, where the largest MCS is located in the first step, highlighted in red. The atoms contained in this MCS are then deleted, generating two fragments. The next largest MCS is found and deleted and so on until it is no longer possible to find an MCS. As the fragments get smaller then so does the MCS, and very small substructures are not likely to be of interest as the molecule becomes so disjointed that it begins to lose its chemical meaning. However, it was not possible to prevent this occurring in the RDKit implementation. Since this work was undertaken, there is now an implementation in ChemAxon (ChemAxon, 2020) where, potentially, a threshold could have been introduced to ignore MCSs that have a smaller number of atoms than the threshold, however, this was not implemented here.

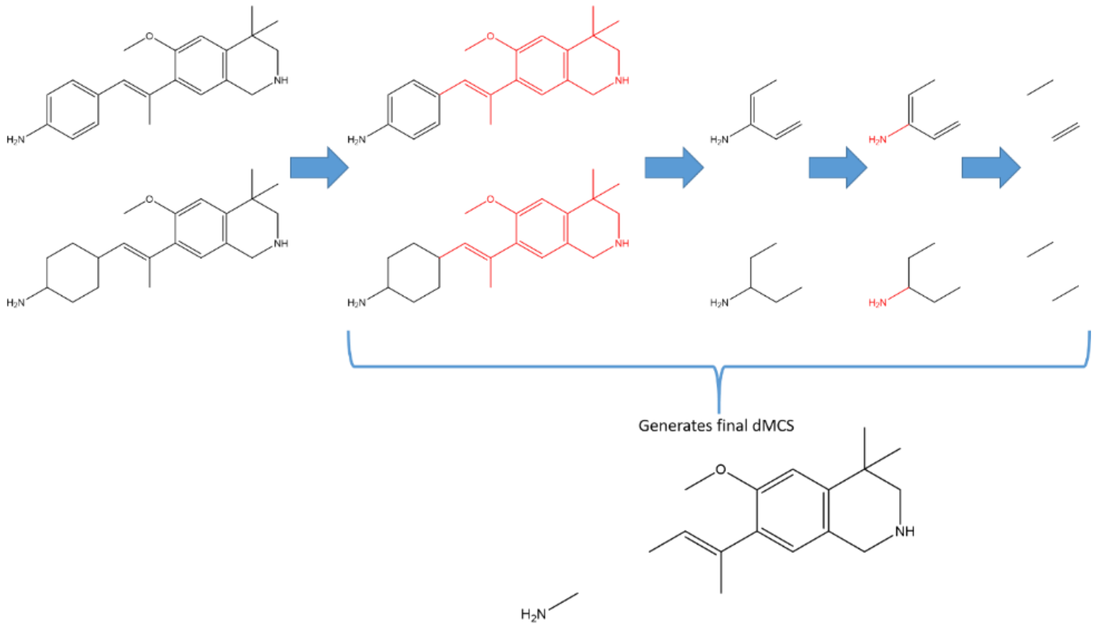


Figure 2-11: Example of the iterative process in generating the dMCS

The RG MCS works in the same way. The only difference is that the RDKit algorithm has been adjusted to allow for the smaller size of the RGs. The adjustment allows for an MCS to be just one node. This is because a node in this instance typically represents more than one atom in the chemical graph. A potential limitation of this approach is that the methodology could recognise two RGs as sharing one node in common, even when that node is not likely to be significant, for example, it could be a linker group which represents just a single carbon atom. Since this work was undertaken RDKit now allow one atom to be the MCS.

Once the MCS has been found, the similarity is calculated using the graph-variant of the Tanimoto coefficient (Maggiora & Shanmugasundaram, 2004).

$$Tc = \frac{MCS}{A + B - MCS} \tag{2.1}$$

Where A is the number of atoms (or nodes for the RG) in the first molecule, B is the number of atoms (or nodes) in the second molecule and MCS is the number of atoms (or nodes) in the MCS.

The disadvantage of the chemical graphs over the RG is that the MCS is a lot more computationally expensive to calculate due to the larger, more complex graphs.

2.2.5 Clustering

The similarity matrices are used as the inputs to three different clustering algorithms: Butina (sphere exclusion), agglomerative (Ward’s), and K-means. These were implemented using algorithms in

RDKit and the scikit learning package (Pedregosa et al., 2011; “RDKit: Open-Source Chemoinformatics,” 2018). These three cluster methods were chosen because they represent the main methods applied to chemical datasets and include both hierarchical and non-hierarchical approaches.

Different parameters were investigated for each method. For the agglomerative and K-means methodologies, the number of clusters is user-defined. In agglomerative clustering, the number of clusters to be output has to be specified due to the hierarchical nature. For K-means clustering, the number of clusters (K) is defined upfront. For both of these methods, the number of clusters was varied between 2 and 150. For the Butina methodology, the number of clusters depends on the similarity threshold used. For the experiments in this chapter, the similarity threshold was varied between 0.1 and 0.9, in increments of 0.1.

Every combination of the molecular descriptors and clustering algorithms was investigated for each dataset for the different numbers of clusters or similarity thresholds. The resulting clusters were analysed to identify the most appropriate combination for each dataset.

2.2.6 Clustering Validity Analysis

Finding the most appropriate clustering technique is not trivial. After generating the clusters for each algorithm and level, various cluster validity scores were calculated provided that the condition in Equation 2-2 was met. Equation 2-2 is designed to prevent clusters from being too small and avoiding the situation of more singletons than clusters. Furthermore, clustering validity scores were not calculated if there was only one cluster, which sometimes occurred for the Butina clustering.

$$number\ of\ clusters + number\ of\ singletons < \frac{number\ of\ molecules}{2} \qquad (2.2)$$

There are three types of clustering validity statistics: internal, external and relative. Those predominately studied in this chapter are internal and external cluster indexes.

Two external indexes were used to evaluate how close each algorithmically determined clustering is to the manual clustering. These are cluster purity and v-measure; these have been defined in Chapter 1. Each cluster in the manually clustered dataset is assigned a label. Each labelled cluster is compared with the computationally assigned clusters and the cluster with the largest number of matching molecules is assigned the same label. The indexes are then calculated, to quantify the differences between the manual clusters and the computed clusters. Cluster purity calculates the percentage of correctly classified molecules, whereas the v-measure is based on a combination of the homogeneity and completeness of the clusters. Homogeneity measures to what extent each cluster only contains members from a single manual cluster, and completeness measures the extent to which all molecules in a single manual cluster are assigned to the same cluster. The advantage of the external indexes is that they are based on how molecules are placed within clusters and not on the similarity values.

Indexes that do not use any prior knowledge were also be examined, as this is how the methods would be used in practice. Six internal indexes were used to see whether the same conclusion could be drawn from the internal indexes as from the external indexes. The internal indexes are based on the underlying similarity values. Since these values are not directly comparable across different descriptors, the internal cluster indexes cannot be compared across the different descriptors. The internal indexes that were used are: Ball-Hall index; Calinski Harabasz; Davies-Bouldin; Dunn; Kelley; and silhouette average. These have been defined in Chapter 1.

2.2.7 Reduced Graph Core Extraction

Once a dataset had been clustered, each cluster was then processed in turn to generate one or more RG core from the RGs within a cluster. The RG subgraph that is contained in most, if not all, molecules within a cluster is extracted and becomes known as the core.

The construction of the RG cores follows an existing method by Gardiner et al. (Gardiner et al., 2007) and is based on calculating MCSs between the RGs. Figure 2-12 shows a flowchart of the algorithm. The steps are as follows:

1) Generate the RGs for each of the molecules within the cluster.
2) Calculate the pairwise similarity between all RGs in the cluster. The molecule with the most neighbours above a user-defined similarity cut off is identified as the centroid.
3) Find the most distant neighbour within the centroid’s neighbours and calculate the MCS between these two RGs. This MCS is then set as the current representative (curr_rep_MCS) for a core MCS.
4) For all other RGs in the cluster,
    a) Compare the current representative with the RG.
    b) If the current representative is contained within the RG, then the RG is noted as being associated with this representative.

c) If the current representative is not contained within the RG, then a new MCS is found between the RG and the current representative. This new MCS must also meet specific requirements: it must be equal to or larger than the minimum number of user-defined nodes, and it must be a subgraph of the current representative.

   i. If this MCS passes these requirements, it is set as the new current representative MCS, and the search continues from the current point within the dataset.

   ii. If the MCS does not pass these requirements, then the current representative remains and the search continues.

5) Once all of the RGs within the cluster have been searched, the current representative MCS becomes a core RG.

6) Check whether all the RGs within the dataset are associated with a core within the core list.

   a) If all molecules are associated with a core the process stops.

   b) If not, then the process is repeated on the remaining RGs.

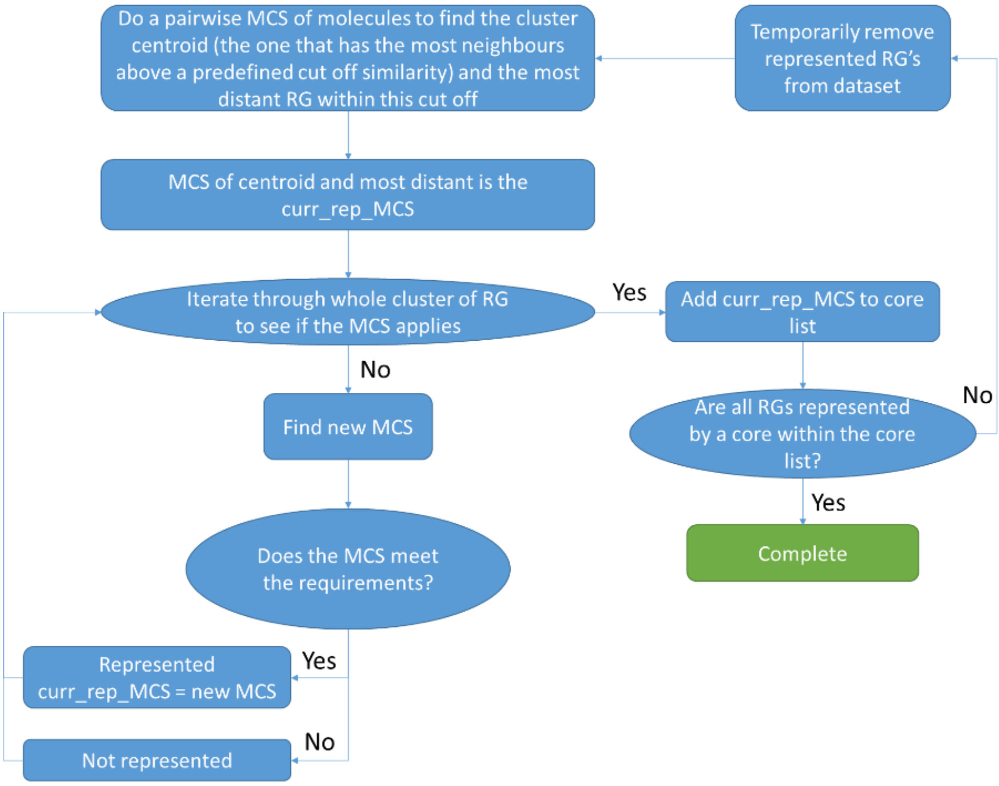

Do a pairwise MCS of molecules to find the cluster centroid (the one that has the most neighbours above a predefined cut off similarity) and the most distant RG within this cut off
Temporarily remove represented RG's from dataset
MCS of centroid and most distant is the curr_rep_MCS
Iterate through whole cluster of RG to see if the MCS applies
Yes
Add curr_rep_MCS to core list
No
Find new MCS
Are all RGs represented by a core within the core list?
No
Yes
Complete
Does the MCS meet the requirements?
Yes
Represented curr_rep_MCS = new MCS
No
Not represented

Figure 2-12: Flowchart of establishing the RG core

Figure 2-13 and Figure 2-14 both show implementation examples of this workflow. Figure 2-13 demonstrates a straightforward example. All instances are neighbours of one another within the defined similarity threshold and, therefore, the first molecule is identified as the centroid. The furthest neighbour is then determined. The MCS between these two molecules is extracted and, as it meets the minimum core size requirement of four, the process continues. This MCS is found in the rest of the molecules within the cluster, and the process completes with one core MCS identified. This is a simple example as all molecules contain the initial MCS.

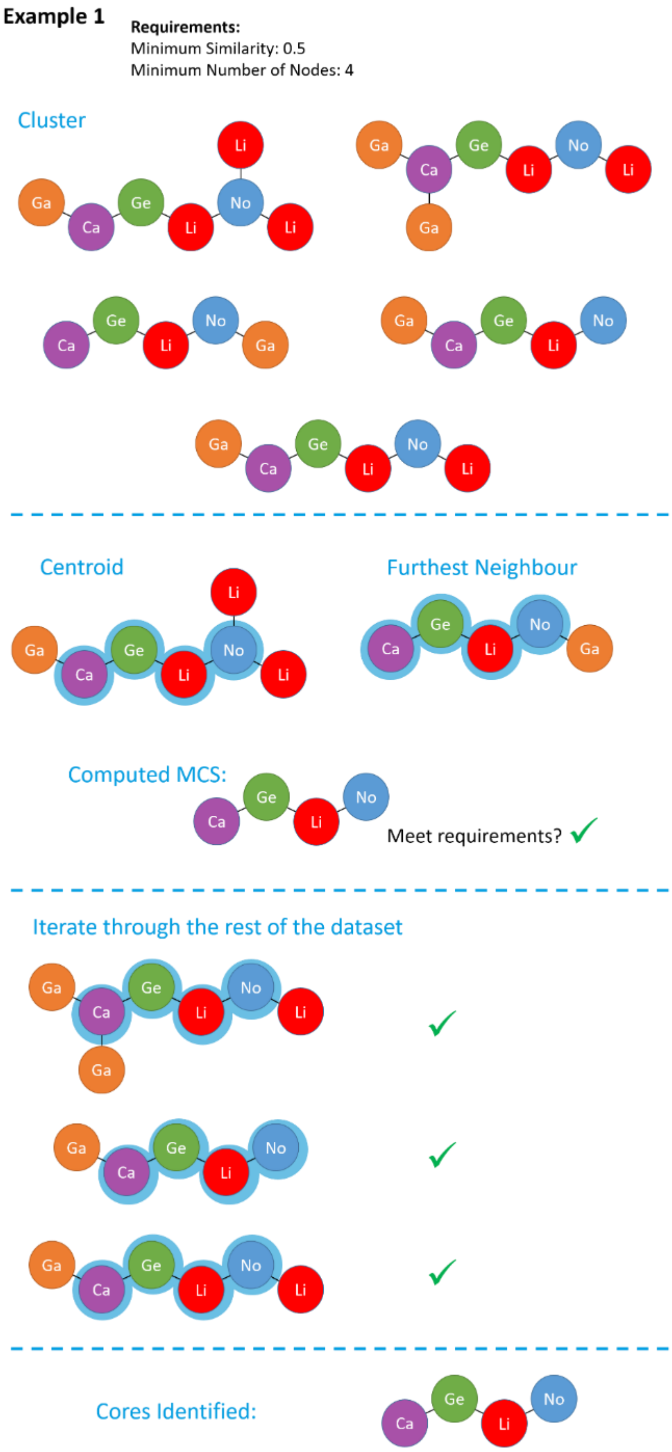


Figure 2-13: First example of the RG core extraction methodology

Figure 2-14 shows a more complex example when the original MCS is not present within all the molecules. This is the same dataset as in Figure 2-13, but it is based on a different similarity threshold. As the minimum similarity has increased the number of neighbours is no longer the same for all of the molecules. Three of the molecules now have three neighbours, including the first molecule. The first molecule is chosen as the centroid, as in the previous example. However, a different molecule is now identified as the furthest neighbour and, therefore, a different initial MCS is constructed. The candidate MCS is then tested on all the molecules within the dataset to analyse whether it is present, as previously. The first molecule contains the MCS, however, the second molecule does not. Therefore, as shown in the workflow, a new candidate MCS is determined by comparing the current MCS with the RG that does not match. A new valid MCS is found which becomes the new representative MCS. The iteration continues using the new MCS, which is found in the remaining molecules. The same RG core is identified in both Figure 2-13 and Figure 2-14 but the data is processed in different ways. However, this is not always the case when the threshold is varied.

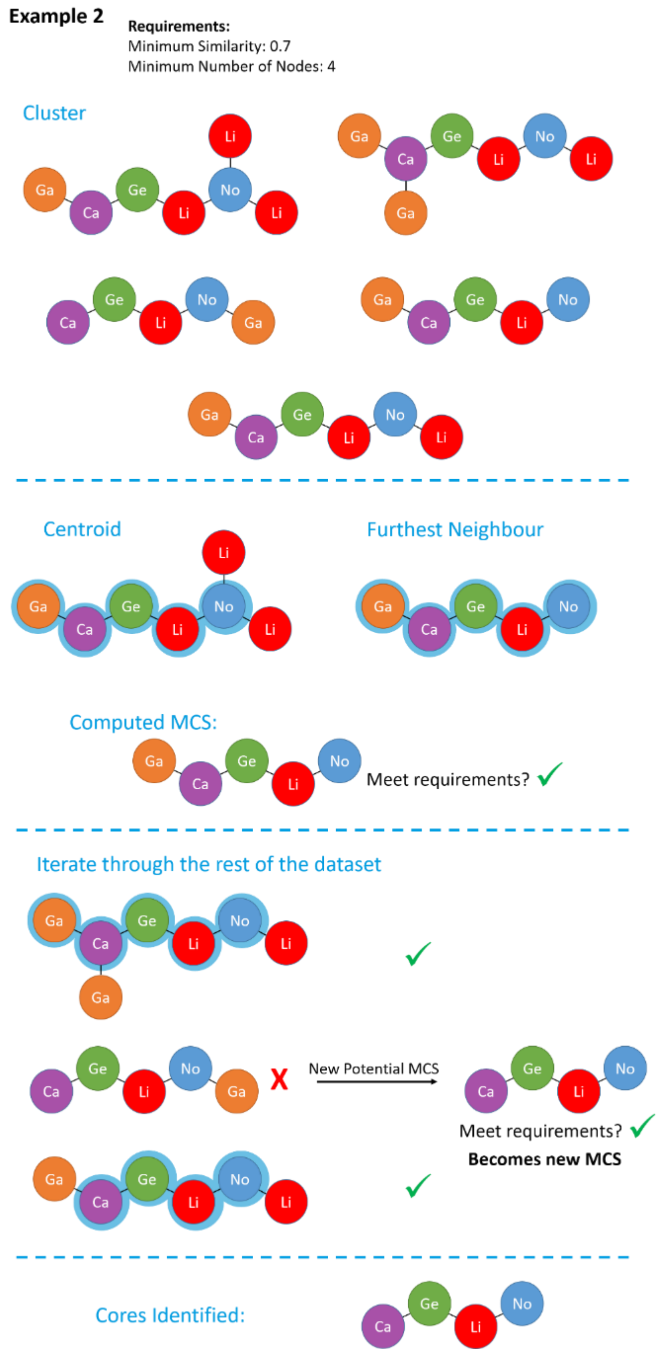


Figure 2-14: Second example of the RG core extraction methodology

2.2.7.1 Re-examining RG Cores

Following the identification of RG cores for each cluster, a second pass through the data is made since some molecules have the potential to map to multiple RG cores. Each molecule is compared to each RG core and is associated with each RG core that it contains so the number of molecules that map to the RG core can increase. In some instances, the RG core could have been initially generated from a single RG, due to the re-examination step it can now be represented by multiple RGs and ultimately molecules, Figure 2-15. RG cores can also represent molecules across several clusters.

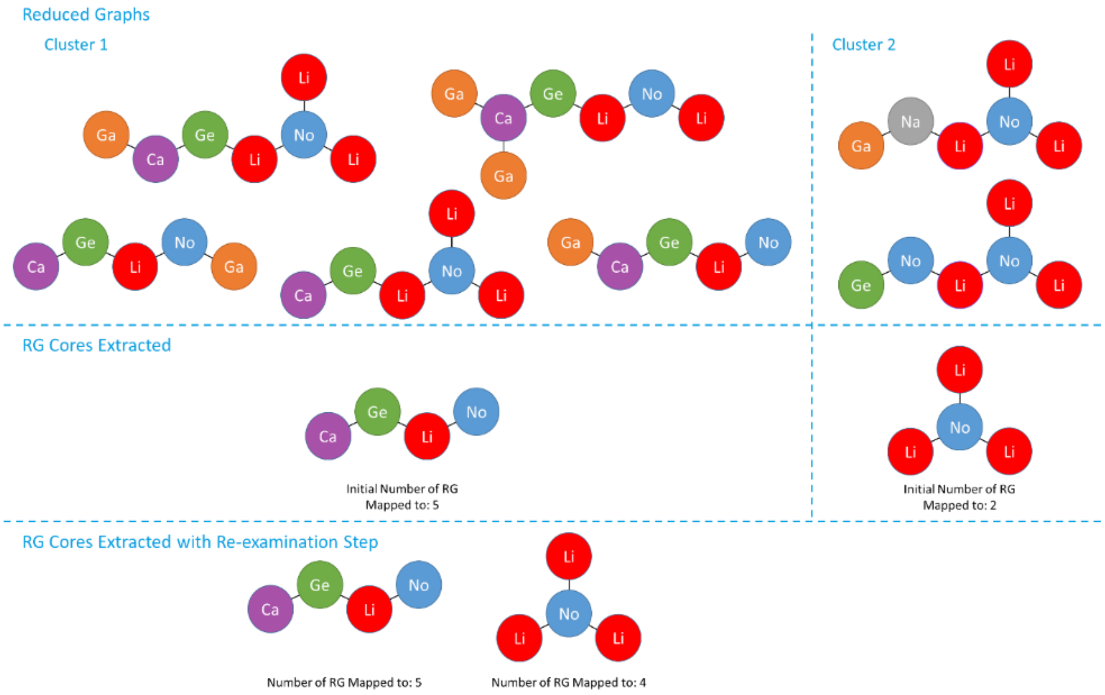

Reduced Graphs
Cluster 1
Cluster 2
RG Cores Extracted
Initial Number of RG Mapped to: 5
Initial Number of RG Mapped to: 2
RG Cores Extracted with Re-examination Step
Number of RG Mapped to: 5
Number of RG Mapped to: 4

Figure 2-15: An example of how the number of mapped molecules to a RG core can vary

2.2.7.2 Processing the Dataset as a Single Cluster

The RG core workflow was also applied to each dataset with the data considered as a single cluster. This was to see if there was any advantage in clustering the datasets or whether this step could be avoided to lower computational costs. Multiple cores were expected as the variation of the RGs will be greater than when considering a subset that has been pre-clustered.

2.3 Results and Discussion

An analysis of descriptors and clustering methods showed that M2FP and agglomerative clustering gave the best performance compared to a "manual" clustering and this was selected for future studies. However, there was variation within the dataset and no universally best method could be identified. RG cores were then extracted from these clusters.

2.3.1 Reduced Graphs

The number of unique RGs for each dataset can be seen in Table 2-4, using the default RG type. Table 2-4 also displays the average size of the molecules and the average size of the RGs. The average sizes are rounded to the nearest whole number. Table 2-4 demonstrates that by using the RG descriptor, the number of unique representations is reduced and therefore, the representation of the chemical space is more condensed. Also, the reduction in the complexity of the descriptors can be recognised by the decrease in the average sizes of the RGs compared to the molecules. Some datasets undergo more compression than others. For example, the Neurokinin dataset contains fewer molecules than the Bajorath dataset, however, the Bajorath dataset contains fewer RGs. The different degrees of data compression reflect differences in the variation in molecular structures in the dataset. The more variation in the dataset the less compression will be seen using the RGs.

Table 2-4: Table showing the number of RG generated for each dataset

Name of Dataset	Number of Molecules	Average Size of Molecule	Number of Unique RG (default)	Average Size of RG (default)
Bajorath	2549	34	920	10
CDK2	1368	28	824	8
Chk1	106	30	91	8
Cyto	6370	37	3762	8
FactorXa	1956	36	883	10
Neurokinin	2475	31	1451	9
P2x7	2259	29	822	9
P2x7 Subset	691	25	162	9
p38α	3644	30	1902	8

All the different types of RG were generated by using all combinations of parameters to identify whether there are substantial differences in the RGs. Table 2-5 shows the number of unique RGs and the average size of the RGs for the different types of RG for the p2x7 Subset dataset and demonstrates that the number of RGs varies depending on the parameters set. Table 2-5 illustrates that as the definition of the RG becomes more complex the number of unique reduced graphs increases and some RGs that previously were the same are no longer the same. There is also an increase in the average size of the RG, as most of the parameters lead to additional nodes. For this dataset, adding the metal parameter does not affect the RG as no metals are present. However, all the other parameters define the RG differently and none of the other variants produce exactly the same RGs. The impact this variation has is explored later in the chapter. The results for the other datasets are in the Appendix, where apart from two datasets, Cyto and Neurokinin, the rest of the datasets follow the same pattern. For the Cyto and Neurokinin datasets, there are also slight variations when using the metal parameter. As the metal parameter is not useful for most datasets, it is not used in any the following experiments.

Table 2-5: A comparison of the effect of the different RG parameters on the number of RG for the P2x7 subset dataset

Parameter	Number of Unique RG	Average Size of RG (rounded to the nearest whole number)
Default	162	9
Terminal	163	10
Complex	220	10
Double Bond	162	9
Metal	162	9
Terminal and Complex	222	10
Terminal and Double Bond	163	10
Terminal and Metal	163	10
Complex and Double Bond	220	10
Complex and Metal	220	10
Double Bond and Metal	162	9
Terminal, Complex and Double Bond	222	10
Terminal, Complex and Metal	222	10
Terminal, Double Bond and Metal	163	10
Complex, Double Bond and Metal	220	10
Terminal, Complex, Double Bond and Metal	222	10

2.3.2 Similarity

For each dataset, similarity matrices were calculated using both the CGs and the RGs and the different similarity methods resulting in six matrices (chemical graph cMCS, dMCS and fingerprint; reduced graph cMCS, dMCS and fingerprint).

A comparison of the data within the matrices was also carried out. Figure 2-16 shows density plots of the pairwise similarity values for all the datasets. A few distinctive features can be seen across all datasets. The RGFP peak is always shifted the most to the lower end of the Tanimoto similarity scale, which could indicate that the fingerprint might be too sparse to hold sufficient information for effective clustering. The RGFP peak is then closely followed by the M2FP, RG cMCS and then CG cMCS peaks. It is also evident that the density plots for the two dMCSs have significantly different shapes from the rest. The different shapes could indicate that the dMCS methods will give more varied results compared to the other methods that all have a much narrower range of values.

Table 2-6 shows the mean, median and mode of the different distributions for each method for each dataset. This gives some indication of the level (density) of chemical space explored. When a dataset has lower values, the molecules are more dissimilar and, therefore, more chemical space has been explored. Large similarity values indicate that lots of the molecules are similar, so a denser area of chemical space has been searched. A good example of this is when the M2FP descriptors for P2x7 Subset and P2x7 are compared. The P2x7 subset is a compact subset of the P2x7 dataset as shown by the mean, median and mode being higher for the P2x7 subset compared to the P2x7 dataset. Additionally, the relative values of the mean and median indicate the distribution of the molecules. If the mean is greater than the median, then this is a right positive skew, which means that there are some high-value similarity outliers. Whereas, when the median is greater than the mean, there is a left negative skew where there are some low-value similarity outliers.

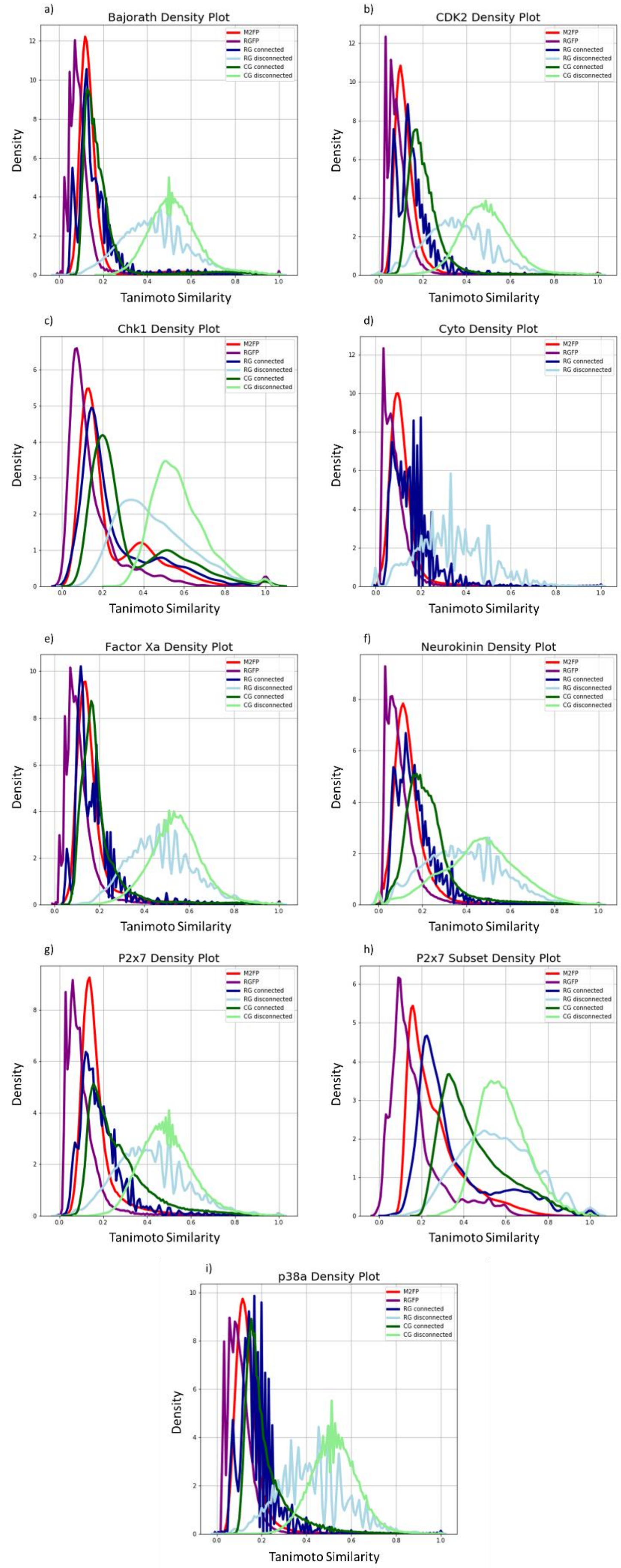


Figure 2-16: Graphs showing the different density plots of the different similarity matrices generated for the all the dataset a) Bajorath b) CDK2 c) Chk1 d) Cyto e) FactorXa f) Neurokinin g) P2x7 h) P2x7 Subset i) p38a

Table 2-6: Table showing the mean, median and mode of the similarity distribution for each of the molecular descriptors for each dataset

Dataset	Molecular Descriptor	Overall Average Pairwise Similarity	Overall Mode Pairwise Similarity	Overall Median Pairwise Similarity
Bajorath	M2FP	0.154	0.125	0.129
	RGFP	0.101	0.100	0.085
	RG Connected	0.165	0.125	0.133
	RG Disconnected	0.432	0.500	0.417
	CG Connected	0.194	0.143	0.160
	CG Disconnected	0.523	0.500	0.512
CDK2	M2FP	0.128	0.100	0.111
	RGFP	0.082	0.000	0.074
	RG Connected	0.157	0.143	0.143
	RG Disconnected	0.362	0.333	0.357
	CG Connected	0.214	0.167	0.191
	CG Disconnected	0.489	0.500	0.484
Chk1	M2FP	0.235	0.167	0.161
	RGFP	0.145	0.083	0.105
	RG Connected	0.259	0.143	0.182
	RG Disconnected	0.444	0.500	0.417
	CG Connected	0.324	0.200	0.238
	CG Disconnected	0.570	0.500	0.556
FactorXa	M2FP	0.168	0.143	0.145
	RGFP	0.111	0.100	0.098
	RG Connected	0.178	0.118	0.150
	RG Disconnected	0.472	0.500	0.462
	CG Connected	0.204	0.167	0.173
	CG Disconnected	0.541	0.500	0.540
Cyto	M2FP	0.114	0.111	0.104
	RGFP	0.074	0.000	0.067
	RG Connected	0.149	0.167	0.133
	RG Disconnected	0.333	0.333	0.333
Neurokinin	M2FP	0.139	0.125	0.125
	RGFP	0.087	0.000	0.075
	RG Connected	0.166	0.167	0.143
	RG Disconnected	0.383	0.500	0.375
	CG Connected	0.228	0.250	0.206
	CG Disconnected	0.457	0.500	0.463
P2x7	M2FP	0.171	0.143	0.148
	RGFP	0.095	0.000	0.080
	RG Connected	0.190	0.143	0.167
	RG Disconnected	0.412	0.500	0.400
	CG Connected	0.270	0.200	0.229
	CG Disconnected	0.489	0.500	0.481
P2x7 Subset	M2FP	0.267	0.167	0.228
	RGFP	0.169	0.125	0.133
	RG Connected	0.352	0.214	0.286
	RG Disconnected	0.553	0.500	0.545
	CG Connected	0.448	0.333	0.405

	CG Disconnected	0.588	0.500	0.576
P38α	M2FP	0.140	0.125	0.127
	RGFP	0.098	0.000	0.091
	RG Connected	0.174	0.143	0.154
	RG Disconnected	0.424	0.500	0.417
	CG Connected	0.217	0.167	0.182
	CG Disconnected	0.525	0.500	0.524

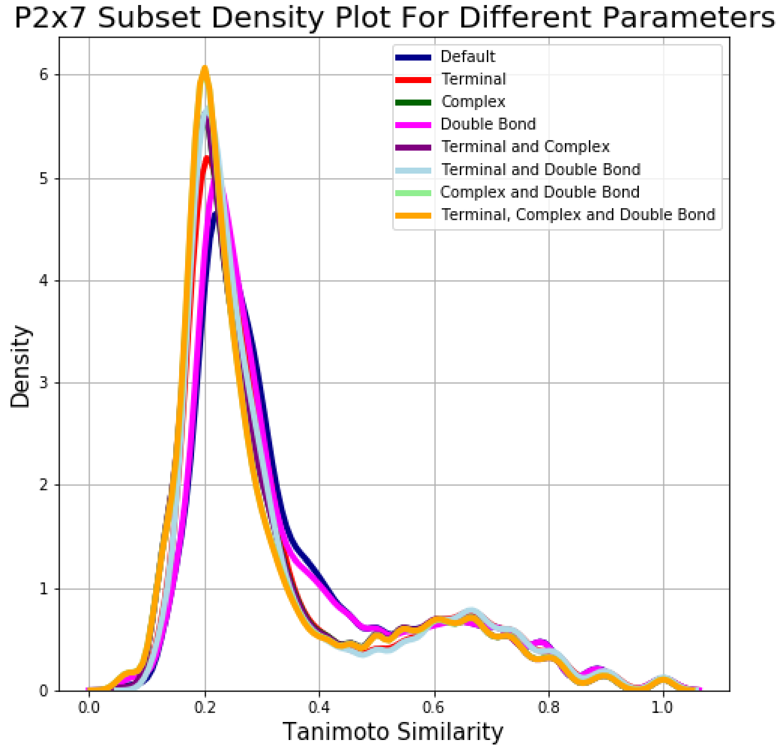


Figure 2-17: Density plot of the different RG types for the P2x7 Subset dataset using RG cMCS

The effect of the different variations of the RG was also examined. Figure 2-17 displays the distribution of similarity values for the various RG types for the P2x7 subset. Overall, the distributions all follow the same shape, but there is some slight variation as they have slightly different peak shifts, although these differences are not as large as between the different molecular descriptors. However, even these subtle differences in similarity are likely to lead to different clustering results.

2.3.3 Clustering

Clustering was carried out to provide a first level of grouping of the molecules within a dataset as it was felt that considering each cluster in turn would lead to more effective extraction of RG cores compared to extracting them from all the molecules within a dataset in a single pass. The experiments in this section aimed to identify the optimum clustering method. All possible combinations of molecular descriptors, similarity methods, and clustering algorithms were investigated for all the datasets. All of the RG types were also explored to examine the effect of these variations on the clustering results. For three of the datasets, the computer generated clusters were compared with the manual clustering. For the other six datasets, the internal indices only were considered. The first section below examines the impact of the RG types and the second explores the effect of the other molecular descriptors.

2.3.3.1 Altering the different RG parameters

The RG variants were explored to see the effect on the clustering results. The different clustering algorithms were used alongside the RG MCS similarities, cMCS and dMCS. Table 2-7 shows the results for each combination of RG parameters for the P2x7 subset dataset when clustered via agglomerative (Ward's), K-means and Butina algorithms, using RG cMCS as the similarity metric. For the agglomerative and K-means clustering, all numbers of clusters were considered from two to one-hundred and fifty and for the Butina clustering, all values between 0.1 and 1 in intervals of 0.1 were used. Table 2-7 reports the number of clusters for which the silhouette average score was a maximum for each method. For comparison, the ideal number of clusters from the manual clustering was four. In all cases, bar the Butina method, the optimum number of four clusters was found, however, some variations are seen in the silhouette average values. The silhouette average scores are similar for the agglomerative and K-means clusters which are relatively evenly populated, however, they are significantly different for the Butina clusters which are not. For example, using the default RG parameters the agglomerative and K-means clusters have 53, 44, 38 and 27 molecules in each cluster, however, the Butina clusters have 157, four and one respectively. The column headed "Number of Molecules Misplaced" gives the numbers of molecules that do not match the manual clustering, that is, although the desired number of clusters was found, the placement of molecules within the clusters was not identical to the manual clustering. The number of molecules misplaced for the agglomerative and K-means clusterings is either 16 or 17, which out of 691 molecules is a small percentage.

Table 2-7: Table showing the P2x7 subset dataset clustering results for each of the different RG parameters based on the best silhouette average, where the ideal clustering is 4 clusters. The number within the brackets for the Butina method is the similarity threshold that generated these results.

Parameter	Clustering Algorithm	Number of Clusters Selected	Silhouette Average	Number of Molecules Misplaced
Default	Agglomerative	4	0.449	16
	Butina	3 (0.2)	-0.079	438
	K-means	4	0.450	17
Terminal	Agglomerative	4	0.486	16

	Butina	4 (0.2)	-0.153	434
	K-means	4	0.487	17
Complex	Agglomerative	4	0.453	16
	Butina	10 (0.3)	-0.255	541
	K-means	4	0.453	17
Double Bond	Agglomerative	4	0.461	16
	Butina	3 (0.2)	-0.091	430
	K-means	4	0.461	16
Terminal and Complex	Agglomerative	4	0.453	16
	Butina	3 (0.2)	-0.059	514
	K-means	4	0.453	17
Terminal and Double Bond	Agglomerative	4	0.495	16
	Butina	2 (0.2)	0.003	440
	K-means	4	0.495	16
Complex and Double Bond	Agglomerative	4	0.463	16
	Butina	3 (0.2)	-0.125	478
	K-means	4	0.463	16
Terminal, Complex and Double Bond	Agglomerative	4	0.464	16
	Butina	2 (0.2)	-0.032	440
	K-means	4	0.464	16

All of the agglomerative clustering methods produced identical clusters, regardless of the RG type used. This can be seen by the number of molecules misplaced from the manual clustering being the same value throughout Table 2-7. When inspecting the K-means clusters, there are some minor changes in cluster assignments, with one molecule moving clusters depending on the reduced graph type, as shown in Figure 2-18. The Butina clustering does not give promising results as the silhouette average is negative or very low and the number of misplaced molecules is very high.

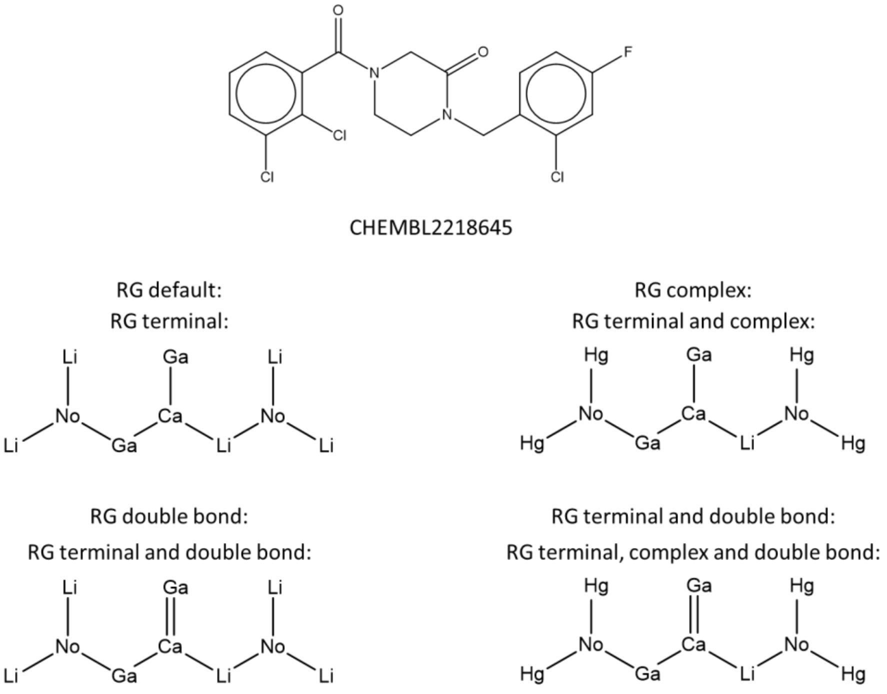


Figure 2-18: ChEMBL2218645 molecule varies in cluster assignment in P2x7 subset when varying the RG parameters. Where RGs were created by different parameters.

Even though the agglomerative clusters are the same in all cases, they do not have identical silhouette average scores, and this is due to the differences in the similarity matrices generated from the different types of RGs. This means that the silhouette average score cannot be used to directly compare the clusters generated from the different molecular descriptors, or even within the different RG types, as they have different similarity matrices.

Table 2-8 shows the result with the highest silhouette average score for the different RGs and clustering algorithms for all datasets. Butina clustering consistently underperforms the other clustering techniques for each of the RG types, indicated by the silhouette average scores always being negative or very low and when compared to the manual clusterings the misplacement of molecules is extensive.

When considering all the datasets, greater variation was seen in the clusterings produced for the different RGs and different clustering algorithms than was seen for the P2x7 subset dataset. The agglomerative clusterings show slight variations across the different RGs, however, the variation is more pronounced for the K-means clusterings. Finally, the differences between the agglomerative and K-means methods are much larger than was seen for P2x7 subset. Some of these variations are caused by the presence of singletons for the agglomerative method, which are uncommon for the K-means method. The dataset that shows the most variation across the different RGs and clustering methods is the Neurokinin dataset. The overlap between the different clusters for this dataset is shown as a pairwise overlap heatmap in Figure 2-19. The overlap indicates how many molecules are contained within the same cluster across the methods that are being compared. This is calculated using the cluster purity method choosing one of the clustering techniques as a reference. The lighter the colour the higher the degree of similarity and overlap, the darker the colour the lower the degree of similarity and overlap. Therefore, it can easily be seen that the Butina clusters differ from the agglomerative and K-means clusters. The lightest colours are observed when comparing different RG parameters for the same clustering method. The overlap between the agglomerative and k-means results is around fifty percent. These results show that there is close alignment between the differing RG parameters but there is variation between different clustering methods. Additionally, these results indicate that the clusters from the Butina methodology vary more than the agglomerative and K-means. The results for the other datasets are within the Appendix.

Table 2-8: The number of clusters for the clustering with the largest silhouette average for each parameter and clustering algorithm combination for all datasets.

Parameter	Clustering Algorithm	Bajorath		CDK2		Chk1		Cyto		FactorXa		Neurokinin		P2x7		P38α	
		Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average	Number of Clusters	Silhouette Average
Default	Agglomerative	30	0.474	146	0.330	3	0.363	135	0.123	92	0.378	145	0.218	119	0.357	149	0.252
	Butina	4 (0.1)	-0.043	5 (0.1)	-0.059	2 (0.1)	-0.038	7 (0.1)	-0.074	2 (0.1)	0.008	4 (0.1)	-0.044	3 (0.1)	-0.019	4 (0.1)	-0.057
	K-means	28	0.457	107	0.293	3	0.363	102	0.101	119	0.310	136	0.183	41	0.288	150	0.228
Terminal	Agglomerative	29	0.455	150	0.334	3	0.332	2	0.117	112	0.358	148	0.210	120	0.364	147	0.256
	Butina	4 (0.1)	-0.025	3 (0.1)	-0.016	5 (0.2)	-0.191	9 (0.1)	-0.113	3 (0.1)	-0.038	4 (0.1)	-0.018	4 (0.1)	-0.073	4 (0.1)	-0.057
	K-means	27	0.441	111	0.285	4	0.338	100	0.103	93	0.303	106	0.167	55	0.303	112	0.222
Complex	Agglomerative	29	0.458	150	0.334	3	0.333	2	0.112	113	0.356	144	0.210	103	0.354	150	0.253
	Butina	6 (0.1)	-0.086	3 (0.1)	-0.011	2 (0.1)	0.041	6 (0.1)	-0.065	3 (0.1)	-0.045	6 (0.1)	-0.018	3 (0.1)	-0.017	5 (0.1)	-0.045
	K-means	28	0.457	115	0.297	4	0.335	116	0.104	90	0.314	131	0.181	38	0.286	138	0.234
Double Bond	Agglomerative	30	0.476	149	0.329	3	0.363	150	0.124	103	0.378	149	0.205	123	0.356	141	0.262
	Butina	5 (0.1)	-0.074	5 (0.1)	-0.076	5 (0.2)	-0.110	5 (0.1)	-0.063	3 (0.1)	-0.026	6 (0.1)	-0.015	2 (0.1)	-0.009	4 (0.1)	-0.058
	K-means	28	0.458	107	0.286	3	0.363	83	0.104	76	0.310	145	0.181	47	0.293	130	0.225
Terminal and Complex	Agglomerative	29	0.459	148	0.333	3	0.332	2	0.112	113	0.357	149	0.215	106	0.354	145	0.249
	Butina	6 (0.1)	-0.068	4 (0.1)	-0.050	2 (0.1)	-0.029	7 (0.1)	-0.102	4 (0.1)	-0.079	6 (0.1)	-0.047	4 (0.1)	-0.079	4 (0.1)	-0.067

	K-means	30	0.448	125	0.287	4	0.335	85	0.103	76	0.311	113	0.178	40	0.291	150	0.237
Terminal and Double Bond	Agglomerative	29	0.457	150	0.329	3	0.332	2	0.116	117	0.362	149	0.209	120	0.364	147	0.260
	Butina	6 (0.1)	-0.089	4 (0.1)	-0.017	2 (0.1)	0.046	6 (0.1)	-0.077	3 (0.1)	-0.034	4 (0.1)	-0.019	3 (0.1)	-0.011	3 (0.1)	-0.023
	K-means	29	0.451	104	0.274	4	0.338	98	0.105	87	0.313	126	0.168	39	0.295	144	0.210
Complex and Double Bond	Agglomerative	29	0.459	136	0.329	3	0.333	2	0.109	118	0.360	149	0.208	88	0.356	150	0.252
	Butina	7 (0.1)	-0.105	4 (0.1)	-0.020	2 (0.1)	-0.036	6 (0.1)	-0.083	3 (0.1)	-0.029	7 (0.1)	-0.056	6 (0.1)	-0.104	4 (0.1)	-0.040
	K-means	29	0.451	127	0.280	4	0.400	90	0.103	76	0.309	147	0.184	37	0.287	150	0.229
Terminal, Complex and Double Bond	Agglomerative	29	0.459	137	0.327	3	0.332	2	0.097	118	0.361	149	0.216	87	0.355	150	0.247
	Butina	8 (0.1)	-0.094	4 (0.1)	-0.035	2 (0.1)	-0.036	7 (0.1)	-0.087	3 (0.1)	-0.033	4 (0.1)	-0.019	4 (0.1)	-0.039	3 (0.1)	-0.012
	K-means	28	0.453	144	0.285	4	0.335	9	0.089	77	0.313	116	0.177	33	0.295	123	0.225

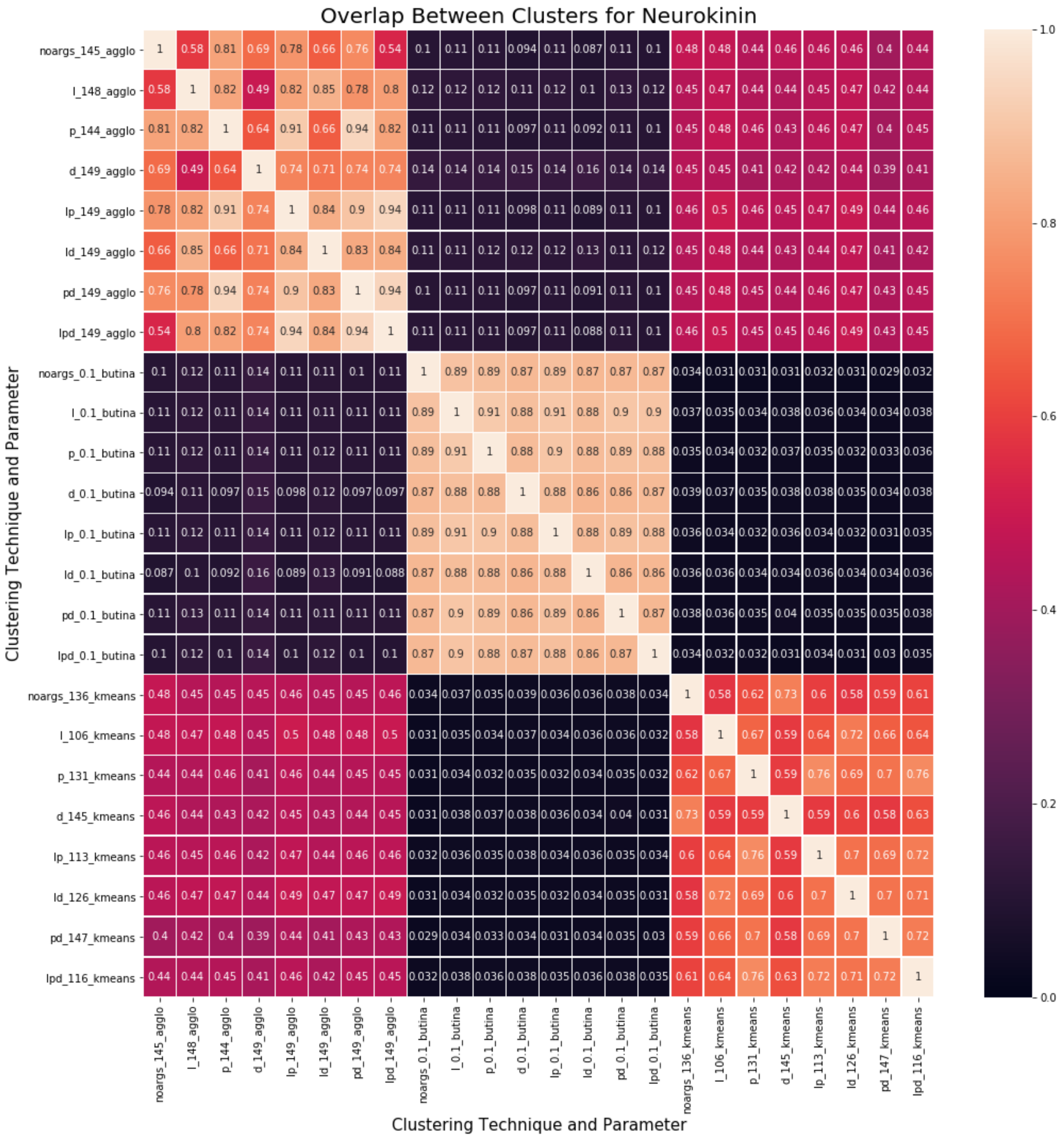


Figure 2-19: Neurokinin heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap

Given that in general the ideal clustering is not known, it would be desirable to have a method to compare the different clusterings that is independent of the values within the similarity matrices.

To compare how 'well' the clustering techniques had performed, the clusterings were compared to the manual clusterings generated for the P2x7 subset, the P2x7 and the Bajorath datasets. The P2x7 subset has four manual clusters; the P2x7 dataset has sixty-one manual clusters; and the Bajorath dataset has twenty-nine manual clusters. Two different external indexes were calculated: the purity and v-measure. The results for P2x7 can be seen in Table 2-9, which clearly shows variations in the purity and v-measures. The technique that has the highest purity and v-measure score is the default RG with agglomerative clustering. The Butina clusterings consistently have very low purity and v-measure scores. The results for the P2x7 subset and the Bajorath dataset can be found in the Appendix.

Table 2-9: Table showing the comparison of the Purity and V-measures for P2x7, the appropriate number of clusters is sixty-one

Parameter	Clustering Algorithm	Number of Clusters Extracted	Silhouette Average	Purity	V-Measure
Default	Agglomerative	119	0.357	0.926	0.904
	Butina	3 (0.1)	-0.019	0.127	0.017
	K-means	41	0.288	0.836	0.863
Terminal	Agglomerative	120	0.364	0.917	0.894
	Butina	4 (0.1)	-0.073	0.127	0.009
	K-means	55	0.303	0.885	0.869
Complex	Agglomerative	103	0.354	0.913	0.897
	Butina	3 (0.1)	-0.017	0.132	0.036
	K-means	38	0.286	0.842	0.869
Double Bond	Agglomerative	123	0.356	0.925	0.902
	Butina	2 (0.1)	-0.009	0.123	0.010
	K-means	47	0.293	0.856	0.870
Terminal and Complex	Agglomerative	106	0.354	0.913	0.896
	Butina	4 (0.1)	-0.079	0.131	0.025
	K-means	40	0.291	0.838	0.854
Terminal and Double Bond	Agglomerative	120	0.364	0.917	0.894
	Butina	3 (0.1)	-0.011	0.143	0.040
	K-means	39	0.295	0.849	0.866
Complex and Double Bond	Agglomerative	88	0.356	0.887	0.886
	Butina	6 (0.1)	-0.104	0.130	0.027
	K-means	37	0.287	0.830	0.859
Terminal, Complex and Double Bond	Agglomerative	87	0.355	0.887	0.886
	Butina	4 (0.1)	-0.039	0.127	0.034
	K-means	33	0.295	0.826	0.859

Unfortunately, the maximum silhouette score does not always correspond to the maximum purity or v-measure score for any of the manually clustered dataset. Therefore, just taking the highest silhouette average is not possible. Although it is interesting to note that the purity and v-measure scores are always higher for the agglomerative clusters compared to the K-means clusters for the two larger datasets (see Appendix).

The Butina clustering was not considered further since it did not perform well for any of the RG types for any of the datasets. When comparing the results of the agglomerative and K-means methods, the agglomerative method results in more singletons, whereas, the K-means clusters are generally larger and have higher average cluster similarity. The agglomerative clusters are more variable in size.

The best RG types and clustering methods as measured using the largest Silhouette average, the largest purity and v-measures, and the smallest number of misplaced molecules for the hand clustered datasets are presented in the Appendix. The RG types that repeatedly perform the best are the default RG and the RG with terminal parameter. These two RG types consistently score the highest or are among the top performers for each of the different indexes explored. As the default RG is the simplest this was used hereon unless stated otherwise.

2.3.3.2 Altering the molecular descriptors

The clusters generated using the M2FP, RG, RGFP and chemical graph descriptions were then investigated with the internal cluster indices calculated in all cases. Two exceptions are datasets Cyto and p38α. The calculation of the similarity matrices for the chemical graphs timed out for Cyto. Additionally, the calculation of the cluster indexes for the chemical graphs timed out for p38α.

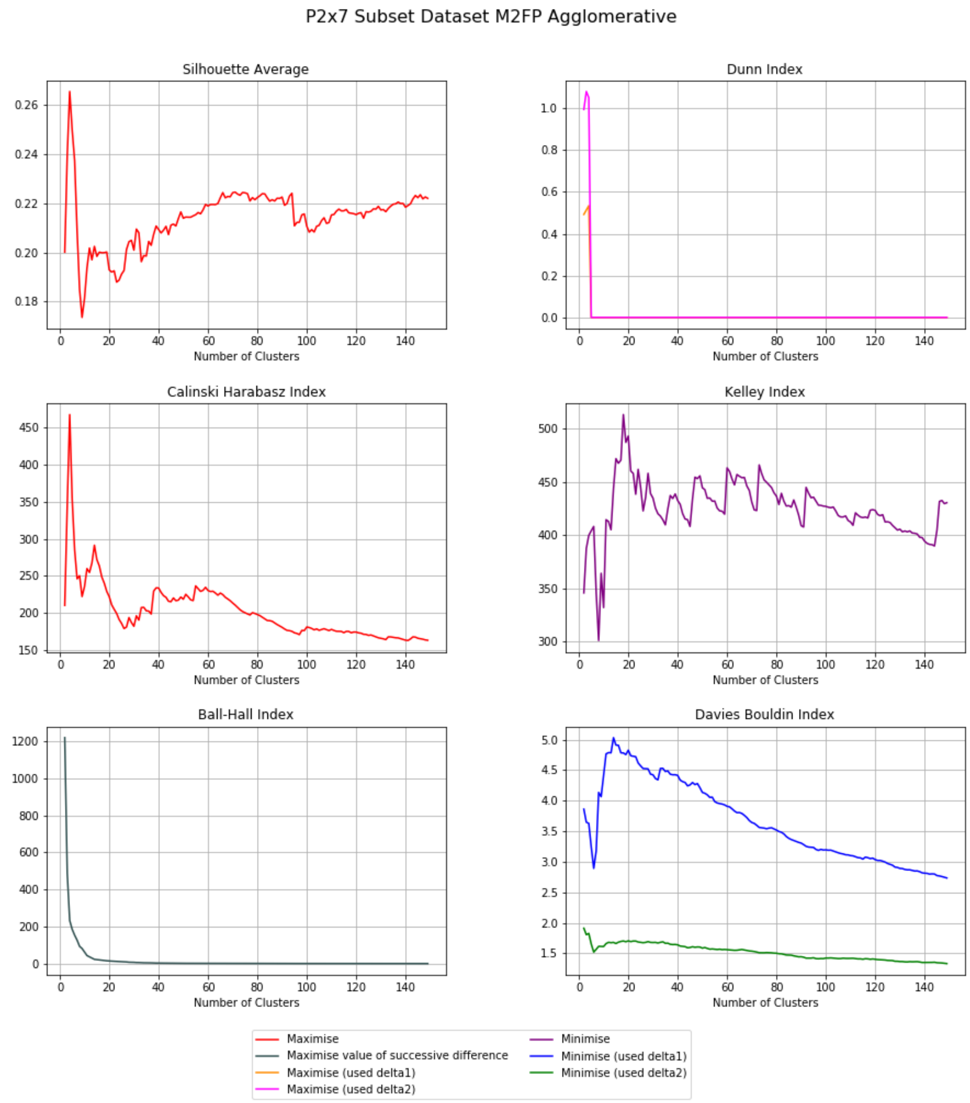


Figure 2-20: P2x7 subset results for the M2FP agglomerative data. The line's colour indicates whether the interest is in finding a maximum or minimum for this index, or the maximise difference i.e. the elbow point. Additionally, it also tells for some indexes whether delta 1 or delta 2 was used in the calculations.:

The internal clustering indexes were then calculated for all clustering levels. For example, Figure 2-20 displays all of the results from 2 to 150 clusters for the P2x7 subset data using the M2FP molecular descriptor and agglomerative clustering algorithm. For the silhouette average, Dunn index and Calinski-Harabasz indexes the clustering with the highest value was selected; for Kelley and Davies Bouldin indexes the clustering with the lowest value was selected; and for the Ball-Hall index the elbow point was chosen. For the P2x7 Subset example, the optimum number of clusters for some methods and clustering indexes corresponds to the ideal (i.e., the number of clusters identified manually). However, the obtained number of clusters did not conform to the ideal clustering for all the different descriptors and clustering indexes for this dataset.

The “best” result for each internal index is reported in Table 2-10 together with the number of clusters generated for each molecular descriptor and clustering algorithm. The number of clusters and cluster index values are also plotted in Figure 2-21. There is no consensus on the optimum number of clusters across the indexes. It was thought that the indexes may converge and be in agreement with the ideal number of clusters to give an overall consensus. Some are in agreement, however, not all of them are. Also, no one index always agreed with the ideal number of clusters as determined by the manual clusterings.

Table 2-10: P2x7 subset table of the results from the clustering validity indexes

		Silhouette [max]		Dunn delta 1 [max]		Dunn delta 2 [max]		Davies Bouldin delta 1 [min]		Davies Bouldin delta 2 [min]		Calinski Harabasz [max]		Ball-Hall [elbow]	Kelley [min]	
Molecular Descriptor	**Clustering Algorithm**	**Clusters**	**Value**	**Clusters**	**Value**	**Clusters**	**Value**	**Clusters**	**Value**	**Clusters**	**Value**	**Clusters**	**Value**	**Clusters**	**Clusters**	**Value**
M2FP	Agglomerative	4	0.265	4	0.534	3	1.080	149	2.734	149	1.333	4	467.883	11	8	300.732
	K-means	3	0.514	3	0.514	3	1.080	145	3.472	149	1.540	19	491.380	9	55	159.252
RG (default) cMCS	Agglomerative	4	0.449	2	0.615	6	1.182	63	2.915	63	1.099	6	276.257	6	4	76.477
	K-means	4	0.450	2	0.615	7	1.166	60	2.934	56	1.128	5	277.675	6	27	40.291
RG (default) dMCS	Agglomerative	55	0.390	9	0.183	7	1.128	61	2.116	61	0.843	6	95.366	6	28	67.565
	K-means	64	0.421	29	0.249	2	1.094	64	2.371	64	1.142	4	139.921	6	14	43.716
RGFP (default)	Agglomerative	19	0.259	2	0.648	2	1.029	51	1.351	51	1.057	46	169.798	9	10	59.725
	K-means	15	0.264	23	0.627	17	1.009	54	1.563	51	1.222	55	337.835	8	6	51.708
Chemical graph cMCS	Agglomerative	5	0.380	2	0.445	6	1.050	149	3.160	149	1.059	6	558.872	9	3	293.546
	K-means	4	0.381	4	0.290	5	1.075	149	5.229	138	1.416	5	676.618	9	138	142.219
Chemical graph dMCS	Agglomerative	144	0.374	7	0.125	2	1.066	149	3.810	149	1.127	144	269.277	13	69	344.851
	K-means	24	0.287	10	0.068	5	1.099	149	5.799	148	1.380	12	307.691	12	15	195.405

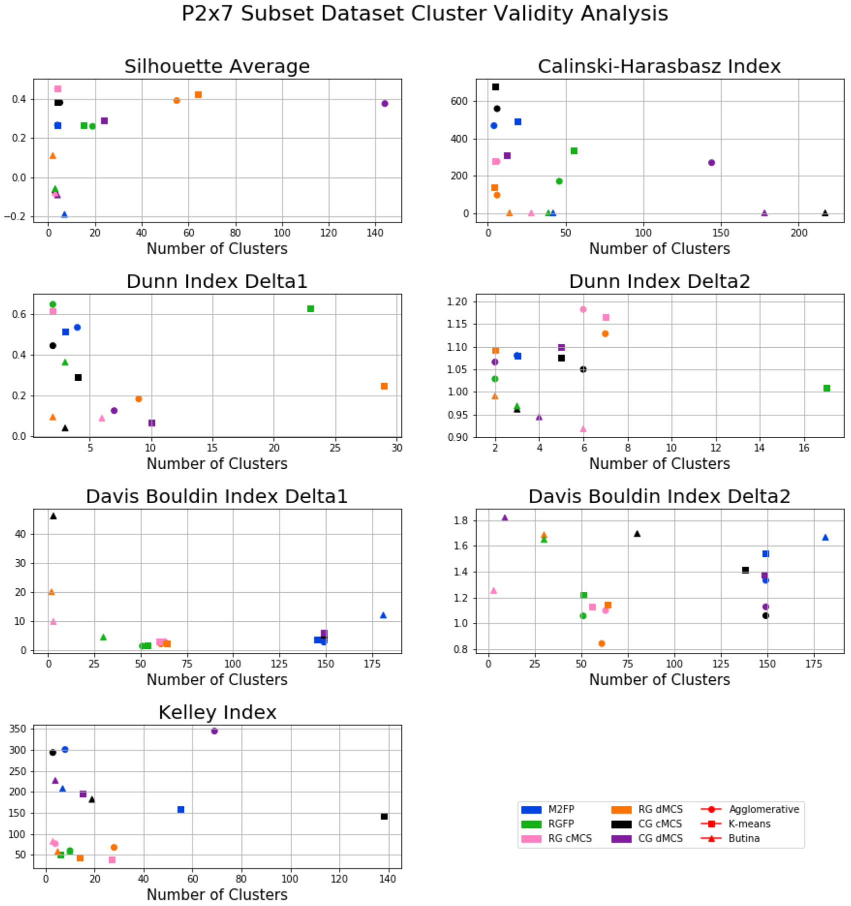


Figure 2-21: Graphs showing all the most appropriate results from all of the different combinations of molecular descriptors and clustering for the P2x7 subset dataset

The overlap between the resulting clusters was investigated to understand how the results from the different combinations of molecular descriptors and clustering algorithm overlap. The clusters that had the largest silhouette average were compared between methods. The closer to one the more molecules were in the same cluster. A heatmap was generated to show the overlaps, Figure 2-22. It demonstrates that there tends to be considerable agreement between the agglomerative and K-means method for each molecular descriptor. It also indicates that there is not much agreement between the connected MCS and FP similarity and the disconnected MCS. This suggests that the disconnected MCS version is not the ideal methodology as it is an outlier amongst the similarity methods explored. Additionally, there is some variation across different descriptors, which indicates that the clustering method depends on the molecular descriptor of choice.

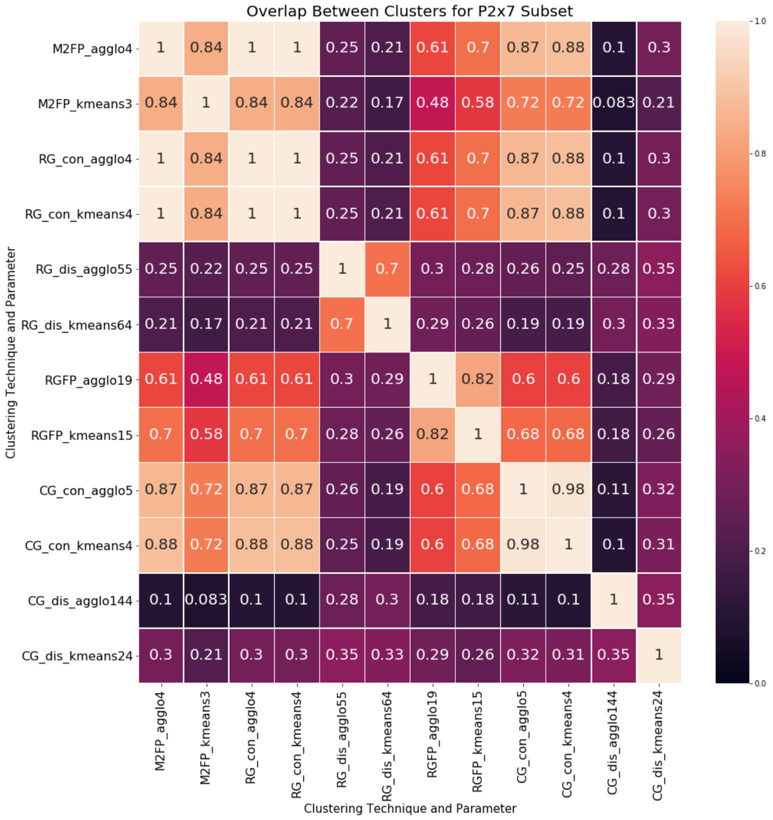


Figure 2-22: Heatmap showing the overlap between clusters for each combination of molecular descriptor and clustering algorithm used for P2x7 Subset dataset

As previously stated, the clustering validity indexes for the different representations cannot be compared directly. Therefore, two external indexes were calculated for the clusterings that achieved the highest silhouette score within each workflow. These were cluster purity and the v-measure. The clusterings that contained more clusters than the manual clusters had higher scores for the purity scores. The higher scores are because purity calculates the misclassification rate per cluster and then averages over all clusters: the extreme case, when all clusters are singletons, gives the maximum value of 1. Thus, if many clusters consist of sub-clusters of the ideal clusters, then a larger score is generated as most of the molecules are similarly correlated to the original clusters. Conversely, the v-measure score is in line with expected values, Table 2-11.

Table 2-11: P2x7 subset different molecular descriptors purity and v-measures

Molecular Descriptor	Clustering Algorithm	Number of Clusters	Silhouette Score	Purity	V-measure
M2FP	Agglomerative	4	0.265	0.975	0.937
	K-means	4	0.265	0.974	0.935
	Agglomerative	4	0.449	0.977	0.941

RG (default) connected	K-means	4	0.450	0.975	0.934
RG (default) disconnected	Agglomerative	2	0.334	0.919	0.487
	K-means	146	0.258	0.926	0.468
RGFP (default)	Agglomerative	19	0.259	0.983	0.712
	K-means	15	0.264	0.973	0.707
Chemical graph connected	Agglomerative	5	0.380	0.858	0.713
	K-means	4	0.381	0.857	0.725
Chemical graph disconnected	Agglomerative	144	0.374	0.968	0.427
	K-means	24	0.287	0.910	0.494

To summarise all the results, the two descriptors and clustering methods that yielded the best results overall are the M2FP and RG connected MCS with agglomerative clustering, Table 2-12. These were selected as they generate the highest scores for the v-measure. The M2FP molecular descriptor and agglomerative clustering method are therefore used for the following steps as, overall, this combination give the highest purity and v-measures and is quicker to calculate. The cluster with the largest silhouette score is then used, as in general the purity and v-measure cannot always be calculated as the known clustering is not always known. Results for the other datasets are in the Appendix.

Table 2-12: The top two results for the hand clustered datasets for their purity and v-measure scores

Dataset	**Top Results**				**Second Top Result**			
	Clustering Algorithm	**Molecular Descriptor**	**Purity**	**V-Measure**	**Clustering Algorithm**	**Molecular Descriptor**	**Purity**	**V-Measure**
Bajorath	Agglomerative	M2FP	0.994	0.972	Agglomerative	RG (default) connected	0.994	0.959
P2x7	Agglomerative	RG (default) connected	0.926	0.904	K-means	RG (default) connected	0.836	0.863
P2x7 Subset	Agglomerative	RG (default) connected	0.977	0.941	Agglomerative	M2FP	0.975	0.937

2.3.4 Reduced Graph Core Extraction

The RG cores were extracted for each cluster and also for the whole dataset. Both methods were investigated to identify if there was an advantage to either methodology.

2.3.4.1 Extraction the RG Cores

When extracting the RG cores two settings have to be specified. These are the minimum core size, that is, the minimum number of nodes a core can have, and a similarity cut-off that establishes the number of neighbours in the first step of the algorithm when the initial MCS is identified. These two settings were altered for the P2x7 Subset dataset to see the effect on the cores generated. The minimum core size was varied from 2 to 7 and the similarity threshold was varied between 0.1 and 0.9. The four clusters generated using M2FP and agglomerative clustering were used, and the cores were found using the default RG parameters.

Table 2-13: Table showing the number of cores extracted for the two different settings for P2x7 Subset for M2FP agglomerative 4 clusters

		Minimum Core Size					
		2	**3**	**4**	**5**	**6**	**7**
Similarity	**0.1**	4	4	6	7	13	16
	0.2	4	4	6	9	12	22
	0.3	4	4	6	8	13	22
	0.4	4	4	6	9	11	16
	0.5	4	4	6	8	13	16
	0.6	4	4	6	8	13	18
	0.7	4	4	6	8	14	20
	0.8	4	4	6	7	13	22
	0.9	4	4	6	7	16	28

Table 2-13 shows the total number of RG cores extracted from all of the clusters for the P2x7 Subset dataset for different values of the two variables. When the minimum core size was set to two or three, the number of RG cores was four for all similarity threshold values; the four cores were the same in all cases; and one core was generated for each cluster. However, the number of cores increases when the minimum core size is increased above 3. This was expected because when the minimum core size is small, it is likely that an MCS can be found that is common to a wide variety of molecules, for example, a linker node and an aromatic hydrogen bond acceptor node. As the minimum core size is increased, the number of molecules that share an MCS is likely to decrease, so therefore, more cores are extracted. Additionally, when the similarity threshold increases, the number of neighbouring molecules decreases. Therefore, the initial MCSs are typically larger. However, as the initial MCS is compared to all the molecules in the cluster, it is typically reduced in size to represent more molecules. Thus, there is some variation in the number of RG cores extracted for each combination of the settings.

In general, there is a need to balance the two settings. If both settings are too low, then the RG cores will be small and generic and could be typical of many molecules and not specific to the molecules within a dataset. Such RG cores are not likely to represent the structure activity relationships in the data. However, if the settings are too large then the cores will be too specific and may only represent a few molecules and could even represent one molecule only, i.e., a singleton. Figure 2-23 shows an example where the minimum core size is too large for the P2x7 Subset dataset. In this case, the algorithm fails to generate a core, as the initial MCS is smaller than the minimum size required. To prevent a scenario where an initial MCS is not found, the method has been adapted so that an RG core is always extracted for a cluster. The adaption is if the furthest neighbour of the centroid does not generate an appropriate MCS, then the next furthest neighbour is considered, and this is repeated iteratively until an appropriate MCS is found. If an MCS is still not found using this iterative process, then the largest MCS that occurs most frequently is used. An example of how this methodology operates is shown using examples from Figure 2-23. The first centroid is found, RG1, the next step is to identify the furthest neighbour within the minimum similarity threshold, RG3. The MCS between RG1 and RG3 is then found and as this consists of three nodes and is lower than the minimum core size, it is put into a “reserve” list and the next furthest neighbour is examined. In this case, this is RG6 which generates an MCS with five nodes.

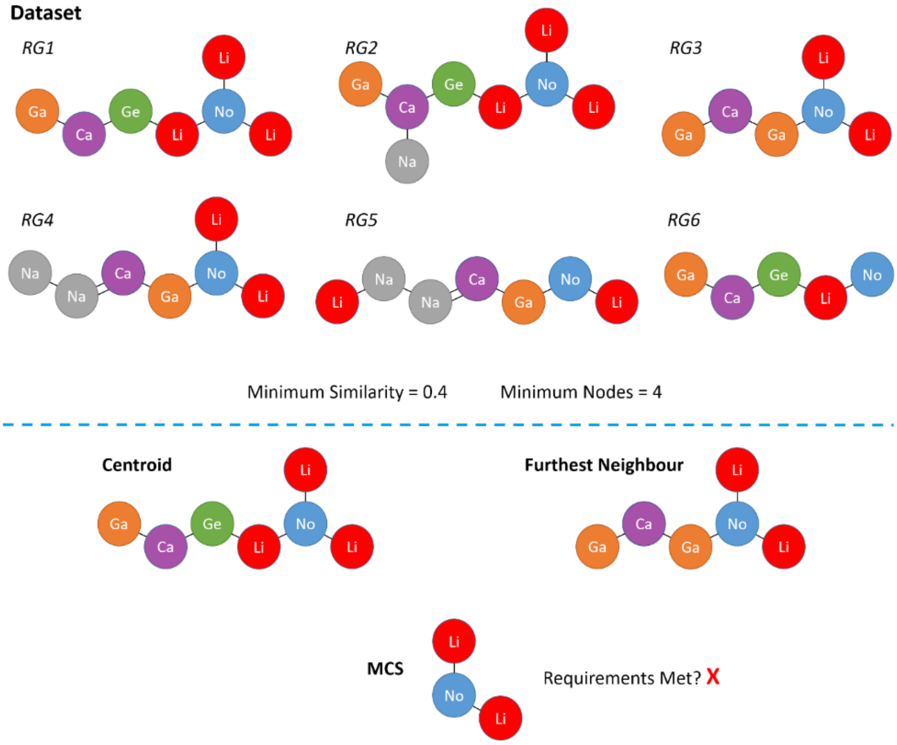
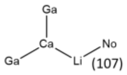
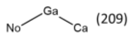
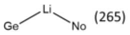
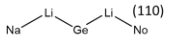
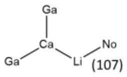
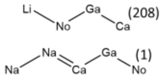
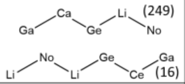
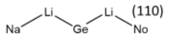
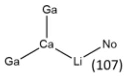
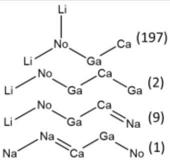
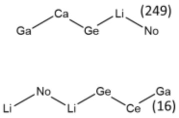
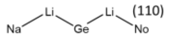
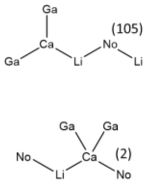
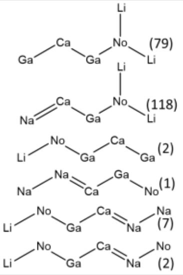
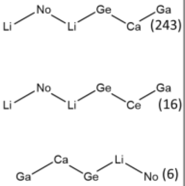
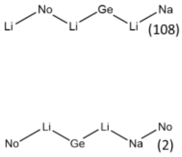
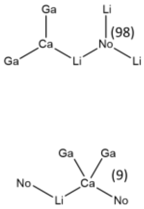
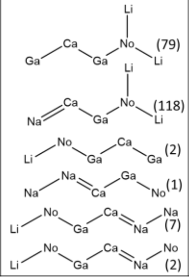
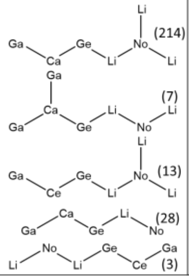
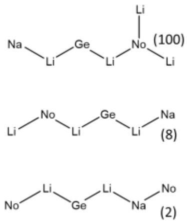


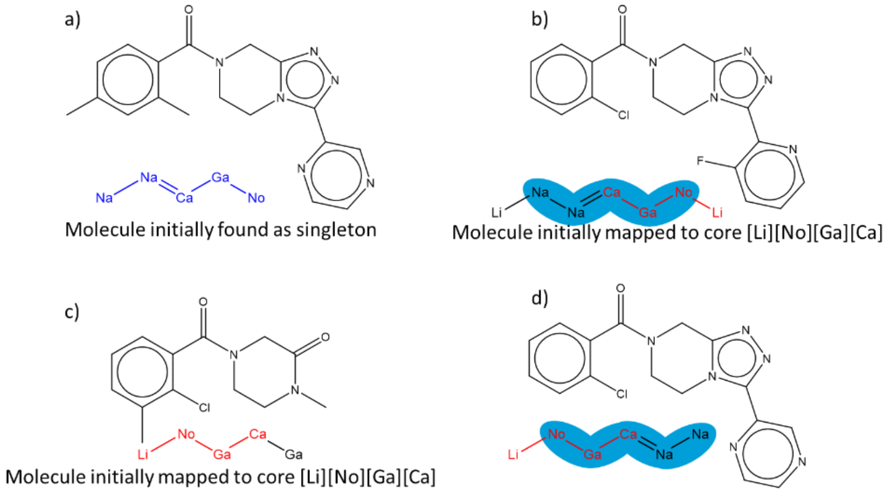
Figure 2-23: An example of an MCS core that does not meet the requirements

The cores that are extracted can change dramatically for some clusters depending on the settings. Table 2-14 shows how the results can vary as the minimum core size is changed for a constant similarity threshold of 0.5. Cluster two and three illustrate a wide range of variation in the generated cores as the minimum core size increases, whereas clusters one and four give more consistent results. The variation seen for clusters two and three is likely due to the smaller average size of the RGs, which is eight nodes compared to nine nodes for clusters one and four, and the lower average pairwise similarities. Minor variations were seen when the minimum similarity threshold was varied.

Table 2-14: RG cores that are extracted for each cluster. The value in brackets demonstrates how many molecules this core represents, for M2FP agglomerative 4 clusters

Minimum Core Size, Similarity Settings	Cluster 1	Cluster 2	Cluster 3	Cluster 4
2, 0.5	 (107)	 (209)	 (265)	 (110)
4, 0.5	 (107)	 (208) (1)	 (249) (16)	 (110)
5, 0.5	 (107)	 (197) (2) (9) (1)	 (249) (16)	 (110)
6, 0.5	 (105) (2)	 (79) (118) (2) (1) (7) (2)	 (243) (16) (6)	 (108) (2)
7, 0.5	 (98) (9)	 (79) (118) (2) (1) (7) (2)	 (214) (7) (13) (28) (3)	 (100) (8) (2)

Cluster 2



Molecule initially found as singleton

Molecule initially mapped to core [Li][No][Ga][Ca]

Molecule initially mapped to core [Li][No][Ga][Ca]

Molecule initially mapped to core [Li][No][Ga][Ca]

Figure 2-24: A selection of molecules and their RGs and the extracted RG cores from cluster two, highlighted core in RG in red

Figure 2-24 illustrates four of the molecules and their RGs in cluster two and their initial assigned RG core. Through the re-examination step, Section 2.2.7.1, after the RG core extraction process, the number of molecules associated with the RG cores changes. The initial singleton found by molecule a) increases as molecules b) and d) also contain the RG core [Na][Na]=[Ca][Ga][No], highlighted in blue.

The higher values set for the minimum core size result in a large number of RG cores many of which represent a relatively small number of molecules (Table 2-14). Therefore, the lower values of 2 or 3 are preferred for this dataset. Additional datasets are considered below.

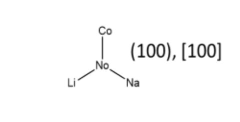
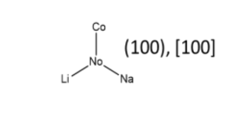
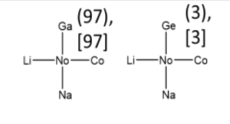
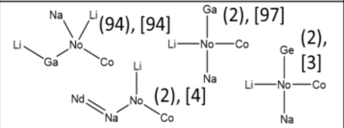
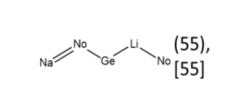
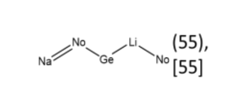
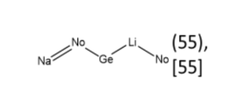
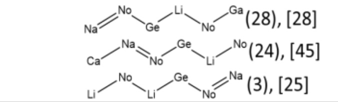
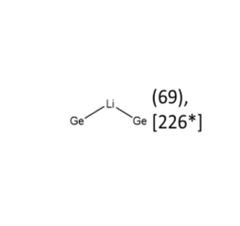
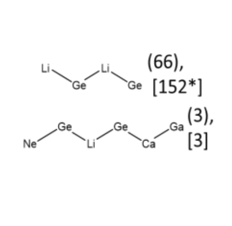
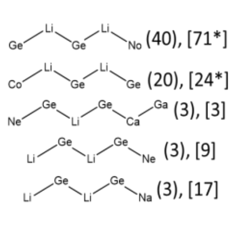
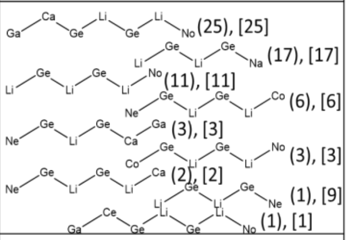
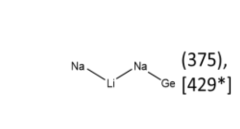
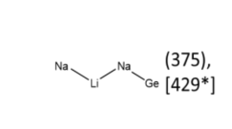
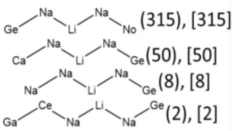
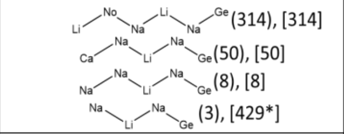
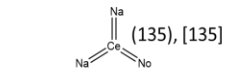
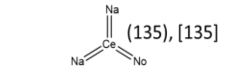
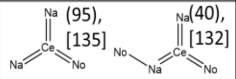
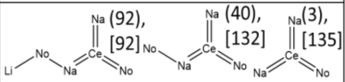
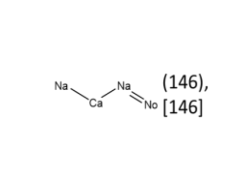
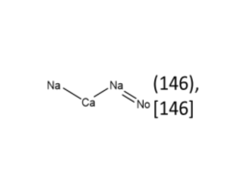
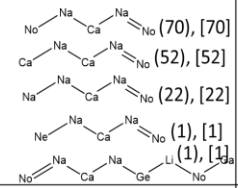
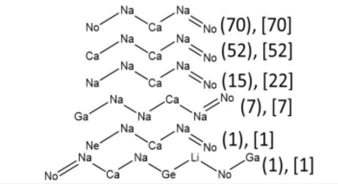
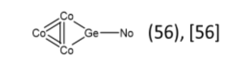
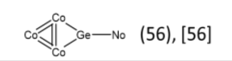
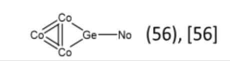
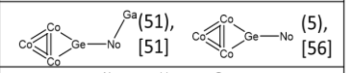
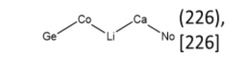
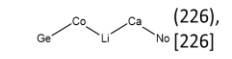
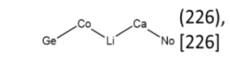
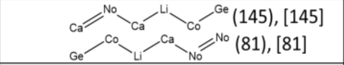
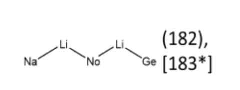
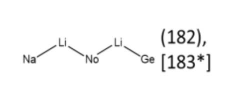
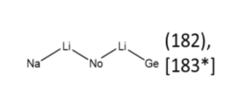
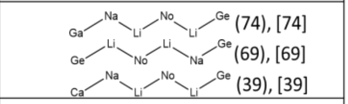
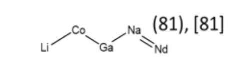
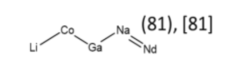
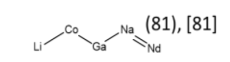
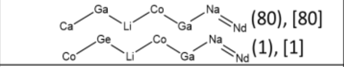
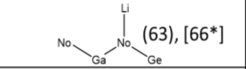
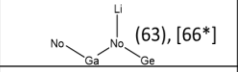
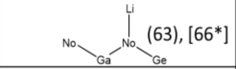
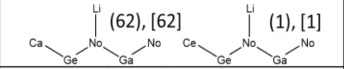
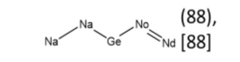
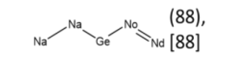
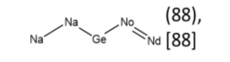
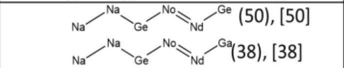
Table 2-15: Table showing the number of cores extracted for the two different settings for Bajorath for M2FP agglomerative 28 clusters

		Minimum Core Size					
		2	**3**	**4**	**5**	**6**	**7**
Similarity	**0.1**	28	28	29	39	60	88
	0.2	28	28	29	39	60	88
	0.3	28	28	29	39	59	88
	0.4	28	28	29	38	56	80
	0.5	28	28	29	41	58	74
	0.6	28	28	29	41	60	74
	0.7	28	28	29	43	63	80
	0.8	28	28	29	41	66	92
	0.9	28	28	29	41	83	130

RG cores were extracted for the Bajorath dataset for different combinations of settings based on the M2FP agglomerative clustering, which produced twenty-eight clusters. The numbers of RG cores are shown in Table 2-15. As for the P2x7 subset there is little to no variation in the number of RG cores extracted for the different similarity settings for low values of the minimum core size. When the larger minimum core sizes are explored, more variation is seen. However, some of the results were unexpected as it was thought that as the similarity threshold increased the number of RG cores would also increase, but this was not always the case.

Table 2-16 shows the RG cores that are extracted for a selection of settings. The results were the same for minimum core sizes of two and three and so three is omitted. Only those clusters are shown for which the RG cores vary with the different settings. The table shows how the RG cores are extended as the minimum number of nodes increases. For cluster 1, when the minimum core size is increased from four to five, the number of RG cores increases from one to two, and increases again to four when the minimum core size is six. The values within the parentheses, (), are the numbers of molecules from which the RG cores are generated, whereas, the values in the square brackets, [], are the numbers of molecules mapped to the RG core when all molecules are mapped back to the RG cores through the re-examination step. An asterisk, *, can be seen in cluster five, this indicates that the RG core has been extracted from several clusters. There are twelve RG cores that have been found in different clusters within the Bajorath dataset. Overall there is little variation in the number of RG cores or the nature of the RG cores for most of the clusters. It should be noted, however, that the Bajorath dataset is a fabricated dataset that was formed to help to evaluate the progress in LO datasets which likely explains this behaviour.

Table 2-16: RG cores that are extracted for each cluster from Bajorath dataset for M2FP agglomerative twenty-eight clusters. The value in parentheses demonstrates how many molecules this core relates to initially, the value in the square brackets demonstrates how many molecules relate to the core when the RG cores are mapped back onto the dataset and * indicates the cores that are seen across different clusters.

Cluster	Minimum Core Size, Similarity Setting			
	2, 0.5	4, 0.5	5, 0.5	6, 0.5
1	 (100), [100]	 (100), [100]	 (97), [97] (3), [3]	 (94), [94] (2), [97] (2), [3] (2), [4]
4	 (55), [55]	 (55), [55]	 (55), [55]	 (28), [28] (24), [45] (3), [25]
5	 (69), [226*]	 (66), [152*] (3), [3]	 (40), [71*] (20), [24*] (3), [3] (3), [9] (3), [17]	 (25), [25] (17), [17] (11), [11] (6), [6] (3), [3] (3), [3] (2), [2] (1), [9] (1), [1]
6	 (375), [429*]	 (375), [429*]	 (315), [315] (50), [50] (8), [8] (2), [2]	 (314), [314] (50), [50] (8), [8] (3), [429*]
7	 (135), [135]	 (135), [135]	 (95), [135] (40), [132]	 (92), [92] (40), [132] (3), [135]
8	 (146), [146]	 (146), [146]	 (70), [70] (52), [52] (22), [22] (1), [1] (1), [1]	 (70), [70] (52), [52] (15), [22] (7), [7] (1), [1] (1), [1]
11	 (56), [56]	 (56), [56]	 (56), [56]	 (51), [51] (5), [56]
12	 (226), [226]	 (226), [226]	 (226), [226]	 (145), [145] (81), [81]
13	 (182), [183*]	 (182), [183*]	 (182), [183*]	 (74), [74] (69), [69] (39), [39]
17	 (81), [81]	 (81), [81]	 (81), [81]	 (80), [80] (1), [1]
24	 (63), [66*]	 (63), [66*]	 (63), [66*]	 (62), [62] (1), [1]
25	 (88), [88]	 (88), [88]	 (88), [88]	 (50), [50] (38), [38]

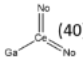
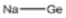
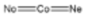
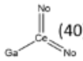
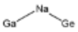
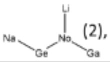
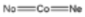
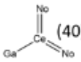
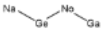
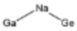
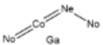
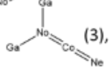
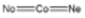
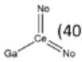
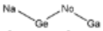
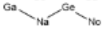
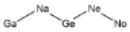
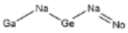
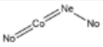
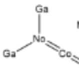
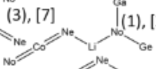
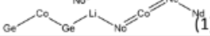
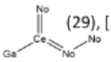
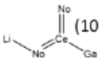
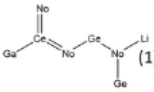
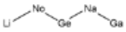
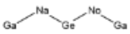
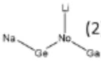
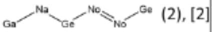
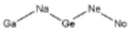
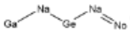
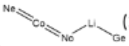
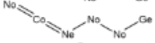
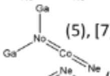
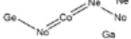
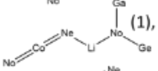
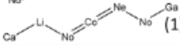
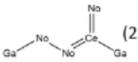
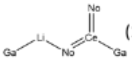
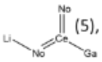
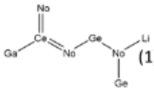
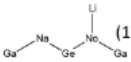
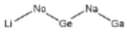
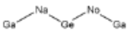
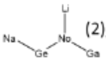
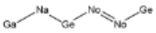
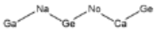
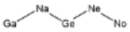
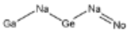
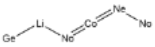
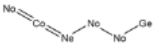
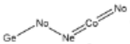
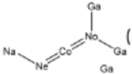
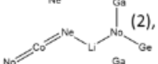
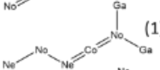
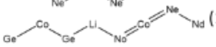
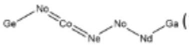
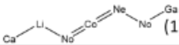
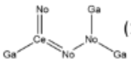
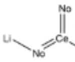
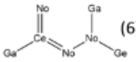
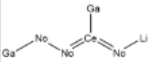
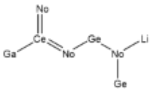
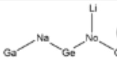
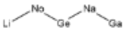
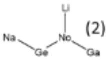
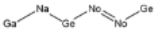
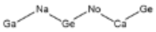
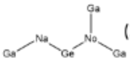
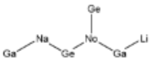
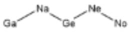
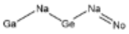
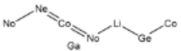
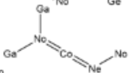
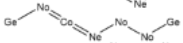
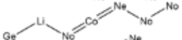
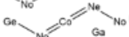
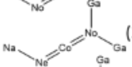
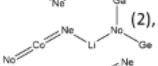
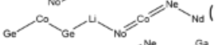
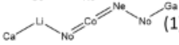
Results are shown for the Chk1 dataset in Table 2-17 and demonstrate a similar pattern to the previous two datasets: there is no difference in the results for the different similarity thresholds for the two lowest minimum node values, however, variation is seen for the larger minimum core sizes. When examining the larger datasets, the number of RG cores extracted varies for all values of the minimum core size.

Table 2-17: Table showing the number of cores extracted altering the two different settings for Chk1 from M2FP agglomerative 3 clusters

		Minimum Core Size					
		2	**3**	**4**	**5**	**6**	**7**
Similarity	**0.1**	3	4	6	12	16	18
	0.2	3	4	6	10	14	21
	0.3	3	4	8	12	16	17
	0.4	3	4	8	12	16	20
	0.5	3	4	9	15	21	23
	0.6	3	4	9	15	21	23
	0.7	3	4	9	15	23	27
	0.8	3	4	9	14	25	36
	0.9	3	4	9	15	24	41

Table 2-18 shows the RG cores generated for each cluster for Chk1 for several settings. Working down the table, when the minimum core size is 5 and similarity 0.5 the number of RG cores extracted increases for all of the clusters. The RG cores that are extracted using a larger minimum core size typically contain subgraphs of the RG cores extracted for the lower minimum core size.

Table 2-18: RG cores that are extracted for each for the Chk1 dataset for M2FP agglomerative 3 clusters. The value in parentheses demonstrates how many molecules this core relates to initially, the value in the square brackets demonstrates how many molecules relate to the core when the RG cores are mapped back onto the dataset through the re-examination step

Minimum Core Size, Similarity Settings	Cluster 1	Cluster 2	Cluster 3
2, 0.5	 (40), [40]	 (31), [33*]	 (35), [35]
3, 0.5	 (40), [40]	 (29), [29]  (2), [19]	 (35), [35]
4, 0.1	 (40), [40]	 (23), [23]  (8), [29]	 (30), [30]  (3), [7]  (2), [35]
4, 0.5	 (40), [40]	 (23), [23]  (6), [27]  (1), [1]  (1), [1]	 (30), [30]  (3), [7]  (1), [2]  (1), [1]
5, 0.5	 (29), [29]  (10), [21]  (1), [1]	 (21), [21]  (4), [21]  (2), [19]  (2), [2]  (1), [1]  (1), [1]	 (19), [19]  (6), [9]  (5), [7]  (3), [7]  (1), [2]  (1), [1]
6, 0.5	 (23), [28]  (11), [11]  (5), [21]  (1), [1]	 (17), [17]  (3), [21]  (3), [21]  (2), [19]  (2), [2]  (2), [2]  (1), [1]  (1), [1]	 (18), [18]  (6), [9]  (3), [10]  (2), [2]  (2), [2]  (1), [1]  (1), [1]  (1), [1]  (1), [1]
7, 0.5	 (19), [19]  (11), [21]  (6), [6]  (1), [10]  (1), [1]	 (17), [17]  (3), [21]  (2), [19]  (2), [2]  (2), [2]  (2), [2]  (1), [1]  (1), [1]  (1), [1]	 (14)  (4), [4]  (4), [4]  (4), [4]  (3), [7]  (2), [2]  (2), [2]  (1), [1]  (1), [1]

Most of the datasets produce RGs with average size eight to ten when using the default RG and a minimum core size of four, which is generally half the average size of the RGs, and seems a good size to use as there is an acceptable balance between the size of the RG cores and the number of cores extracted. The average of the means of the RG connected similarity values is around 0.2 for all datasets. The similarity threshold should be larger than the average similarity as the RG cores attempt to encompass relationships that are similar throughout several molecules within a dataset and not just most drug-like molecules in general. However, this is also a balance to be had since too large a similarity threshold would lead to RG cores that are too detailed whereas too small a value will lead to RG cores that are too small and too generic. The values of 0.5 appears to achieve a balance between specificity and generalisability.

2.4 Conclusions

This chapter describes the construction of a workflow to generate RG cores that show the relationships between molecules in a lead optimisation dataset. The workflow consists of five steps that have been explored within this chapter. The first step consists of cleaning the data and generating four different representations: M2FP, RG, RGFP and chemical graph. The second step is to create the corresponding similarity matrix depending on the type of representation. The third step is to cluster the datasets. From here, cores are extracted to establish a relationship of the clusters within the dataset easily. The final step of visualising the RG cores is examined in a later chapter.

It was concluded that the RGFP and chemical graph representations did not produce the most appropriate clusters and so will not be explored in further. Furthermore, it took considerably longer to compute the similarity matrix for the chemical graph. It is also important to note that if the reduced graph representation is used to create the similarity matrix, it should be created using the connected MCS, as the disconnected MCS produces too sparse similarity matrices leading to inappropriate clustering.

Although it was not possible to identify a single descriptor and clustering methods that was consistently the best, the M2FP representation and agglomerative clustering were chosen for the subsequent work as these gave good results in most cases.

RG cores were then extracted from the datasets to represent the molecules. After experimenting with the different settings that are available for generating the RG core, the use of a minimum of four nodes and a minimum similarity of 0.5 is suggested to produce the best results. The two settings could be changed depending on the similarity and size of the molecules and RGs within a dataset.

As an alternative to the clustering approach, it is possible to extract cores considering the full dataset. This would avoid selection of a descriptor and clustering method with no a priori information on which is most appropriate for a dataset. The two methods are compared in the next chapter. The subsequent chapters will look at how best to visualise these RG cores to obtain a better understanding and identify the relationships. Also, how a chemist using the tool can understand the exploration and exploitation that has occurred within the dataset.

3 Evaluating The Extracted Reduced Graphs Cores

3.1 Introduction

The previous chapter demonstrated how reduced graph (RG) cores could be derived to represent lead optimisation (LO) series. As described in the previous chapter, the RG core extraction process can be applied to a whole dataset or clusters derived from a pre-clustering step. This chapter aims to evaluate the RG cores from both methods to understand which methodology gives RG cores that best describe the relationships within the dataset.

Additionally, the RG cores will be compared with existing methods that aim to serve a similar function. These include a maximum common substructure (MCS) extraction method, RDKit's fMCS, which has been adapted for RGs and two other dataset representations methods, Markush structures and Bemis-Murcko scaffolds. Comparing the RG core extraction to existing methods will hopefully indicate how the RG core can be used to overcome some of their limitations. For example, the RG cores should represent LO series and bring together molecules containing similar but not necessarily identical scaffolds to overcome the issues associated with Markush structures and SAR tables.

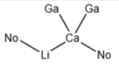
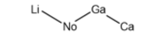
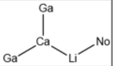
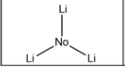
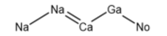
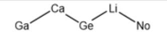
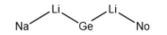
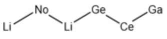
3.2 To Cluster or Not to Cluster

The RG core extraction process can be applied to both a clustered dataset or a whole dataset. There is added computation expense associated with the clustered dataset as the dataset has to be clustered prior to extracting the RG cores. There are also challenges related to clustering a dataset as clustering indexes that are typically used to determine the optimum number of clusters do not always give the most appropriate clusters for this purpose. Therefore, a comparison of the RG cores extracted with and without clustering was carried out to see whether the extra computational expense and time are needed. The comparison was based on RG core extraction parameter settings of the minimum core size four and the minimum similarity 0.5 in both cases. The clustering was carried out using M2FP fingerprints and agglomerative clustering with the clustering level chosen using the largest silhouette average.

3.2.1 Initial Comparison

The first thing to note is that there was very little difference in the time taken to compute the cores for both methods. Table 3-1 shows the RG cores extracted for both methods for the P2x7 Subset. The RG cores extracted for both methods are shown in the central portion of the table, additional RG cores unique to the whole data set are shown in the left section, and an RG core unique to the clustered data is shown in the right-hand section. For each section, the second column contains the number of molecules that were initially mapped to the RG core in the RG core extraction process; the next column contains the molecules that are mapped after remapping occurs; and the final column shows which cluster or clusters the molecules belong to for each RG core.

Table 3-1: Table displaying the cores extracted from the two different processes for the P2x7 subset

Additional Cores Extracted From The Whole Dataset				Cores Extracted From Both Methods				Additional Cores Extracted From the Clustered Dataset			
RG Core	Initial Number of Examples	Number of Examples Once Reexamined	Cluster Represented	RG Core	Initial Number of Examples	Number of Examples Once Reexamined	Cluster Represented	RG Core	Initial Number of Examples	Number of Examples Once Reexamined	Cluster Represented
	9	102	1		208	208	2		107	109	1, 2
	437	437	1, 2, 3, 4		1	50	2				
					249	249	3				
					110	110	4				
					16	16	3				

The results are very similar with and without clustering. Seven cores were found for the unclustered data and six for the clustered data, with five common to both. The remaining core found for the clustered data is closely related to one core found for the unclustered data which has an additional node. The same RG core can be extracted from multiple clusters when extracting RG cores from a clustered dataset. For example, the RG core [Ga][Ca]([Ga])[Li][No] is seen in cluster one and two. When this occurs, a unique set of RG cores is always reported, and the molecules that map to it are merged together regardless of which cluster they appear in.

A similar comparison was carried out for the Bajorath and P2x7 datasets. As larger numbers of cores were produced just the numbers are reported in this chapter, however, the Appendix contains the structures of each of the cores. Table 3-2 shows the number of RG cores extracted with and without clustering. Fewer RG cores were extracted from the whole datasets for the Bajorath and P2x7 datasets compared to when the data was clustered and the average core size is only slightly smaller. Therefore, extracting cores from the whole datasets provides a more condensed summary of the relationships within the datasets. The RG cores from the entire datasets also have fewer singletons, this is particularly noticeable for the P2x7 set, which results in thirty singleton RG cores when the data is pre-clustered. Additionally, over half the cores were the same whether they were extracted from the clustered dataset or the whole dataset.

Table 3-2: Table showing the number of RG cores extracted from the two different approaches, for clustering technique M2FP agglomerative

Dataset	Clusters					Whole dataset				Number of Common Cores
	Number of Clusters	Number of RG Cores	Average Core Size	Number of Singletons		Number of RG Cores	Average Core Size	Number of Singletons		
				Initial	Remap			Initial	Remap	
Bajorath	28	29	6	0	0	24	5	0	0	15
P2x7 Subset	4	6	5	1	0	7	5	1	0	5
P2x7	67	97	6	30	23	58	5	13	9	35

The ten RG cores reflecting the largest number of compounds in each dataset are shown in Table 3-3, from the whole dataset, and Table 3-4, clustered dataset. The number of molecules that map to the core is given in the brackets and the cells shaded in grey in Table 3-4 show RG cores that were also extracted from the whole dataset.

It can be seen in both tables that the majority of the top ten cores are linear RGs and, with the exception of the Cyto and Neurokinin datasets, mainly consist of four nodes. Some of the cores have a branch point and these tend to be focused around a ring node, particularly inert aromatic No. The most populated core in each dataset typically represents around a third of the dataset, although, the top RG core for each of the CDK2, FactorXa and p38α datasets has a lower representation of around fifteen percent.

The Cyto and Neurokinin datasets have multiple single and two node cores as well as disconnected RG cores. (As discussed in the previous chapter, the RG core extraction algorithm allows smaller RG cores to be found when cores with the minimum size are not possible.) Disconnected RG cores are generated as some of the RGs within these datasets were disconnected molecules, giving rise to disconnected RGs. The disconnected molecules are largely due to salts that were not removed in the initial cleaning process. Both datasets contain molecules that have very small RGs. The Cyto dataset has eight molecules that reduced a single node; 39 molecules were reduced to two nodes; and 86 molecules were reduced to three nodes. For the Neurokinin dataset, 17 molecules were reduced to a single node; 22 molecules were reduced to two nodes; and 74 molecules were reduced to three nodes. For example, the Neurokinin dataset contains the molecule carbon tetrachloride, a carbon atom attached to four chlorine atoms, which reduces to a single linker node, Li. The disconnected

RG core and small RG cores indicates that the dataset cleaning step should be improved before running this methodology.

When comparing the clustered and non-clustered results, out of the top ten results for all datasets 71 of the 90 RG cores are the same, with two datasets Neurokinin and p38α having all the same top ten RG cores. The large overlap between the top RG cores extracted for both methods demonstrates that any variation in the RG cores overall generally occurs for RG cores that are represented by fewer molecules.

Table 3-3: Top ten cores from each dataset with the number of molecules associated with the core using the whole dataset extraction method

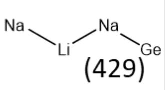
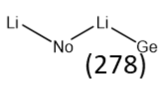
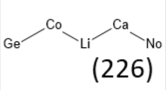
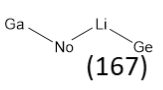
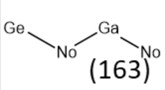
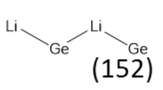
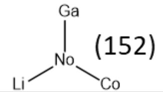
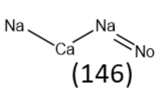
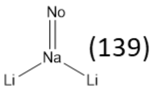
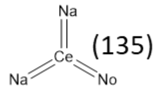
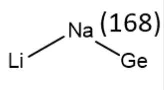
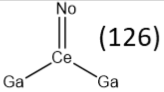
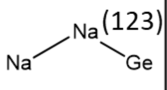
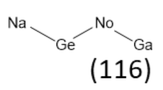
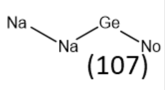
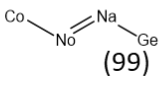
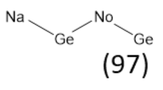
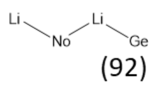
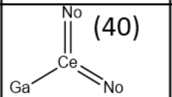
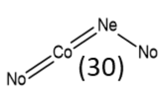
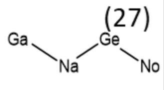
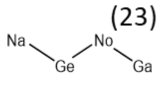
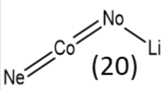
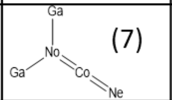
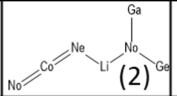
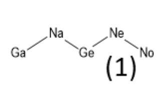
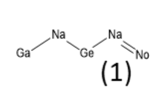
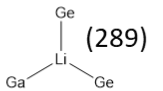
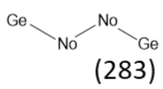
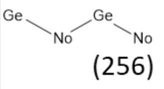
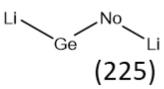
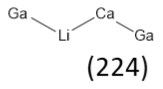
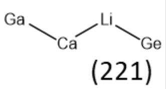
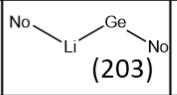
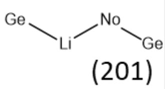
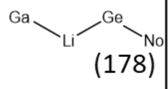
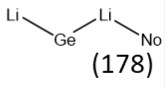
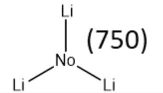
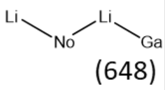
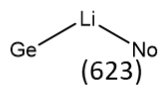
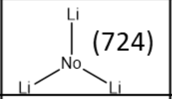
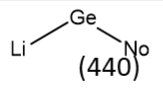
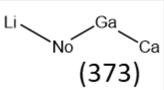
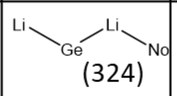
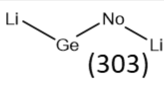
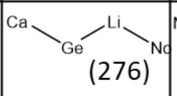
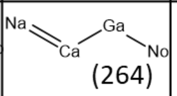
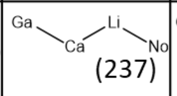
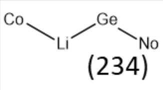
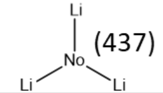
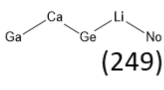
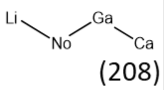
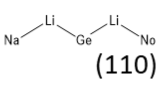
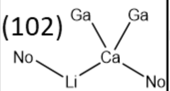
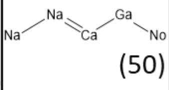
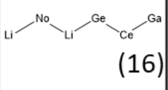
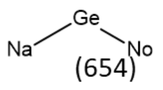
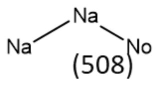
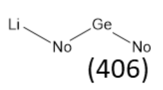
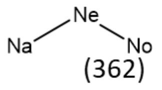
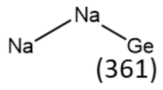
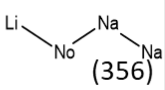
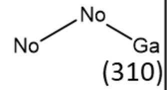
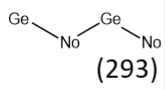
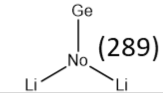
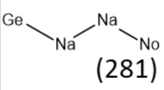
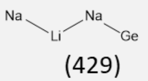
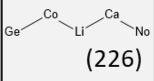
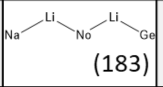
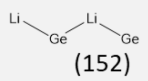
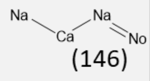
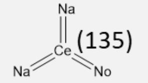
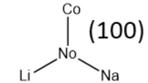
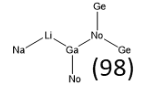
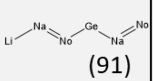
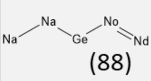
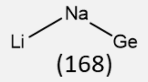
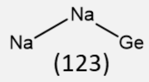
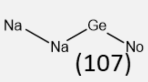
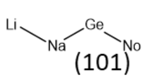
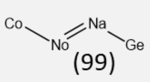
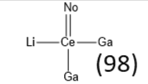
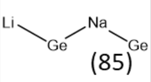
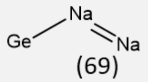
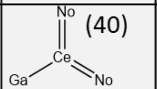
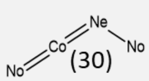
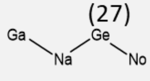
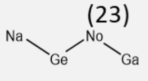
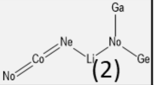
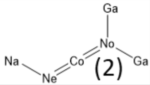
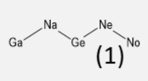
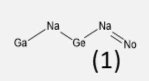
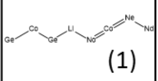
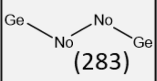
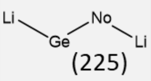
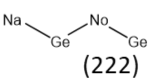
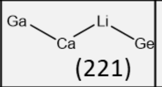
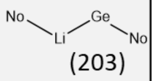
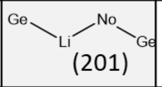
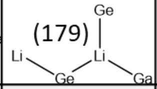
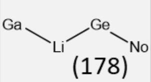
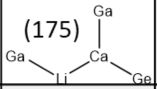
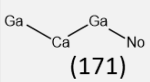
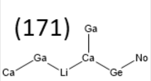
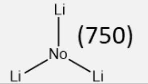
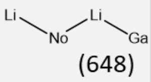
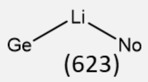
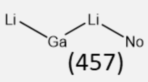
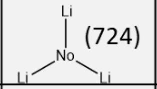
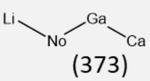
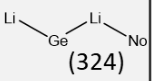
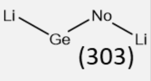
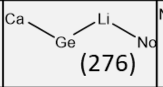
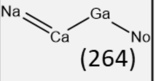
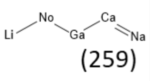
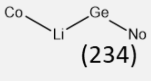
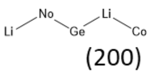
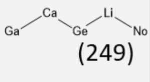
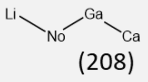
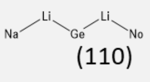
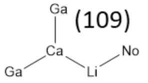
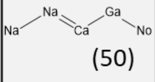
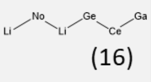
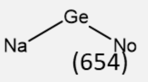
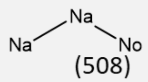
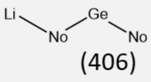
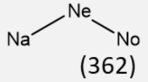
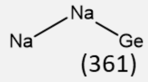
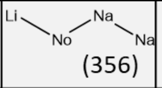
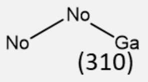
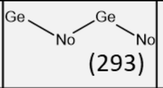
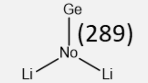
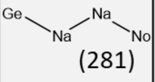
Dataset	Top Ten RG Cores Extracted									
	1	2	3	4	5	6	7	8	9	10
Bajorath	 (429)	 (278)	 (226)	 (167)	 (163)	 (152)	 (152)	 (146)	 (139)	 (135)
CDK2	 (168)	Ge—Ne (131)	 (126)	 (123)	 (116)	 (107)	 (99)	 (97)	 (92)	No═Ne (75)
Chk1	 (40)	 (30)	 (27)	 (23)	 (20)	 (7)	 (2)	 (1)	 (1)	
Cyto	Ga (4980)	Ge (3628)	Li—No (3276)	Ga—No (2835)	Li No No (2752)	Li Li (2332)	Li Na (2176)	Li—Ge (1912)	Ca—Ga (1721)	Li—Ga (1615)
FactorXa	 (289)	 (283)	 (256)	 (225)	 (224)	 (221)	 (203)	 (201)	 (178)	 (178)
Neurokinin	Li (2250)	Li No (2074)	Ga (2014)	Li—No (1913)	Ge (1538)	Li—Ga (1302)	Li—Ge (1078)	 (750)	 (648)	 (623)
P2x7	 (724)	Ge—No (602)	 (440)	 (373)	 (324)	 (303)	 (276)	 (264)	 (237)	 (234)
P2x7 Subset	 (437)	 (249)	 (208)	 (110)	(102) 	 (50)	 (16)			
p38α	 (654)	 (508)	 (406)	 (362)	 (361)	 (356)	 (310)	 (293)	 (289)	 (281)

Table 3-4: Top ten cores from each dataset with the number of molecules associated with the core using the clustered dataset extraction method. The shaded RG cores are cores that are also seen in the whole dataset extraction method results

Dataset	Top Ten RG Cores Extracted From Clustered Dataset										
	1	2	3	4	5	6	7	8	9	10	=10
Bajorath	 (429)	 (226)	 (183)	 (152)	 (146)	 (135)	 (100)	 (98)	 (91)	 (88)	
CDK2	 (168)	Ge—Ne (131)	 (123)	 (107)	 (101)	 (99)	 (98)	 (85)	No=Ne (75)	 (69)	
Chk1	 (40)	 (30)	 (27)	 (23)	 (2)	 (2)	 (1)	 (1)	 (1)		
Cyto	Ga (4980)	Ge (3628)	Li—No (3276)	Ga—No (2835)	Li No No (2752)	Li Li (2332)	Li Na (2176)	Li—Ge (1912)	Ca—Ga (1721)	Li—Ga (1615)	
FactorXa	 (283)	 (225)	 (222)	 (221)	 (203)	 (201)	 (179)	 (178)	 (175)	 (171)	 (171)
Neurokinin	Li (2250)	Li No (2074)	Ga (2014)	Li—No (1913)	Ge (1538)	Li—Ge (1078)	 (750)	 (648)	 (623)	 (457)	
P2x7	 (724)	Ge—No (602)	 (373)	 (324)	 (303)	 (276)	 (264)	 (259)	 (234)	 (200)	
P2x7 Subset	 (249)	 (208)	 (110)	 (109)	 (50)	 (16)					
p38α	 (654)	 (508)	 (406)	 (362)	 (361)	 (356)	 (310)	 (293)	 (289)	 (281)	

A summary of all the RG cores generated from the clustered and non-clustered datasets is shown in Table 3-5 which gives: the number of molecules; the number of unique RGs; the number of RG cores extracted from both methods; the number of RG cores that appear in both methods; and the average percentage overlap of the methods. For the small datasets, Chk1 and P2x7 Subset, there is a considerable overlap of 77%, with only two RG cores different in both cases. However, the larger datasets with more RG cores extracted have fewer RG cores that appear within both methods. As more molecules cover a larger area of chemical space, a greater variation in the RG cores is observed. This is because the clustered datasets lead to RG cores being extracted from lots of small areas of chemical space, whereas, when extracting cores from the whole dataset there is a tendency to identify relationships between a larger number of molecules across a more diverse chemical space.

Table 3-5: Comparison of the number of cores extracted from both methods, as well as the overlap between the two methods for each dataset

Dataset	Number of Molecules	Number of RGs	Number of RG Cores in Clustered Dataset	Number of RG Cores in Whole Dataset	Number of RG Cores that are Common	Average Overlap Percentage (%)
Bajorath	2549	920	29	24	15	57.11
CDK2	1368	824	195	114	68	47.73
Chk1	106	91	9	9	7	77.78
Cyto	6370	3762	181	182	132	72.73
FactorXa	1956	883	126	42	22	34.92
Neurokinin	2475	1451	85	86	56	65.50
P2x7	2259	822	97	58	35	48.21
P2x7 Subset	691	162	6	7	5	77.38
p38α	3644	1902	245	125	82	49.53

Table 3-6 shows the details of the RG cores extracted using the whole dataset extraction methodology. One of the aims of the RG core is to represent the relationships between the molecules within a dataset whilst also reducing the amount of information to process. The amount of data has been reduced vastly for all of the datasets, with some more so than others. Additionally, for most of the datasets, the average RG core size is five nodes, indicating that the relationship identified by the RG core is significant, i.e., it represents a substantial portion of the molecules.

When examining the number of molecules each of the RG cores represents, there are a few key things to note. The first is that several of the datasets have RG cores representing just one molecule. These are present in all the datasets except two. The second is that when looking at the average number of molecules that each RG core represents, the numbers vary from just over one percent, CDK2, to just over 24%, P2x7 Subset. For the P2x7 Subset and Chk1 datasets each core represents a large percentage of the molecules, 24.2% and 15.8% respectively. These were small datasets with a less diverse chemical space, shown by the relatively large average pairwise similarity across the dataset in Chapter 2. The two datasets that have the largest average number of molecules represented are Cyto and Neurokinin, however, this is due to the size and nature of the RG cores that have been extracted from these two datasets with both of the datasets extracting several RG cores that consist of just one or two nodes. Cyto has two single node cores and fifteen cores with two nodes, and Neurokinin has four single node cores and nine cores with two nodes. These small cores consist of simple common nodes such as a linker node. Therefore, these RG cores are not selective and do not demonstrate meaningful relationships between the molecules within the dataset, as the RG cores could be generic to any molecule. As mentioned previously, these small cores were due to some of the RGs within the dataset being themselves small, i.e., consisting of one or two node RGs.

Table 3-6: Table showing the number of RG cores extracted for each dataset using whole dataset extraction method

Dataset	Number of Molecules	Number of RGs	Number of RG Cores	Average Number of Nodes in RG Core	Number of Singletons	Number of Molecules on Average a Core Represents [%]	Average % The Core Represents of The RG (%)
Bajorath	2549	920	24	5.33	0	115.58 [4.5%]	53.72
CDK2	1368	824	114	4.81	26	23.11 [1.7%]	49.70
Chk1	106	91	9	4.67	2	16.78 [15.8%]	50.86
Cyto	6370	3762	182	3.68	22	282.26 [4.4%]	32.55
FactorXa	1956	883	42	4.83	7	87.52 [4.8%]	42.70

Neurokinin	2475	1451	86	3.60	10	226.36 [9.1%]	25.61
P2x7	2259	822	58	5.26	10	90.50 [4.0%]	45.96
P2x7 Subset	691	162	7	5.00	0	167.43 [24.2%]	56.73
P38α	3644	1902	125	4.66	23	75.14 [2.1%]	46.85

Table 3-7 shows similar results to Table 3-6 but for the clustered datasets. One notable difference is the larger number of RG cores and singletons extracted. The average number of nodes in the RG core is slightly larger, with a broader coverage of the RG of the molecules. However, they do not represent as many molecules per core. The RG cores extracted from the whole dataset were slightly smaller in size but represented more molecules. Therefore, there is a trade-off between the size of the RG cores and the number of molecules they represent.

Table 3-7: Table showing the number of RG cores extracted for each dataset using clustered dataset extraction method from M2FP agglomerative results

Dataset	Number of Molecules	Number of RGs	Number of RG Cores	Average Number of Nodes in RG Core	Number of Singletons	Number of Molecules on Average a Core Represents [Percent]	Average % The Core Represents of The RG (%)
Bajorath	2549	920	29	6.31	0	92.93 [3.6%]	61.33
CDK2	1368	824	195	6.25	81	12.11 [0.9%]	55.60
Chk1	106	91	9	5.22	3	14.11 [13.3%]	51.66
Cyto	6370	3762	181	3.72	16	290.38 [4.5%]	32.39
FactorXa	1956	883	126	7.07	28	37.94 [1.9%]	53.68
Neurokinin	2475	1451	85	3.71	5	215.86 [8.7%]	26.26
P2x7	2259	822	97	6.18	23	61.04 [2.7%]	50.94
P2x7 Subset	691	162	6	5.00	0	123.67 [17.9%]	60.43
P38α	3644	1902	245	5.83	73	45.66 [1.3%]	49.48

To summarise this section, the RG cores extracted from the clustered datasets tend to be larger and more selective as each core, on average, matches to fewer molecules than the cores extracted from the whole dataset. Furthermore, the clustered datasets tend to include more RG cores that represent just one molecule. The overlap between the RG cores extracted from the two methods is around 50%. The RG cores extracted from both methods were analysed further below to identify if one set of RG cores is more appropriate than the other.

3.2.2 Scaffold Score

As noted above, the RG cores extracted from the clustered datasets are generally larger and represent fewer molecules, whereas the RG cores extracted from the whole datasets were mainly smaller and represent more molecules. A scaffold score was explored to understand further which RG cores would be more beneficial to see if one methodology was favoured over the other. Bandyopadhyay et al. presented a scaffold score for chemical graphs to quantify the quality of a scaffold (Bandyopadhyay et al., 2019). The S score is as follows:

$$S = -\log_{10}\left(\sqrt{N_{core}.\frac{N_m}{N}.\frac{1}{\sqrt{\sigma}}.\frac{1}{R}}\right) \tag{3.1}$$

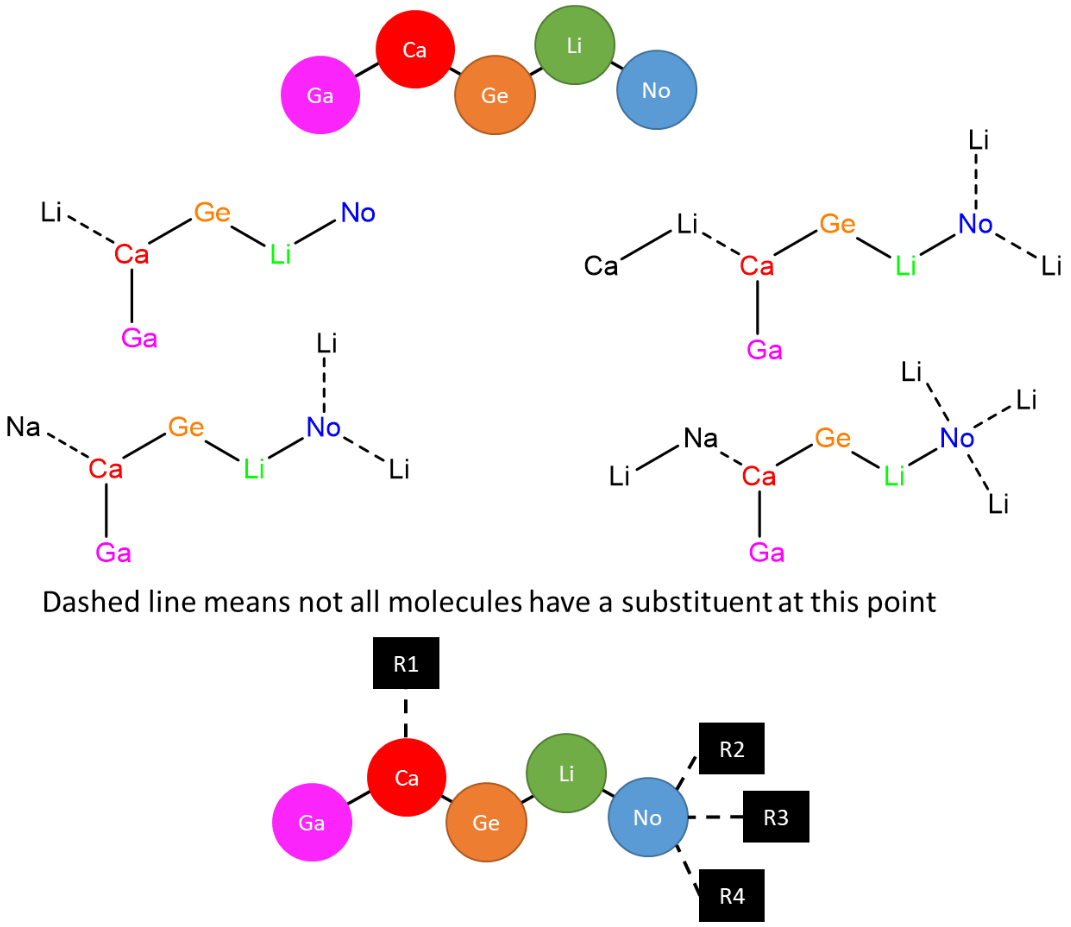
Where N_{core} is the number of atoms within the scaffold, N_m is the number of molecules associated with the scaffold, N is the number of molecules within the dataset, R is the number of R-groups identified for the scaffold and σ is a measure of how close the molecules are to the scaffold and is defined as

$$\sigma = \sum_{i=1}^{N_m}(A_i - N_{core})^2 \tag{3.2}$$

Where A_i is the number of atoms within molecule *i*. σ is, therefore, a measure of the proportion of each molecule that is represented by the scaffold. A high value of σ indicates relatively large molecules and a relatively small scaffold, whereas a small value of σ indicates that the scaffold represents a large proportion of the molecules.

The S score has been modified for the RG core analysis as follows: the N_{core} becomes the number of RG nodes within the RG core and A_i is the number of nodes within the RG representing molecule, *i*.

In order to extract the number of R groups the data was reorganised. The RG for each molecule was mapped to the RG core. The additional nodes in each RG were then attached to the respective node in the RG core as R substituents, Figure 3-1. Where multiple molecules had substituents attached to the same node on the RG core, these were aggregated as alternative nodes at that substituent position. Figure 3-1 demonstrates an example for an RG core which has four molecules mapped to it. The core consists of five nodes, [Ga][Ca][Ge][Li][No]. Each of the RGs of the molecules can be mapped onto the core and the additional RG nodes were considered as substituents. These substituents are shown with dashed bonds. Therefore, there were four different substitution sites on the RG core and seven unique R groups for this example. There are a few instances when an RG core represents the whole RGs, which leads to no R groups for that molecule. If either sigma or R are zero, then $\frac{1}{\sqrt{\sigma}}$ and $\frac{1}{R}$ are set to 1.1 to avoid division by zero.



Dashed line means not all molecules have a substituent at this point

R1	Li, Li-Ca, Na, Na-Li
R2	Li
R3	Li
R4	Li

Figure 3-1: An example of how five molecules map onto RG core [Ga][Ca][Ge][Li][No] to demonstrate R groups

A larger RG core score indicates a bigger scaffold with fewer R-groups where the majority of the molecules are of similar size to the RG core and where the RG core covers a large portion of the dataset. The median RG core score is reported for each dataset in Table 3-8 for RG cores calculated from the whole dataset and from the pre-clustered dataset. The median score was quoted as it is less affected by outliers than the mean, as the distributions are not normally distributed.

Table 3-8: Median RG core scaffold score for each dataset

Dataset	Number of Molecules	Number of RGs	Whole dataset			Clustered Dataset		
			Number of RG Cores	Number of Singletons	Median RG Core Score	Number of RG Cores	Number of Singletons	Median RG Core Score
Bajorath	2549	920	24	0	1.883	29	0	1.820
CDK2	1368	824	114	35	1.681	195	89	1.226
Chk1	106	91	9	3	1.126	9	3	1.126
Cyto	6370	3762	182	41	2.245	181	16	2.265
FactorXa	1956	883	42	9	1.991	126	31	1.589
Neurokinin	2475	1451	86	21	2.070	86	8	2.137
P2x7	2259	822	58	13	1.813	97	16	1.662
P2x7 Subset	691	162	7	1	1.394	6	1	1.435
P38α	3644	1902	125	30	1.975	245	88	1.781

For five of the nine datasets, the RG cores extracted using the whole dataset methodology have a higher median RG core score indicating that they are more representative of the data. The three datasets for which the median RG core scores are higher for the clustered datasets are Cyto, Neurokinin and P2x7 Subset. This could be because for these datasets both methods produce very similar numbers of RG cores, which probably indicates that when the number of RG cores is similar between methods, the clustered dataset produces slightly better RG cores. However, the difference in scores is minimal, so the extra computational expense of clustering the datasets does not generate significantly better results for these three datasets.

Identical median RG core scores were obtained for both methods for the Chk1 dataset. The RG cores extracted from the Chk1 dataset only vary by one RG core as shown in Figure 3-2. The RG core that has been extracted from the whole dataset is more representative as it is derived from 20 molecules, whereas the RG core for the clustered dataset represents a single molecule only. The RG core derived from the whole dataset is a subgraph of that derived from the clustered dataset and has a higher RG core score due to the larger number of molecules it represents. Therefore, in this case, the whole dataset methodology is more representative and produces better RG cores overall.

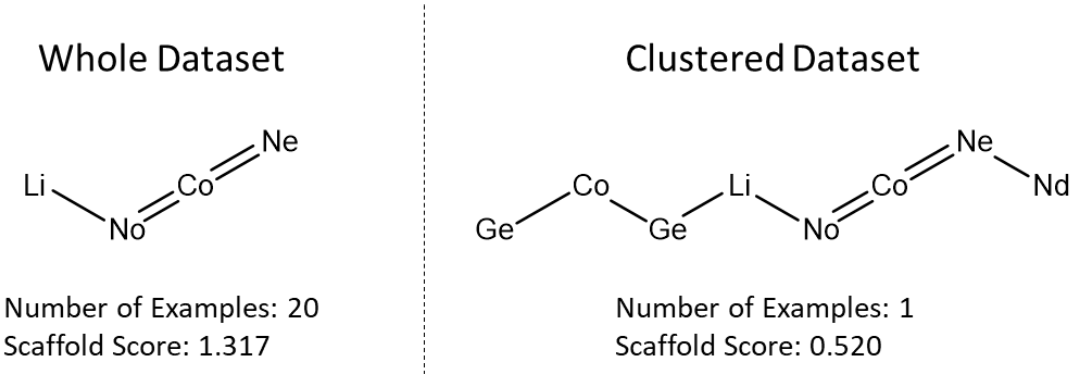


Figure 3-2: RG cores that vary within the Chk1 dataset

3.2.3 Summary

To summarise the results from the clustering vs using the whole dataset analyses, it was concluded that extracting the cores from the whole dataset was more appropriate as similar results were generated in a shorter amount of time and the RG cores were produced that generate higher RG core scores, overall. From here on, the RG cores were extracted from the whole dataset.

3.3 Applying RG Extraction to Known Datasets

In this section, the RG core extraction is applied to datasets of increasing difficulty to validate its performance.

The MMP12 dataset is a publicly available LO dataset developed at GSK (Pickett, Green, Hunt, Pardoe, & Hughes, 2011). The MMP12 dataset is constructed around one Markush structure shown in Figure 3-3.

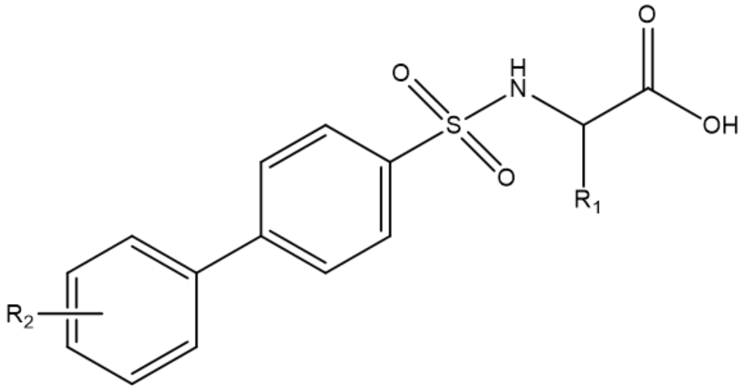


Figure 3-3: Markush structure that the MMP12 dataset has been built around

There are 2500 molecules and 484 unique RGs within the MMP12 dataset. The RG core extraction process resulted in a single RG core that corresponds to the Markush structure and represents all of the molecules in the dataset. The RG core is shown in Figure 3-4 along with the substructure of the Markush where the atoms are coloured according to the corresponding nodes in the RG core. The MMP12 RG core has a scaffold score of 1.623. This dataset represents an "easy" case in which the RG core extraction method successfully reproduced the Markush.

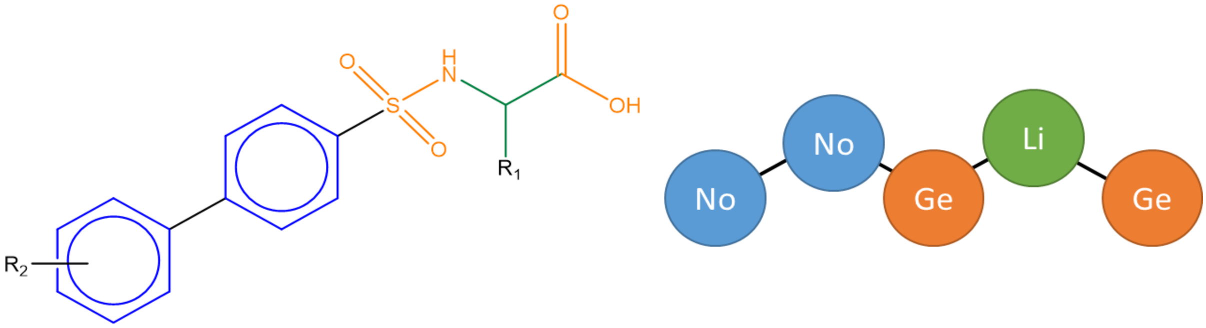


Figure 3-4: RG core extracted from the MMP12 dataset

The Bajorath dataset is a constructed dataset that was put together to investigate the progression of LO series (Vogt, Yonchev, & Bajorath, 2018). It is more complex than the MMP12 dataset and is based on 34 analog series-based (ASB) scaffolds. Scaffolds are typically sourced from individual molecules, however, ASB scaffolds are generated from analog series (AS) which are extracted from the dataset using retrosynthetic combinatorial analysis procedure (RECAP) matched molecular pairs (MMPs), Figure 3-5 (Dimova, Stumpfe, & Bajorath, 2018; Dimova, Stumpfe, Hu, & Bajorath, 2016; Stumpfe, Dimova, & Bajorath, 2016).

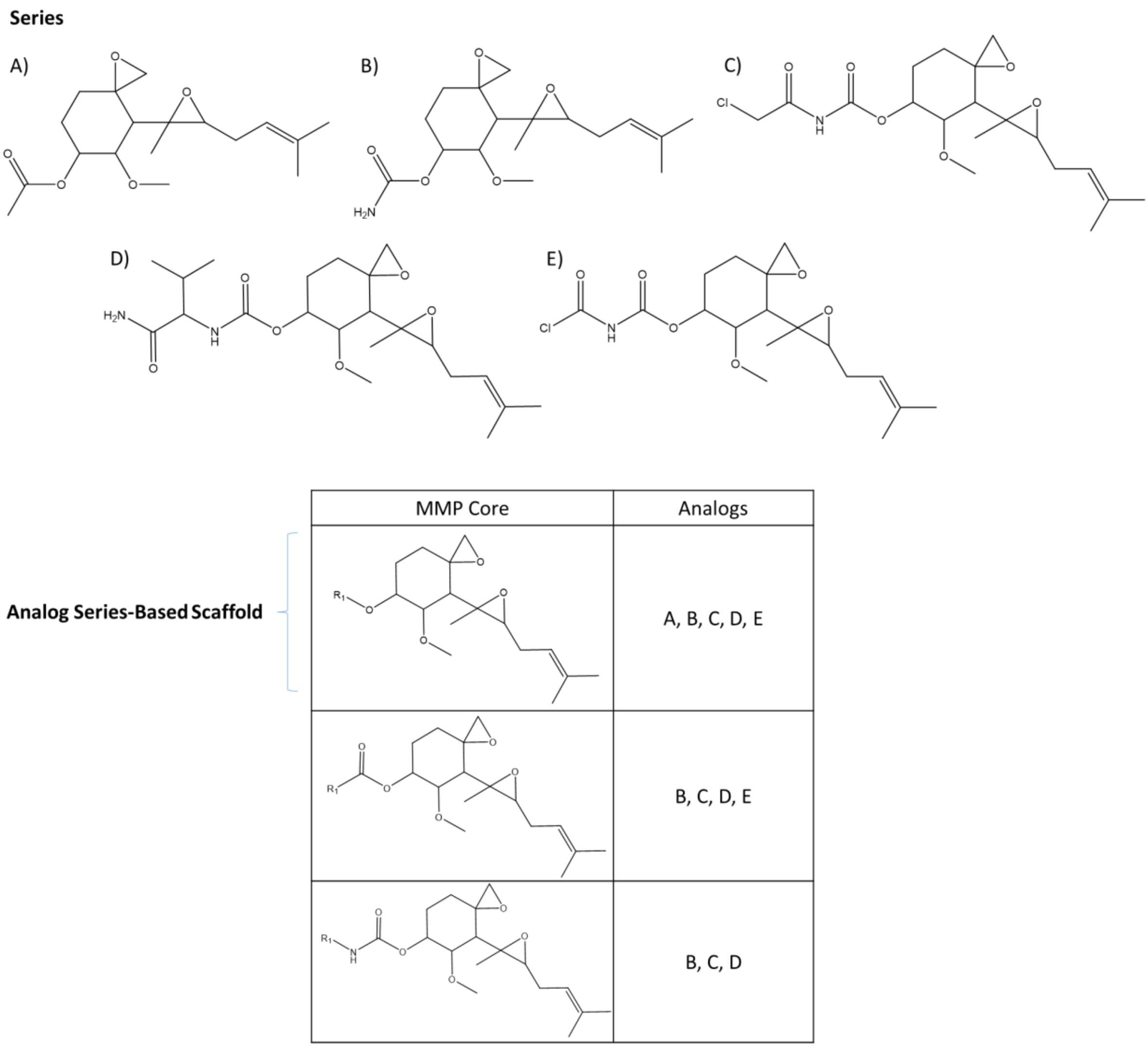
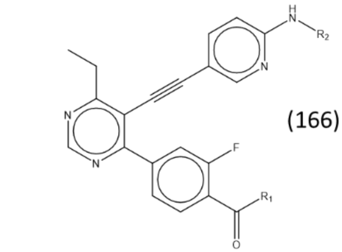
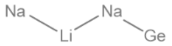
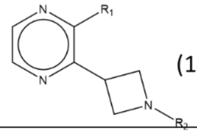
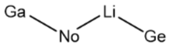
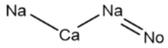
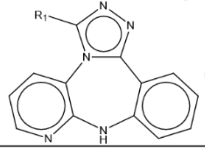
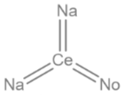
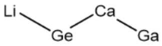
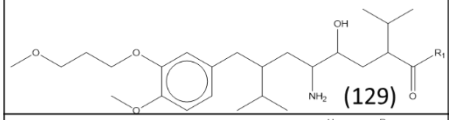
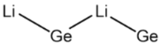
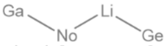
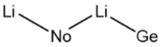
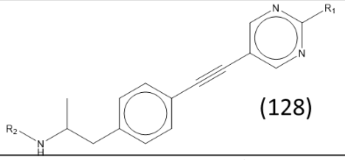
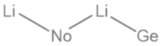
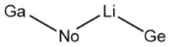
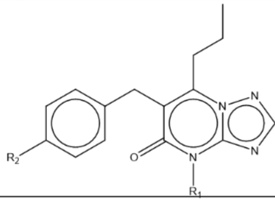
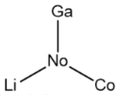
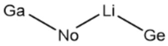
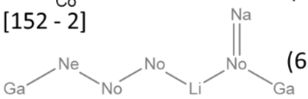
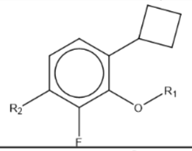
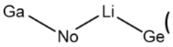
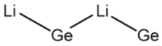
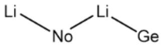
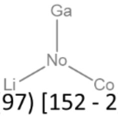
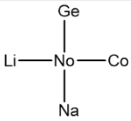
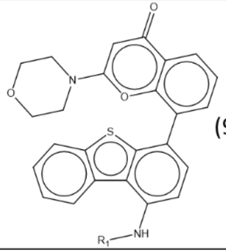
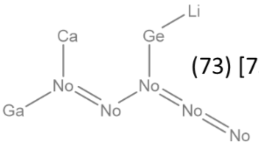
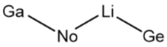
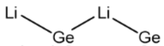
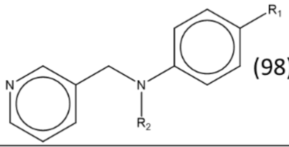
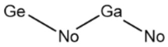
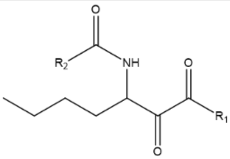
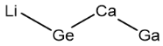
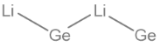


Figure 3-5: Identification of an analog series-based scaffold *(Dimova et al., 2016)*

The top ten most populated ASB scaffolds are shown in Table 3-9 with the number of molecules that each ASB scaffold represents in parentheses. A total of 24 RG cores were extracted for the dataset. The second column shows the corresponding RG cores which represent the same set of molecules as the ASB scaffold. The numbers in the parentheses, “()“, indicate how many molecules that have that ASB scaffold contain the associated RG core. The numbers within the square brackets, “[]” which are separated by a dash, indicate how many molecules within the dataset contain that RG core and how many different ASB scaffolds these molecules cover, respectively. For example, the first ASB scaffold represents 166 molecules. There is one RG core that represents all 166 molecules covered by the ASB. A total of 429 molecules contain this RG core and these are covered by a total of 5 ASBs. In most cases, one of the RG cores corresponds directly to the ASB scaffold (as for the first row) and is shown in grey. However, the RG cores shown in black reflect that some RG similarities may be observed outside the ASB scaffolds and instead are detected within the R-group substituents throughout the dataset. Some of the RG cores map to more than one ASB.

Figure 3-6 and Figure 3-7 show examples where the RG cores have merged multiple ASBs together. In Figure 3-6 three ASB scaffolds were combined to create a five node RG core, [No][Ca][Li][Co][Ge], each node has been highlighted in its respective colour. The second ring within the fused ring has not been incorporated into the RG core as in molecule (a) it is an inert aromatic node, No, whereas molecules (b) and (c) have hydrogen bond acceptor aliphatic rings at this position, Ca. Figure 3-7 shows four ASB scaffolds that were combined into a four-node RG core, [Na][Li][Na][Ge], each node has also been highlighted in its respective colour. The RG core comprises two ring nodes rather than three due to molecule (c) containing only two rings. The ethyl group that is a substituent on the pyrimidine ring and is common to all four ASBs is not part of the RG core due to using the default settings when generating the RGs. Therefore, the RG core extraction method has produced fewer scaffolds than the ASB with, in general, each core representing more molecules than the ASBs. However, in some cases the RG cores represent fewer atoms overall due to the way the RG nodes are defined.

Table 3-9: Top ten most populated ASB Scaffolds and RG core present for each for Bajorath dataset

ASB Scaffold	RG Core
 (166)	 (166) [429 - 5]
 (147)	 (1) [167 - 10]  (146) [146 - 1]
 (136)	 (135) [135 - 1]  (1) [32 - 4]
 (129)	 (61) [152 - 4]  (62) [167 - 10]  (6) [278 - 8]
 (128)	 (126) [278 - 8]  (2) [167 - 10]
 (120)	 (55) [152 - 2]  (1) [167 - 10]  (65) [65 - 1]
 (104)	 (1) [167 - 10]  (1) [152 - 4]  (2) [278 - 8]  (97) [152 - 2]  (3) [3 - 1]
 (99)	 (73) [73 - 1]  (2) [167 - 10]  (24) [152 - 4]
 (98)	 (98) [163 – 3]
 (94)	 (28) [32 - 4]  (66) [152 – 4]

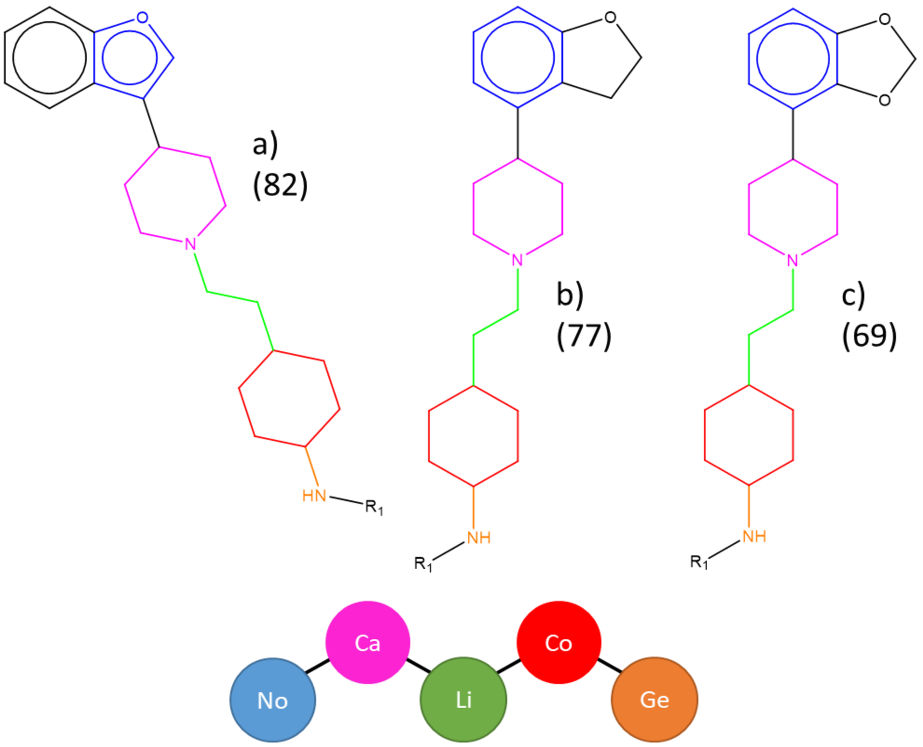


Figure 3-6: ASB Scaffolds that are combined into one RG core [No][Ca][Li][Co][Ge]

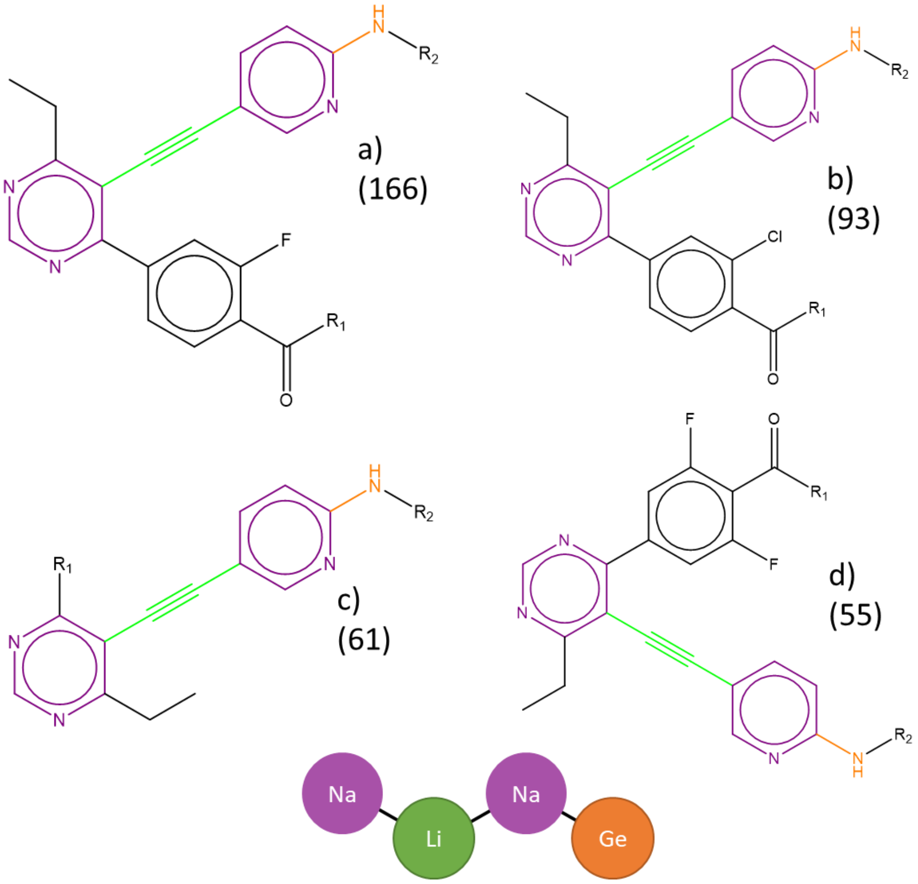


Figure 3-7: ASB Scaffolds that are combined into one RG core [Na][Li][Na][Ge]

Three of the datasets introduced in Chapter 2, Neurokinin, P2x7 and P2x7 Subset, have been extracted from ChEMBL, with each molecule annotated by the journal in which the molecule was published. These datasets were examined to see if the RG cores were able to merge molecules together than are derived from different journal papers and which might be based on different chemical scaffolds.

For each dataset, the number of unique DOC CHEMBLIDs present is shown Table 3-10 alongside the number of unique RG cores extracted for the dataset and the average number of RG cores per DOC CHEMBLID showing that for most journal articles the molecules are represented by multiple RG cores. For both the Neurokinin and P2x7 datasets there are more DOC CHEMBLIDs than RG cores. When examining which RG cores correspond to molecules within each DOC CHEMBLID, for the P2x7 dataset, only nine journal articles have one RG core. As, in general, there is more than one RG core per paper, this indicates that the molecules within a journal article are not always closely aligned or based on one Markush structure. For the Neurokinin and P2x7 Subset datasets, the journal articles all contain multiple RG cores per paper.

Table 3-10: Table showing the number of DOC CHEMBLIDs for each dataset

Dataset	Number of Molecules	Number of RGs	Number of RG Cores	Number of DOC CHEMBLIDs	Average Number of RG Cores per DOC CHEMBLID
Neurokinin	2475	1451	86	130	11.57
P2x7	2259	822	58	61	4.89
P2x7 Subset	691	162	7	4	3.25

There were four DOC CHEMBLID's within the P2x7 Subset dataset that represent three journal articles, CHEMBL1157114 (supplementary information of (Chambers et al., 2010)) and CHEMBL2218064 (Chambers et al., 2010), CHEMBL1221272 (Abdi et al., 2010) and CHEMBL1268987 (Abberley et al., 2010). The Markush structures representing the molecules within each journal article are shown in Figure 3-8. Figure 3-9 shows the Markush structures associated with each of the journal articles alongside RG representations of each Markush. *R* indicates where an R group variation occurs; *X* indicates an amide linker; the grey RG node, hetero, indicates a hydrogen bond acceptor or hydrogen bond acceptor and donor; Aro, the black node, is an aromatic ring; and Ali, the yellow RG node (mapping to Ce, Ca, Na and Ne nodes), is an aliphatic ring variation.

Seven RG cores were extracted from the whole of the P2x7 Subset. When comparing these RG cores to the Markush structures there are four RG cores that represent all or part of the Markush structures, [Li][No]([Li])[Li], [Ga][Ca][Ge][Li][No], [Na][Li][Ge][Li][No], [Li][No][Li][Ge][Ce][Ga]. These are the shaded RG cores within Table 3-11. Also, there are five out of the nine Markush structures that are incorporated within the RG cores, a(iii), a(iv), b(i), c(i) and c(ii). Therefore, most of the Markush structures are represented and RG cores represent multiple Markush structures and across different journal articles.

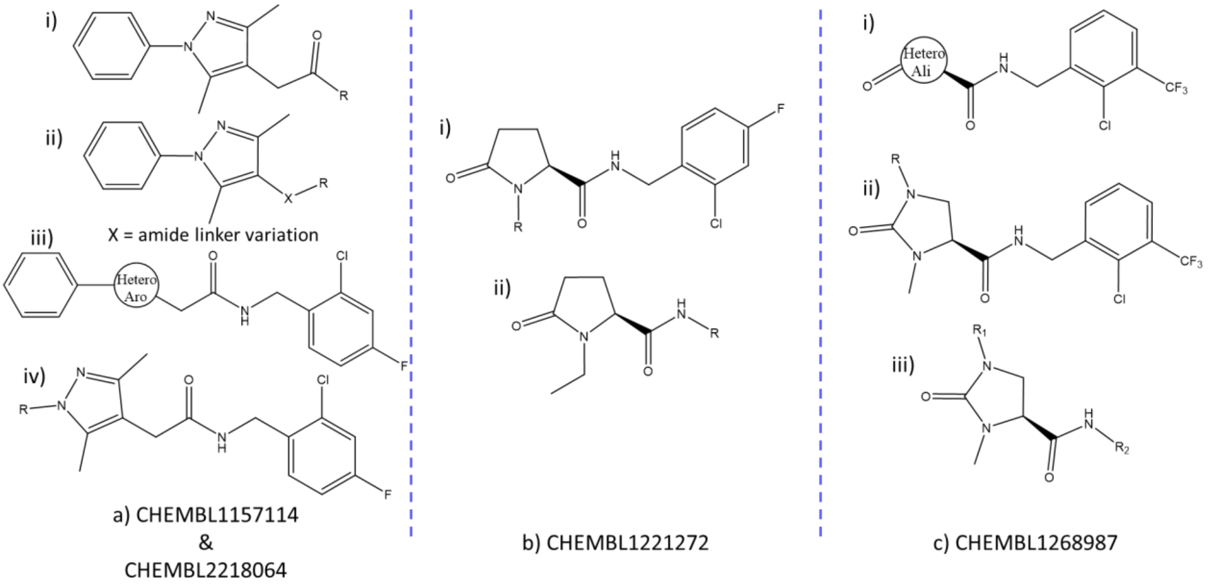
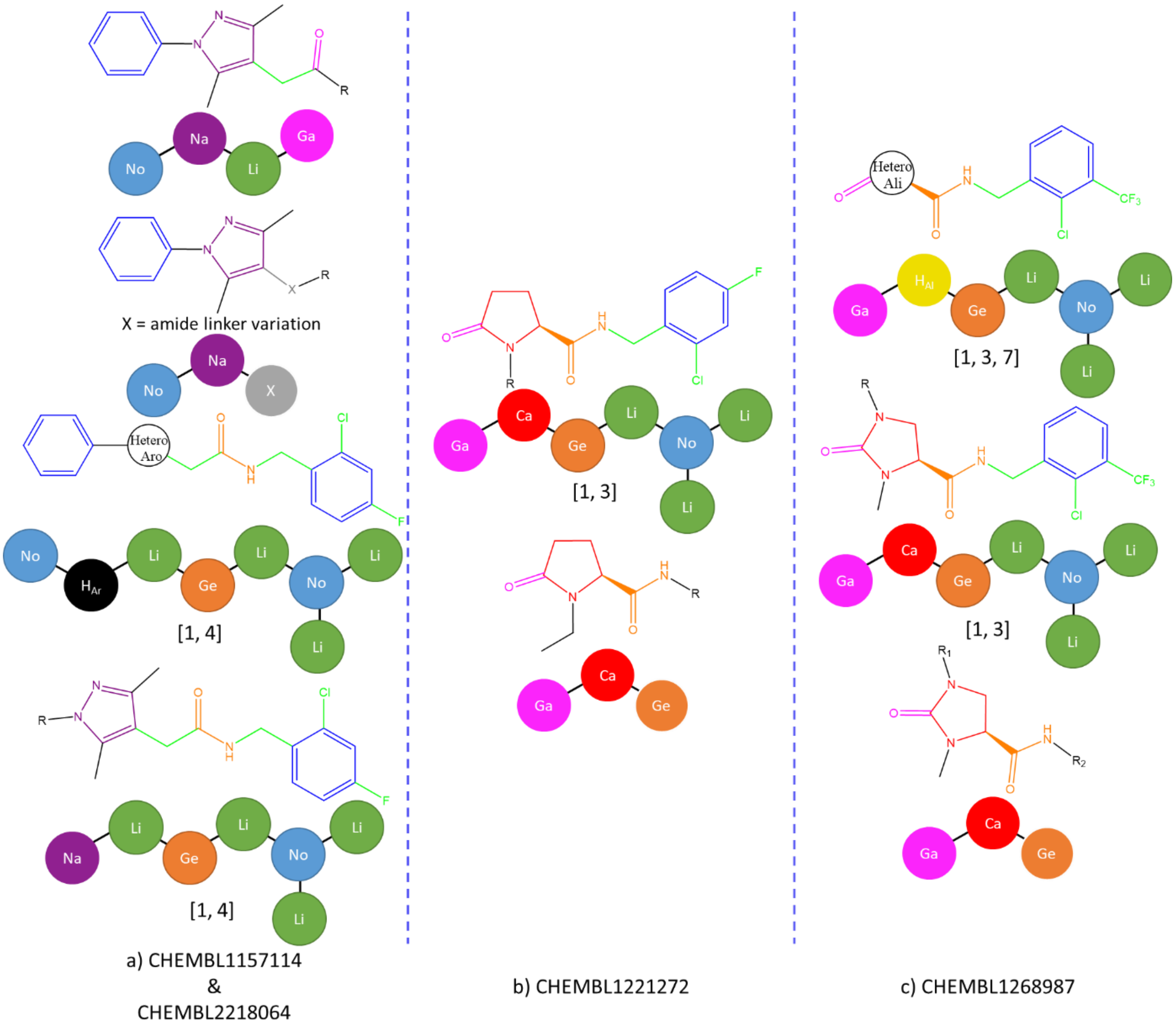


Figure 3-8: Markush structures that are present in each of the DOC CHEMBLID a) CHEMBL1157114 and CHEMBL2218064 *(Chambers et al., 2010)* b) CHEMBL1221272 *(Abdi et al., 2010)* c) CHEMBL1268987 *(Abberley et al., 2010)*



a) CHEMBL1157114 & CHEMBL2218064

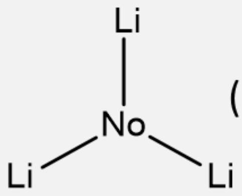
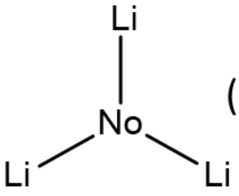
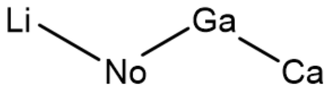
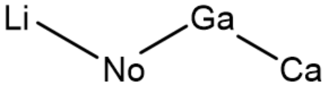
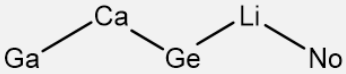
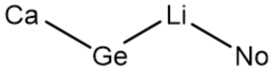
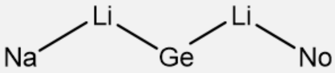
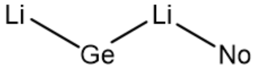
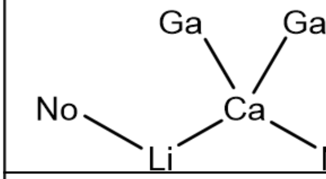
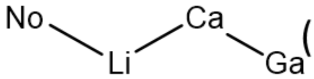
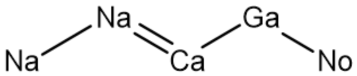
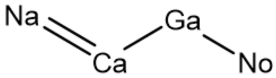
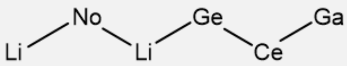
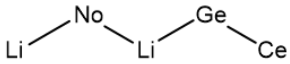
b) CHEMBL1221272

c) CHEMBL1268987

Figure 3-9: RGs of the Markush structures in each DOCCHEMBLID a) CHEMBL1157114 and CHEMBL2218064 *(Chambers et al., 2010)* b) CHEMBL1221272 *(Abdi et al., 2010)* c) CHEMBL1268987 *(Abberley et al., 2010)*

As the P2x7 Subset is a subset of the P2x7 dataset, the RG cores extracted from each dataset were compared to see if the RG cores extracted from the subset were also extracted from the full dataset. However, only two of the seven RG cores within the P2x7 Subset were also extracted from the larger P2x7 dataset. The other five RG cores are not present, although a substructure of each of these RG cores is present. The P2x7 Subset RG cores and the related P2x7 RG cores are shown in Table 3-11. The top two cores are the cores that were found in both datasets. The following five either have one or two nodes removed, resulting in more molecules being brought together. In some cases, the number of molecules represented by the core has increased significantly.

Table 3-11: P2x7 Subset RG cores and the related RG cores in the P2x7 dataset. The starred RG cores are ones that represent parts or all of a Markush structure within the P2x7 Subset dataset.

P2x7 Subset Cores	Related P2x7 Cores
 (437)	 (724)
 (208)	 (373)
 (249)	 (276)
 (110)	 (324)
 (102)	 (237)
 (50)	 (264)
 (16)	 (25)

The final step of the RG core extraction process is to identify all the RG cores present within a molecule. The rationale for this process was discussed in the previous chapter and arises because one molecule can map to multiple different RG cores. The average number of RG cores each molecule maps to is shown in Table 3-12. The ideal outcome would be for each molecule to be represented by one RG core so there is no ambiguity. However, for the majority of the datasets the average number of RG cores per molecule is between one and two. For the Cyto and Neurokinin datasets the average number of cores per molecule is above seven and eight, respectively. These large numbers indicate that these datasets were too diverse for the RG cores extraction process to extract meaningful and efficient RG cores. Additionally, and as discussed previously, these two datasets identify RG cores with one and two nodes.

Table 3-12: Table showing the average number of RG core per molecules for each dataset

Dataset	Number of Molecules	Average Number of Cores per Molecules
Bajorath	2549	1.09
CDK2	1368	1.93
Chk1	106	1.42
Cyto	6370	8.18
FactorXa	1956	1.88
Mmp12	2500	1
Neurokinin	2475	7.05
P2x7	2259	2.32
P2x7 subset	691	1.70
p38a	3644	2.58

3.4 Comparison to Other Methods

This section compares the RG cores with representations generated by established methods, including RDKit's function fMCS, Chemaxon's Markush structures and Bemis-Murcko scaffolds.

3.4.1 Comparison to an Alternative Maximum Common Substructure Method

A comparison was made between RDKit's function fMCS and the RG core extraction process. The fMCS ("RDKit: Open-Source Chemoinformatics," 2018) is a RDKit function that identifies the MCS from a list of molecules. The fMCS implementation was developed for chemical graphs rather than RGs, however, it takes input as SMILES strings and given that the RGs are also represented as SMILES it was used here to calculate the MCS from the RG representations of the molecules.

fMCS was first applied at the whole dataset level, however, no MCSs were produced. This is because in the default setup the fMCS function attempts to find an MCS that represents all of the input molecules and the variation of the RGs is too large so that an MCS cannot be found. Therefore, fMCS was then applied to the clustered datasets.

Table 3-13: P2x7 subset RG core results from the different methods

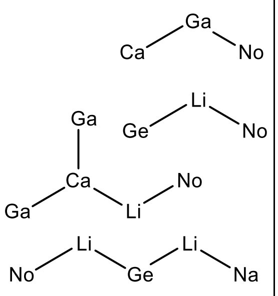
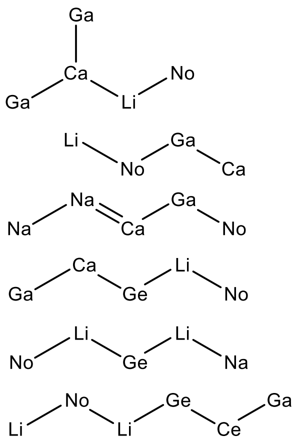
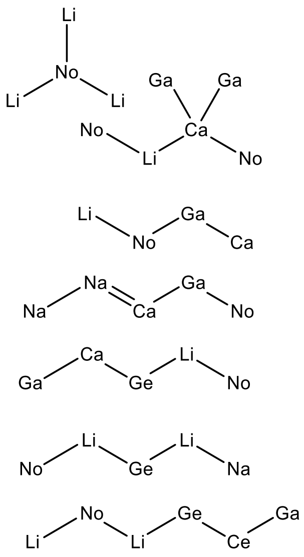
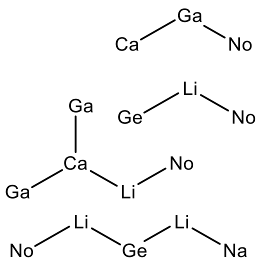
RG Core Extraction (Min Core Size 2 or 3 with similarity 0.5) Clustered Dataset	RG Core Extraction (Min Core Size 4 similarity 0.5) Clustered Dataset	RG Core Extraction (Min Core Size 4 similarity 0.5) Whole Dataset	RDKit fMCS applied to the clustered dataset
			

Table 3-13 shows that the fMCS method extracts the same RG cores from the clustered P2x7 Subset data as the RG core extraction method for minimum core size of 2 or 3 and similarity set to 0.5. However, when the minimum core size is set to 4 for the RG core extraction method, larger RG cores are found that represent superstructures of those found with the smaller minimum nodes threshold and by fMCS. Thus, the RG core extraction method provides greater flexibility than the fMCS approach.

When fMCS was applied to the Bajorath dataset more differences were seen between the two methods. Although the numbers of RG cores extracted are similar, 28 for fMCS compared to 29 for the RG core extraction method, the cores themselves vary, with those extracted using fMCS having a higher average core size of 7.29 nodes compared to 6.31 nodes for the RG core extraction method. The number of RG cores extracted from the whole dataset by fMCS was slightly fewer at 24 unique RG with an average core size of 5.13 nodes.

The RG cores generated from the clustered data were compared for both methods as shown in Table 3-14. The results were identical except for cluster 5, where the fMCS extracts just one RG core and the RG core extraction method produces two RG cores which are superstructures of the core found by fMCS. There were more differences between the RG cores extracted using the whole dataset and the fMCS methods and the comparison is shown in Table 3-15.

Table 3-14: Core extraction from the clustered Bajorath dataset for the RG core extraction and the fMCS methods (threshold level 1)

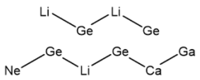
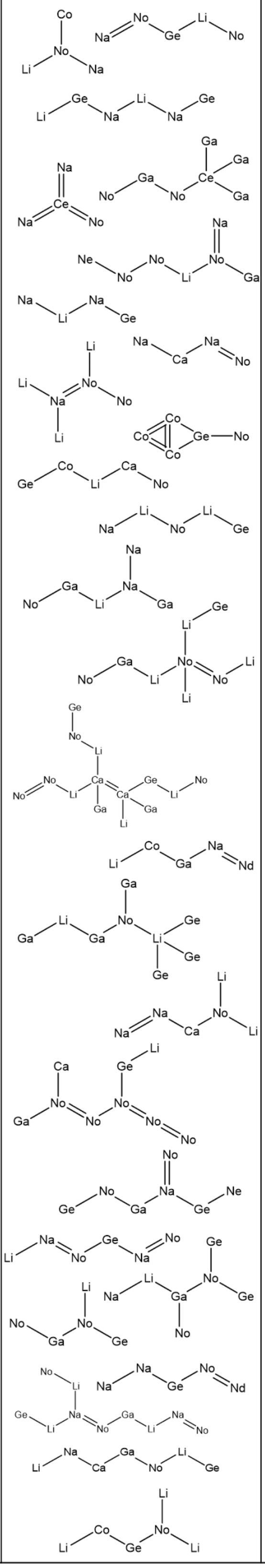
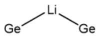
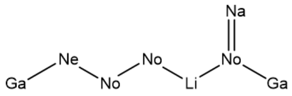
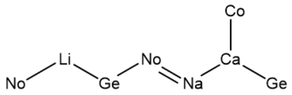
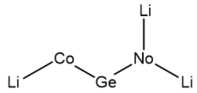
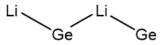
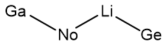
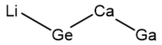
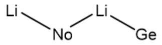
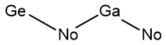
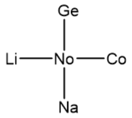
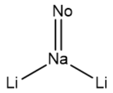
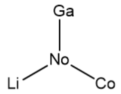
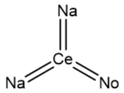
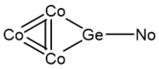
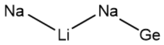
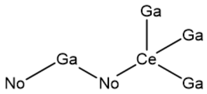
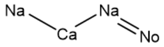
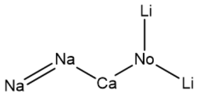
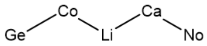
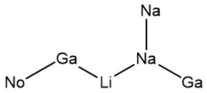
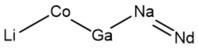
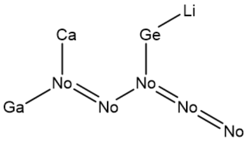
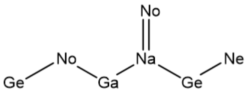
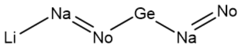
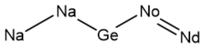
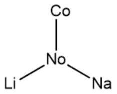
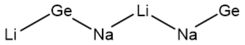
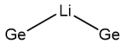
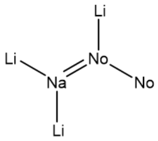
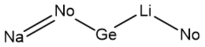
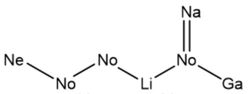
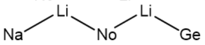
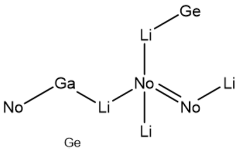
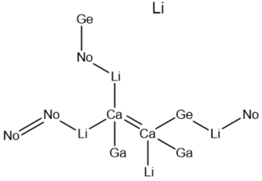
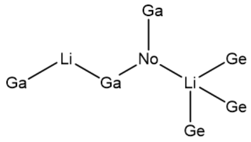
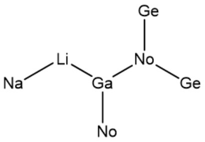
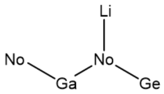
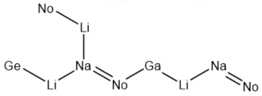
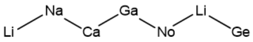
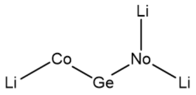
Additional Cores Extracted From the Clustered Dataset RG Core Extraction [2]	Cores Extracted From Both Methods [27]	Additional Cores Extracted From the fMCS Method (clustered) [1]
		

Table 3-15: A table comparing the RG cores extracted from the whole dataset RG core extraction and from the fMCS method (threshold level set to 1) for the Bajorath dataset

Additional Cores Extracted From the Whole Dataset RG Core Extraction [11]	Cores Extracted From Both Methods [13]	Additional Cores Extracted From the fMCS Method (clustered) [15]
          	            	              

It is possible to relax the condition that all molecules must contain the MCS in the fMCS method. The threshold setting was reduced to 0.75 so that only 75% of the molecules need to contain the MCS, to see how this affected the RG cores extracted. The RG cores that were extracted are bigger than the previous RG cores with a threshold setting of 1, however, the RG cores do not always represent all of the molecules in the cluster. Table 3-16 shows the number of molecules that match the fMCS core for each cluster when the similarity threshold was set to 0.75. It can be seen that 153 molecules are not represented by the RG cores extracted, which is six percent of the dataset. In contrast, all of the molecules are represented using the RG core extraction method.

Table 3-16: Bajorath fMCS results for each cluster when the threshold setting is 0.75

Cluster	Number of Molecules	Number of Unique RGs	Number of Molecules represented by fMCS Core	Number of Molecules not represented by fMCS Core
0	100	47	94	6
1	54	34	46	8
2	52	29	47	5
3	55	31	45	10
4	69	37	66	3
5	375	101	314	61
6	135	58	132	3
7	146	35	146	0
8	65	31	64	1
9	53	14	42	11
10	56	15	51	5
11	226	65	226	0
12	182	43	182	0
13	53	10	52	1
14	47	18	43	4
15	2	2	2	0
16	81	29	80	1
17	62	25	61	1
18	60	24	54	6
19	73	22	73	0
20	51	31	47	4
21	91	45	88	3
22	98	41	86	12
23	63	36	55	8
24	88	26	88	0
25	86	20	86	0
26	57	21	57	0
27	69	30	69	0

When larger datasets were examined larger differences between the two methods start to arise. For all the datasets the fMCSs were calculated with the thresholds 1 and 0.75. When examining the results in Table 3-17 and Table 3-18, the RG core extraction method always represents all molecules within the dataset or cluster. fMCS threshold 1 results are provided in the Appendix as the comparison is similar to the results obtained with a threshold of 0.75. However, for both fMCS thresholds, some molecules are not represented, and in some cases, a considerable number of molecules are not characterised. This is detrimental for a method that aims to summarise a dataset. Additionally, for some datasets the fMCS does not generate an RG core for some clusters as the molecules within the cluster are too diverse. For the Neurokinin dataset and fMCS threshold 1, no RG cores were extracted for all the clusters, and therefore, there is no RG core to represent any molecules within the dataset.

The fMCS results are more aligned with those generated by the RG core extraction method when the datasets were pre-clustered due to the molecules being more similar. However, the RG core found using the fMCS method and threshold 0.75 do not always represent all of the molecules in a cluster, Table 3-16. This, therefore, explains why the RG cores were slightly larger from the fMCS clustered dataset as the extracted MCS does not have to represent all of the molecules. Also, for fMCS with a threshold of 1, the RG core sizes were smaller because the fMCS attempts to represent all the molecules within one cluster by one RG core. A comparison of the RG cores details are highlighted in Table 3-17.

Table 3-17: Comparison of the two different methods for all nine datasets. RG core extraction whole dataset results are within the Appendix.

Dataset	Number of Clusters	RG Core Extraction			fMCS									
		Clustered Dataset			Threshold 1 (Clustered Dataset)					Threshold 0.75 (Clustered Dataset)				
		RG Cores	Mean Core Size	Number of Singletons	RG Cores	Mean Core Size	Number of Singletons	Clusters without a core extracted	Molecules Not Represented	RG Cores	Mean Core Size	Number of Singletons	Clusters without a core extracted	Molecules Not Represented
Bajorath	28	29	6.31	0	28	6.29	0	0	0	28	7.29	0	0	84
CDK2	150	195	6.25	89	142	5.93	64	3	30	149	6.33	64	1	91
Chk1	3	9	5.22	3	3	3.00	0	0	0	3	4.00	0	0	11
Cyto	2	181	3.72	16	0	-	-	-	6370	0	-	-	-	6370
FactorXa	112	126	7.07	31	108	6.68	24	1	7	108	7.29	24	0	104
Neurokinin	2	85	3.71	8	0	-	0	2	1483	1	4.00	0	1	1189
P2x7	67	97	6.17	16	58	5.98	16	4	152	64	6.70	16	0	112
P2x7 Subset	4	6	5.00	1	4	4.00	0	0	0	4	6.50	0	0	23
p38α	150	245	5.56	88	113	5.24	51	13	649	136	5.65	51	2	361

Table 3-18: Difference between the fMCS method and the method presented in this chapter on a cluster level and a whole dataset level. Threshold Level set to 0.75.

Dataset	Unique Number of Cores from			Comparison of RG cores for…						
				RG extraction method and fMCS for clustered data				RG extraction method for whole dataset and fMCS for clustered data		
	Cluster Method	Whole Method	fMCS Method	Number of Clusters that Cores Exactly Match	Number of Cores the Same	Number of Unique Cores to Cluster Method	Number of Unique Cores to fMCS Method	Number of Cores the Same	Number of Unique Cores to Whole Method	Number of Unique Cores to fMCS Method
Bajorath	29	24	28	9	10	19	18	7	17	21
CDK2	195	115	145	99	102	93	43	27	88	118
Chk1	9	9	3	1	3	6	0	3	6	0
Cyto	181	182	0	0	0	181	0	0	182	0
FactorXa	126	42	108	61	80	46	28	14	28	94
Neurokinin	85	86	1	0	1	84	0	1	85	0
P2x7	97	58	64	25	33	64	31	10	48	54
P2x7 Subset	6	7	4	0	0	6	4	0	7	4
p38α	245	125	136	69	90	155	46	36	89	100

Table 3-19 shows the number of cores that change between the two threshold settings for fMCS, 0.75 and 1 and in some cases the difference is large. Therefore, if this method were to be used it would need to be optimised to achieve the most optimal RG cores.

Table 3-19: Capturing the number of clusters cores that change in the fMCS method when adapting the threshold setting

Dataset	Number of Clusters	Number of Clusters Cores that change between threshold settings 0.75 to 1
Bajorath	28	19
CDK2	150	36
Chk1	3	2
Cyto	2	NA
FactorXa	112	35
Neurokinin	2	1
P2x7	67	32
P2x7 Subset	4	4
p38α	150	48

From this investigation, it can be considered that both methods could be used to generate RG cores, if the dataset was pre-clustered and the parameters for both methods can be adjusted produce varying RG cores. However, the RG core extraction method is more flexible and the settings can be adjusted to produce a better optimum set of RG cores. Although the RG cores become larger as the threshold for the fMCS is reduced, these are less representative of the dataset as not all of the molecules are represented by RG cores. This is more pronounced for more diverse datasets where the fMCS does not allow RG cores to be extracted with either threshold setting. Additionally, on average, the fMCS with threshold 0.75 generates slightly larger RG cores than the RG core from the RG core extraction. However, this is only slight and does not outweigh the benefits of applying the RG core extraction method to the whole dataset.

Furthermore, the RG core extraction method applied to the whole datasets produces results that represent all molecules with considerably less computation as there is no need for the data to be clustered.

3.4.2 Comparison to Markush Structures

In this section, the RG cores are compared with Markush structures generated using ChemAxon's Markush Editor (“ChemAxon Markush Editor,” 2020). The Markush structures were made using all default settings such as: enable nested R-groups; enable position variation; enable atom lists; merge duplicate R-groups; the scaffold option was automatic detection rather than user-defined; calculation mode was set to normal; the minimum scaffold size was two; and the minimum common size for nesting R-groups was three.

Only two Markush structures were generated for the P2x7 Subset, shown in Figure 3-10, compared to the seven RG cores, Figure 3-11. It can be seen that the two Markush structures are smaller than the RG cores represented for this dataset, as these two Markush structures correspond to one or two nodes, respectively, whereas the average RG core size is five. These Markush structures initially do not easily allow a chemist to see a relationship between the molecules. The structures identified are small and could be considered common to a wide number of molecules.

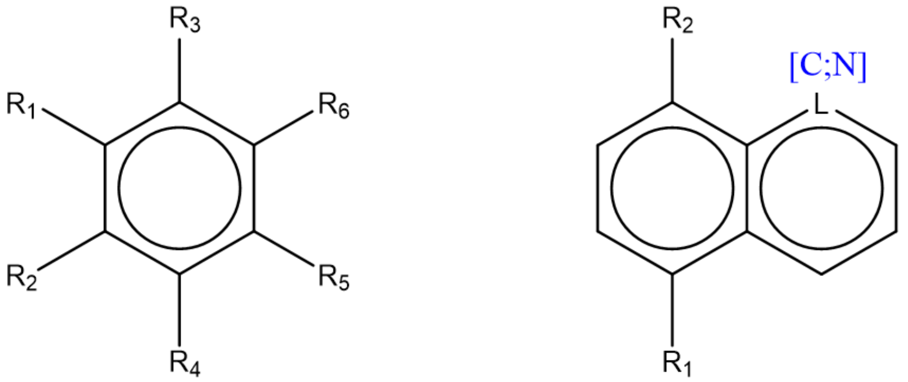
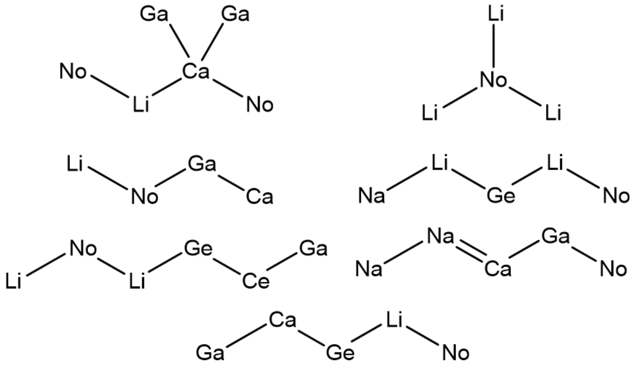


Figure 3-10: Markush structures generated for the p2x7 subset dataset



Node Definition	SMILES Code
Acyclic Inert	Li
Acyclic HBA	Ga
Acyclic HBA-HBD	Ge
Aromatic Inert	No
Aromatic HBA	Na
Aliphatic HBA	Ca

Figure 3-11: RG cores identified from p2x7 subset dataset

The ChemAxon Markush Editor tool provides the option to drill down the representation via the R group substituents which can be nested and contain further R-groups. An example of this breakdown is demonstrated in Figure 3-12, which shows the R groups that were associated with the second of the

Markush structures. Both R1 and R2 were fragmented further. A greater breakdown and analysis of the R-groups is not currently provided in the method presented within this thesis.

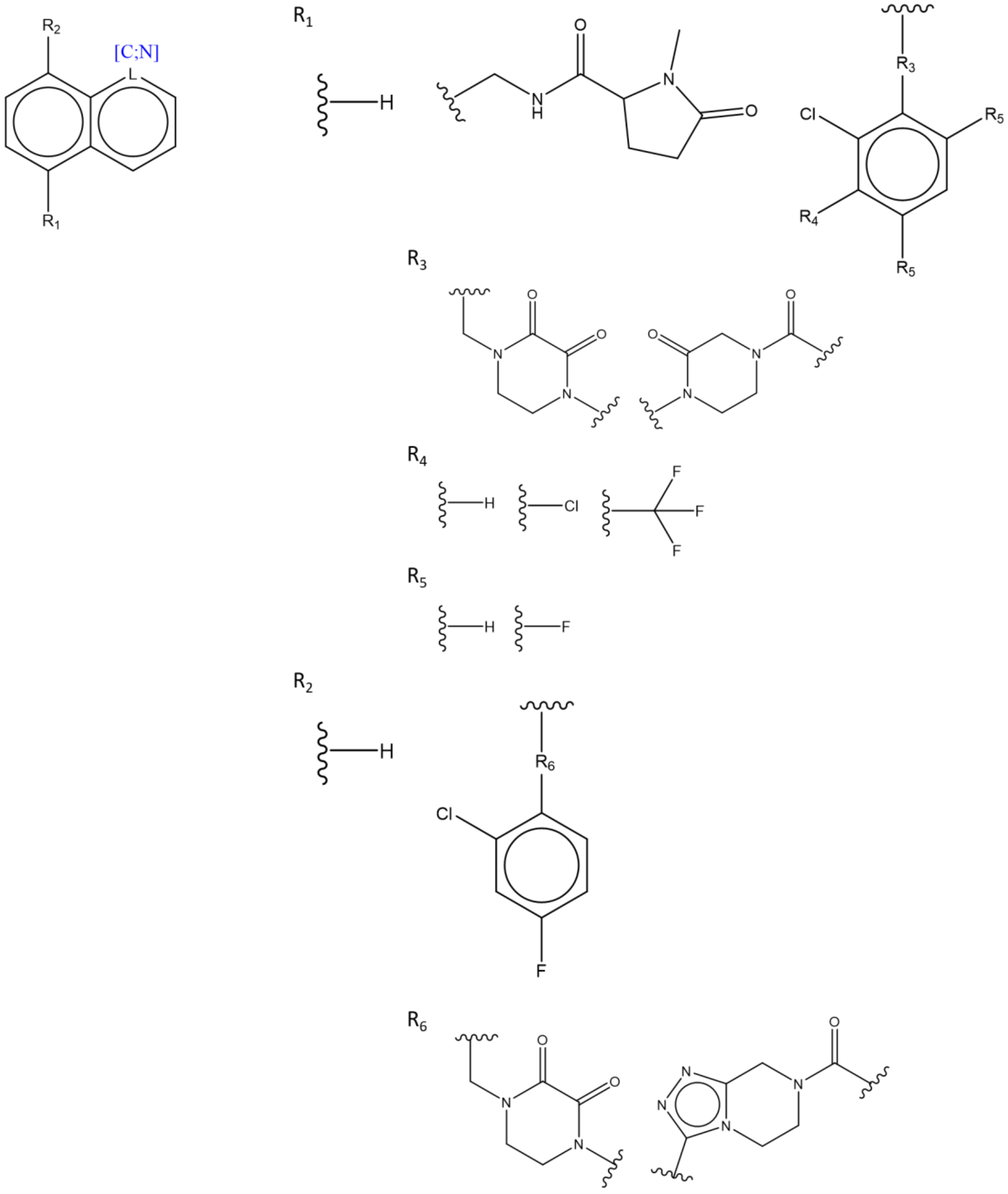


Figure 3-12: Second Markush structure along with R-groups

The Markush structures were generated for the rest of the datasets. Table 3-20 shows that all of the datasets produce more RG cores than Markush structures. As for the P2x7 Subset, the reduced number of Markush structures is because they are less complex and represent less information in the main scaffold, shown by the overall reduction in the median number of atoms. Therefore, the highest level Markush structures are not as useful as the RG cores to clearly identify structural relationships between the molecules within a dataset. The Markush structures also do not allow for closely related structures to be compared. In contrast, the RG cores allow variation within the core, as long as the pharmacophoric features remain the same. When examining the Markush structures they are all small cores, and are mainly either a ring or a fused ring system, whereas, the RG cores are constructed from more than just the ring or ring systems.

Table 3-20: Table showing the comparison between the number of Markush structures and RG scaffolds extracted

Dataset	Number of Markush Structures	Median Number of Atoms within Markush Structures	Number of RG Cores	Median Number of Atoms within RG Core
Bajorath	7	6	24	16
CDK2	16	6	114	14
Chk1	3	6	9	18
Cyto	63	N/A *	182	7
FactorXa	8	6	42	15
MMP12	1	20	1	21
Neurokinin	2	7.5	86	14
P2x7 Subset	2	8	7	15
P2x7	5	6	58	12
P38α	11	6	125	16

*Cyto only shows five random Markush Structures once calculated.

3.4.3 Comparison to Bemis-Murcko Scaffolds

Along with Markush structures, Bemis-Murcko scaffolds are also sometimes used to determine scaffolds. Bemis-Murcko scaffolds can be constructed in three ways: removing side chains, replacing all heteroatoms with carbons, and finally, a scaffold that combines both methods to replace all heteroatoms and side chains (Bemis & Murcko, 1996). Therefore, all of these methods build a framework based on the ring systems within a molecule. An example of how all three of these scaffolds relate to a molecule, A01B01 from MMP12 dataset, is shown in Figure 3-13.

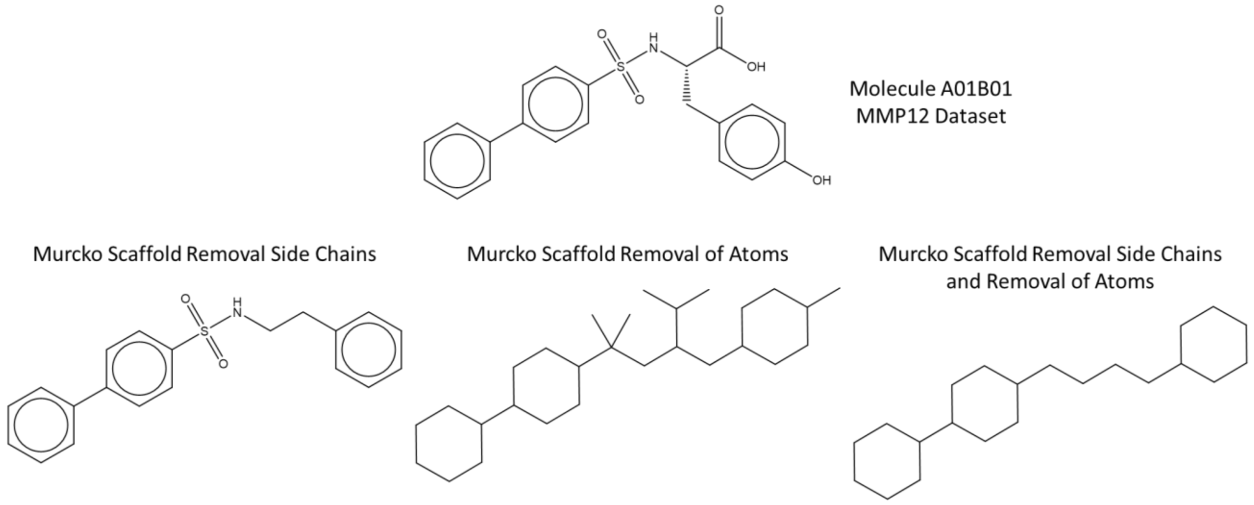

Molecule A01B01
MMP12 Dataset
Murcko Scaffold Removal Side Chains
Murcko Scaffold Removal of Atoms
Murcko Scaffold Removal Side Chains and Removal of Atoms
O
H
N
S
OH

Figure 3-13: Bemis-Murcko scaffolds for molecule A01B01 from the MMP12 dataset

The number of unique representations for each dataset for all three Murcko scaffolds methods are reported in Table 3-21. For all implementations, a much larger number of Murcko scaffolds were found than RG cores. The RG cores provide more details of the actual atomic structure as two of the Murcko methods remove this detail. However, the Murcko scaffolds are larger.

Table 3-21: Table showing the number of Murcko scaffolds extracted from all dataset using all three methods

Dataset	Number of Murcko Scaffolds			Number of RG Cores
	Removal Side Chains	**Remove all Carbons**	**Remove Side Chains and Carbons**	
Bajorath	969	1726	470	24
CDK2	661	1018	397	114
Chk1	60	99	39	9
Cyto	3190	5169	1409	182
FactorXa	867	1356	414	42
MMP12	194	1220	114	1
Neurokinin	1194	1888	545	86
P2x7 Subset	137	413	55	7
P2x7	751	1422	316	58
P38α	1405	2546	698	125

Considering the MMP12 dataset which is based on one Markush structure, which is also extracted from the RG core extraction process, even the most generic Murcko scaffold produces 114 scaffolds. These Murcko scaffolds are therefore not as good as the RG cores for providing information about the relationships within the dataset, although they would deliver more details about the shape and overall framework of the molecules in the dataset.

3.5 Conclusions

Two different ways of extracting RG cores from a dataset were explored. The first extracted RG cores after clustering the dataset and the second extracted RG cores from the dataset as a whole. The results presented here show that it is more effective to extract the RG directly from the dataset as a whole rather than from pre-clustered data.

There are several reasons for this. The first is that there is no easy way to select the most appropriate clustering as it is difficult to compare the clustering indexes across methods, which was demonstrated in the previous chapter. The clustering methods used in the previous chapter do not seem to produce suitable clusters. Furthermore, the clustered datasets do not sufficiently organise the molecules to demonstrate an improvement on the relationships identified within the dataset. The superiority of the RG cores extracted from the whole dataset is confirmed by them being more concise and superior representations of the relationships in the dataset, reflected by their larger scaffold scores, on average, compared to those generated from clustered data.

There were three main hopes for the work undertaken. The first is that it would be able to deal with slight variations within a core structure as this is a current issue with Markush structures and SAR tables as it can lead to multiple representations. The second is that it would be able to bring together scaffolds as well as molecules from different journals. The final hope is that it would be able to represent the analogue series that formed the Bajorath dataset. All three were achieved as the RG cores deal with a slight variation within molecular structures as long as they are retaining the same RG node type. The bringing together of molecules was demonstrated as for three datasets, Neurokinin, P2x7 and P2x7 Subset, the RG cores represent molecules across different journal articles. Finally, all of the analogue series were represented in 28 RG cores, however, some were combined together, as of the Bajorath dataset is constructed from 34 analogue series.

The RG core extraction method was shown to be advantageous over using RDKit's fMCS function since the datasets must be clustered for the latter increasing the computational time but not substantially increasing the quality of the RG cores. The RG cores from the fMCS method, also, do not represent all of the RGs within a dataset and, therefore, this is not as reliable as the method outlined within this thesis.

Furthermore, the benefits of the RG cores over the Markush structures and the Murcko scaffolds were demonstrated. The Markush structures represent more molecules, however, this is due to them being considerably smaller than the RG cores. The Murcko scaffolds contain less atomic data or reflect a small number of molecules even though they are larger than the RG cores.

The RG cores that have been extracted from the whole dataset shall be taken forward for more analysis. The next chapter will look at how it is best to visualise these RG cores to understand better and identify the relationships.

4 Reduced Graph Core Mapping

4.1 Introduction

The previous chapter demonstrated the optimum extraction methods for the reduced graph (RG) cores to represent lead optimisation (LO) series. This chapter demonstrates how the RG cores are mapped back onto the RGs in order to extract substructural information for each node. The RG core mapping is a pre-processing step to organise the data in a dataset so that it can be visualised appropriately, which will be examined in the next chapter.

4.2 Pre-Processing Cleaning

As discussed in the previous chapters, disconnected RG cores were discovered in some of the datasets and so an additional pre-processing cleaning step has been added. The molecules undergo two new steps, a salt remover step and a neutralising step. The salt remover step is a function within RDKit that extracts the largest fragment as the main molecule (“RDKit: Open-Source Chemoinformatics,” 2018). There are some instances where this function still returns a disconnected SMILES, for example, when two fragments are the same size. The disconnected SMILES that are generated in these instances are kept.

The second step is to neutralise the molecules. This neutralisation step has been added as the RG definitions used within this thesis do not take into consideration charged atoms. The neutralisation step is the O’Boyle neutralisation code using RDKit (O’Boyle, 2019). This algorithm neutralises the molecules on an atom-by-atom basis if there are no neighbouring counter charges to prevent groups such as nitro groups being altered. The RDKit rdMolStandardize uncharger function was not used as this does not alter the charges on an atom if there is a counter charge anywhere within the molecule to prevent a change in the overall molecular charge.

For six of the datasets, both steps alter some of the RGs generated as well as the RG cores, as shown in Table 4-1. These datasets are marked by an asterisk.

Table 4-1: Table showing the change in the molecules at each stage of the new cleaning steps. Row contain an asterisk indicate that the RG cores have also been altered.

Dataset	Number of Molecules	Number of Molecules…			Molecules Filtered Out as Now Duplicates	Number of Molecules RGs altered	Number of RG Cores
		Changed Due to Salt Removed	Changed Due to Neutralisation	Changed Overall			
Bajorath	2549	0	0	0	0	0	24
CDK2	1368	0	480	480	0	446	114*
Chk1	106	0	0	0	0	0	9
Cyto	6370	252	59	266	60	220	187*
FactorXa	1956	0	1317	1317	0	668	44*

Mmp12	2500	0	0	0	0	0	1
Neurokinin	2475	92	7	94	4	92	84*
P2x7	2259	25	0	25	0	25	61*
P2x7 subset	691	0	0	0	0	0	7
p38a	3644	0	894	894	0	871	123*

Table 4-2 displays the number of RG cores for each of the datasets together with the number of these that vary following this pre-processing. Some of the RG cores vary by a single node only, as the neutralisation step has caused a change in the node type. Some variations occur due to the removal of salts.

Table 4-2: Number of RG cores that vary from the previous chapter

Dataset	Number of RG Cores	Number of RG Cores That Remain The Same	Number of RG Cores That Are Different
CDK2	114	70	44
Cyto	187	121	66
FactorXa	44	21	23
Neurokinin	84	53	31
P2x7	61	48	13
p38a	123	82	41

4.3 Reduced Graph Core Mapping Procedure

To create the visualisation and examine the amount of chemical space that has been explored the substructural chemical graph features need to be extracted. This is achieved by mapping the RG of each molecule on to the corresponding RG core and analysing the metadata associated with each of the nodes that map to the RG core. Mapping of a molecule back onto the core is not trivial, as there may be multiple ways in which a molecule could be mapped. Figure 4-1 illustrates a molecule within the MMP12 dataset with two different mappings onto an RG core. The MMP12 dataset only generates one RG core which is based on one Markush structure. The RG of the molecule has two Ge nodes attached to the linker node representing the substructures labelled as (1) and (2). Both of these can map onto the starred Ge node in the core. Each mapping would lead to a different substructure being associated with the node and it is therefore necessary to select the mapping that best represents the LO series. The metadata stored with the RG nodes of each molecule includes information on the substructural fragment that each node represents along with information on stereochemistry (when present in the molecule) and substitution patterns and is used to resolve ambiguous mappings as described below.

The numbers of molecules where there are multiple potential mappings to the associated RG core are displayed in Table 4-3 for the nine datasets. The table also shows the average number of RG cores each molecule maps to. For the majority of the datasets, the average number of RG cores each molecule maps to is between 1 and 2.68; the exceptions are the datasets Cyto and Neurokinin datasets where the average is more than 7. The number of mappings to consider is therefore more than the number of molecules in the dataset as multiple molecules map to multiple RG cores.

Table 4-3: Table showing the number of mappings to create and the number of molecules that can have multiple mappings

Dataset	Number of Molecules	Average Number of Cores Each Molecule Maps To	Number of Mappings	Number of Molecules that have multiple mappings for an RG core
Bajorath	2549	1.09	2774	377
CDK2	1368	1.93	2637	355
Chk1	106	1.42	151	2
Cyto	6370	7.31	46150	15192
FactorXa	1956	2.22	4333	668
Mmp12	2500	1	2500	600
Neurokinin	2475	7.37	18203	9316
P2x7	2259	2.36	5342	961
P2x7 subset	691	1.70	1172	290
p38a	3644	2.68	9752	1814

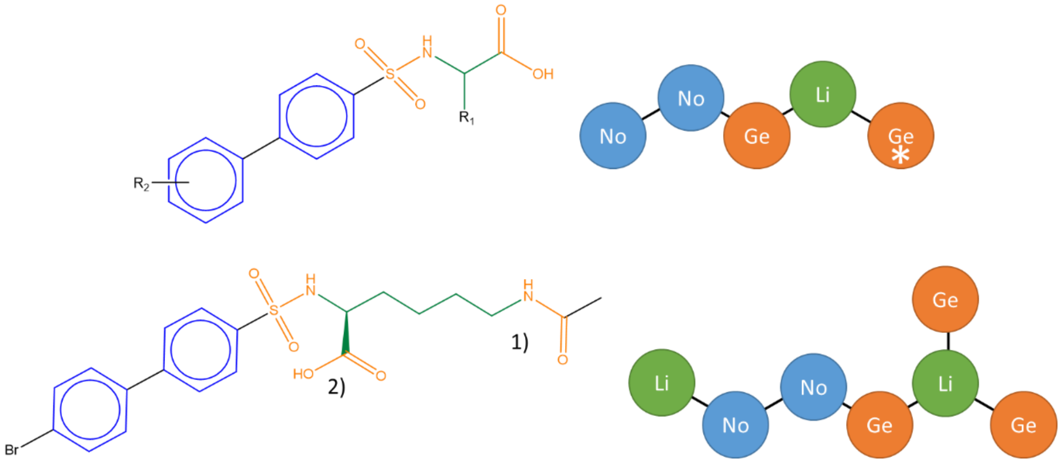
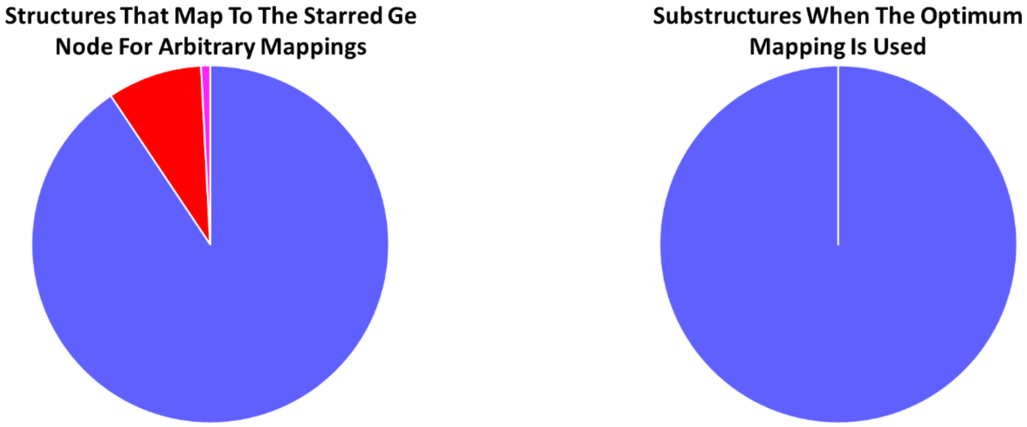
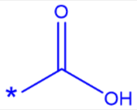
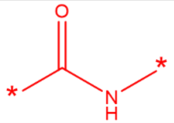
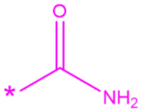


Figure 4-1: The Markush structure (with R-groups) is shown at the top together with RG core extracted from the dataset. The molecule bottom left has two different mappings to the starred Ge node, substructural fragment 1 or 2.

When there are multiple ways of mapping a molecule to an RG core, in the majority of cases one of the mappings is the "correct" mapping as it maximises the overlap between the molecules when the underlying substructures are considered. For example, in the case shown in Figure 4-1, the correct mapping is substructure (2) since the majority of the molecules represented by the core have a COOH substructure at this position. There are 1704 molecules that map to this RG core for this dataset, many of which have multiple mappings. Figure 4-2 shows the effect of simply taking the first mapping compared with resolving the mappings according to the majority substructure in the molecules. The first method on the left identifies three substructural fragments with varying number of examples, whereas, all 1704 molecules contain the COOH mappings as one of the multiple mappings. This example shows the importance of attempting to resolve multiple mappings in order to create a visualisation that is representative of how the LO series was developed.



Substructural Fragment	Number of Examples
	1544
	146
	14

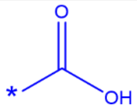
Substructural Fragment	Number of Examples
	1704

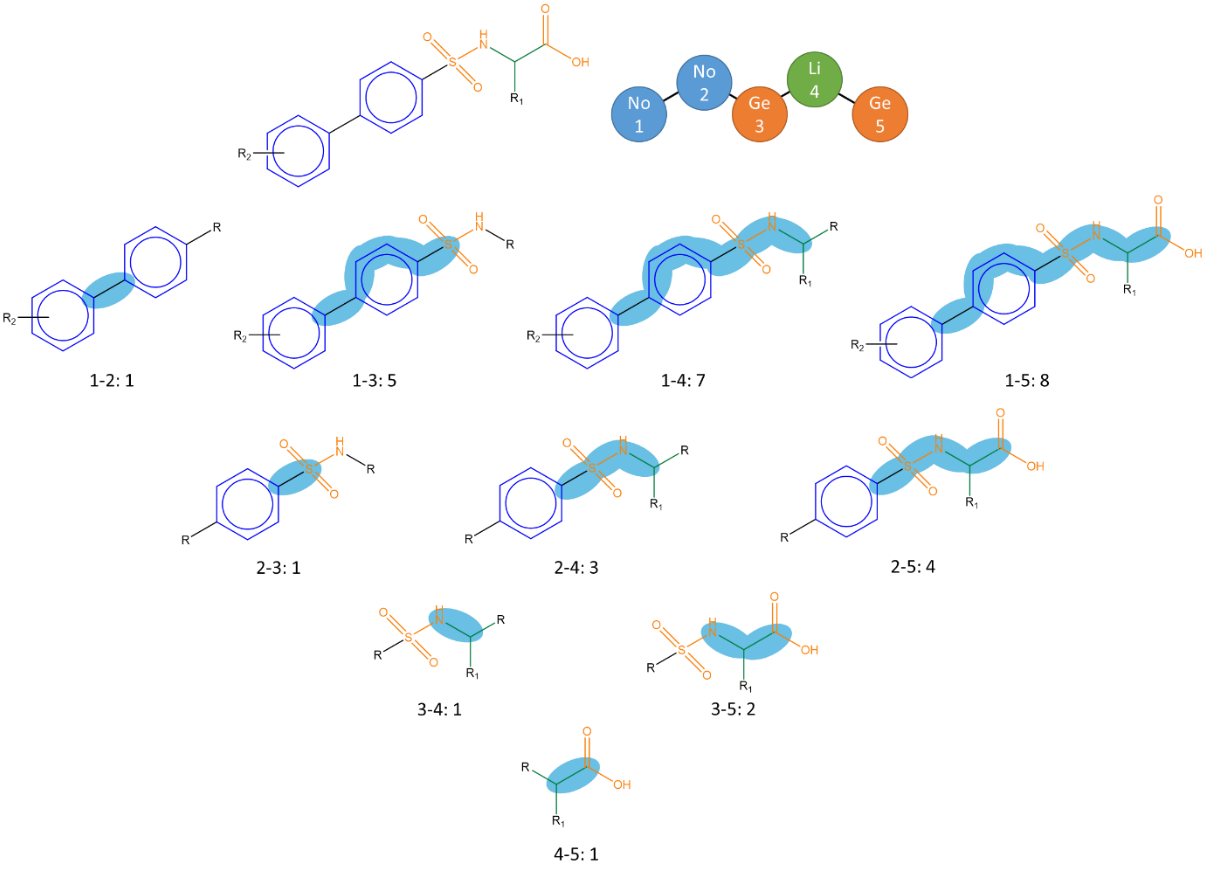
Figure 4-2: Demonstrating the difference between selecting the first seen mapping and the mapping that maximise the overlap between all molecules

Ideally, just one mapping should be identified per molecule, otherwise, the analysis may not be representative of the underlying LO series. Furthermore, multiple mappings will impact on further analysis as the number of substructures represented by a node would be more than the number of molecules.

Therefore, where there are multiple potential mappings, each instance has to be analysed and considered with one being selected. The benefits of the RG should be retained as they are an abstract representation that allows different substructures to be brought together so it is not desirable to look for exact matches at the substructure level. Instead, the different substructures are compared based on topological distances, since the distance between key features in a molecule can be important for binding to a biological receptor.

4.3.1 Methodology

For each RG core, the molecules that have only one potential mapping are mapped first and the metadata for each of the nodes is extracted. The unique mappings then allow a basis for choosing mappings for those molecules where multiple mappings are possible. As the main goal is to maximise the overlap between the molecules, inspiration is taken from the node-bond-pair fingerprint in Barker et al. and a node topological distance map is created for each molecule (Barker et al., 2006). The node topological distance map is created by finding the shortest topological distance between each pair of RG nodes which is the shortest bond length between any pairs of atoms where one atom is taken from each node. An example of a node topological distance map is shown in Figure 4-3. Each node is labelled, and the topological distance between each node is found, for example, 1-2:1 indicates that the topological distance between nodes 1 and 2 is 1. Each mapping to the RG core will give rise to a different topological distance map.



Topological Distance Map
{1-2: 1, 1-3: 5, 1-4: 7, 1-5: 8, 2-3:1, 2-4: 3, 2-5: 4, 3-4: 1, 3-5: 2, 4-5: 1}

Figure 4-3: Example of the node topological distance map for a molecule

A substituent topological distance map is also created by finding the shortest topological distance between each node and each of the R-group substitution sites within the RG core. An example of a substituent topological distance map is shown in Figure 4-4. As before, the nodes are labelled and the topological distance to each of the substitution sites is found, for example, 2: [3, 4, 4] indicates that the topological distance between node 2 and the three different substituent sites is 3, 4 and 4, respectively.

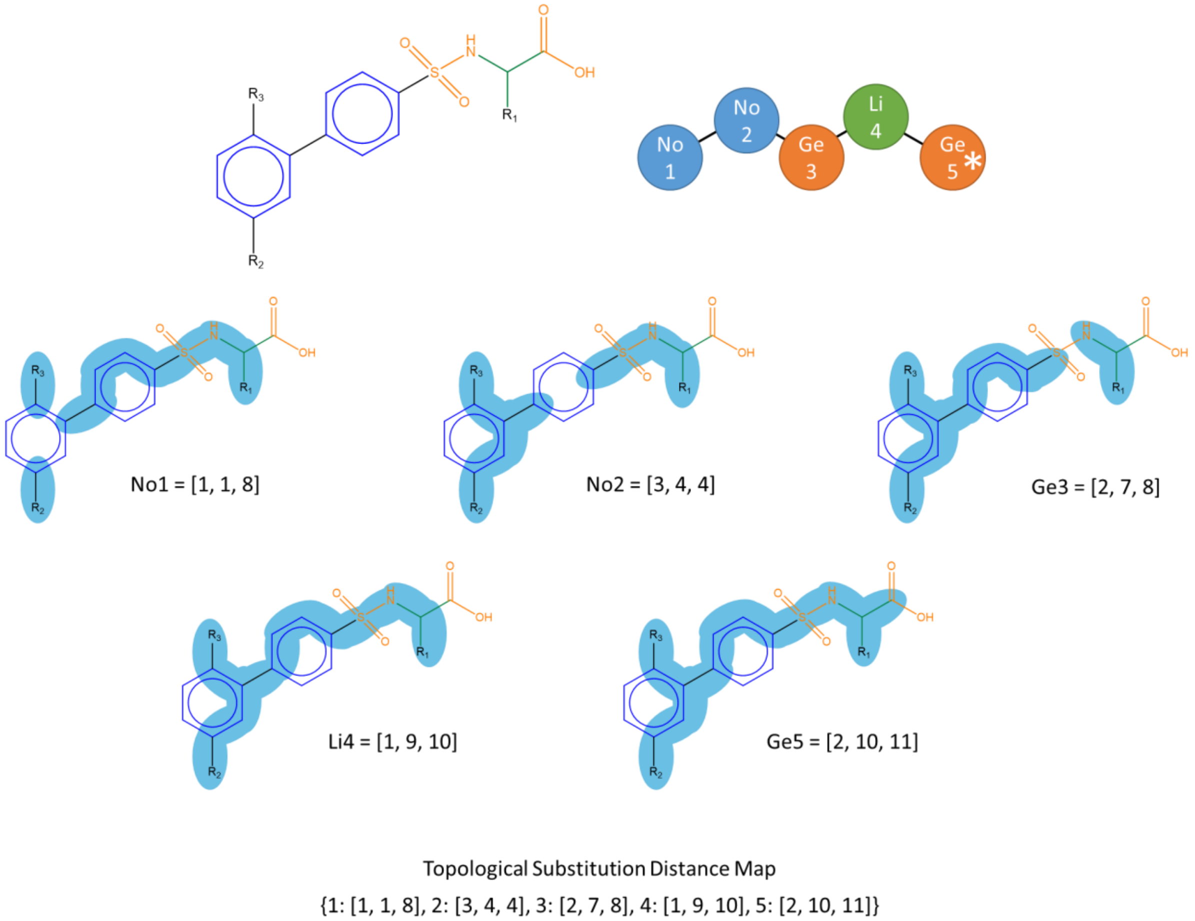


Figure 4-4: Example of the substituent topological distance map for a molecule

The workflow used to identify the optimal mapping to an RG core is shown in Figure 4-5. Following each stage, if the potential mappings still cannot be resolved they are taken forward to the next stage of analysis. The first step is to calculate the topological distance maps and the substituent topological distance maps for all molecules. Those molecules that have a unique mapping to the RG core are then identified and their topological distance maps are aggregated. The molecules with multiple mappings are then studied.

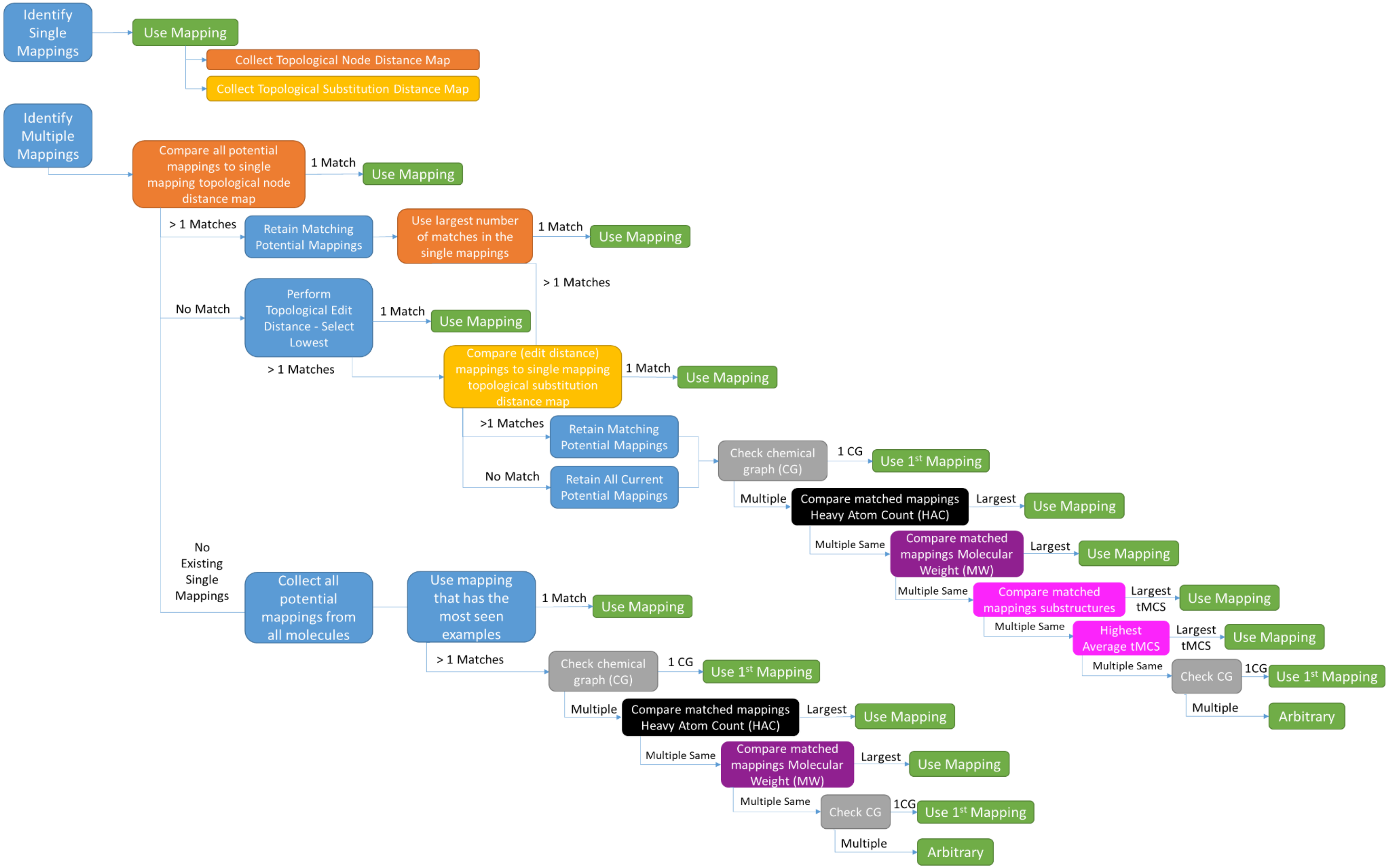

Identify Single Mappings
Use Mapping
Collect Topological Node Distance Map
Collect Topological Substitution Distance Map
Identify Multiple Mappings
Compare all potential mappings to single mapping topological node distance map
1 Match
Use Mapping
> 1 Matches
Retain Matching Potential Mappings
Use largest number of matches in the single mappings
1 Match
Use Mapping
> 1 Matches
No Match
Perform Topological Edit Distance - Select Lowest
1 Match
Use Mapping
> 1 Matches
Compare (edit distance) mappings to single mapping topological substitution distance map
1 Match
Use Mapping
>1 Matches
Retain Matching Potential Mappings
No Match
Retain All Current Potential Mappings
Check chemical graph (CG)
1 CG
Use 1st Mapping
Multiple
Compare matched mappings Heavy Atom Count (HAC)
Largest
Use Mapping
Multiple Same
Compare matched mappings Molecular Weight (MW)
Largest
Use Mapping
Multiple Same
Compare matched mappings substructures
Largest tMCS
Use Mapping
Multiple Same
Highest Average tMCS
Largest tMCS
Use Mapping
Multiple Same
Check CG
1CG
Use 1st Mapping
Multiple
Arbitrary
No Existing Single Mappings
Collect all potential mappings from all molecules
Use mapping that has the most seen examples
1 Match
Use Mapping
> 1 Matches
Check chemical graph (CG)
1 CG
Use 1st Mapping
Multiple
Compare matched mappings Heavy Atom Count (HAC)
Largest
Use Mapping
Multiple Same
Compare matched mappings Molecular Weight (MW)
Largest
Use Mapping
Multiple Same
Check CG
1CG
Use 1st Mapping
Multiple
Arbitrary

Figure 4-5: Workflow indicating how mappings to the RG core are found

For a molecule with multiple mappings, each mapping is compared to the aggregated node topological distance maps derived from the molecules with unique mappings. If there is a single match then this mapping is used. If there are multiple matches, then the mapping with the largest number of previously seen examples in the aggregated maps is used. If there is more than one such match, then the substituent topological distance maps from these aggregated maps are compared. If there is no matching topological distance map, then a topological edit distance is carried out to select the mapping with the lowest edit distance.

Edit distance is used to quantify the distance between two objects by identifying the number of operations (insertion, deletion or replacement) needed to change one object into another. As the mappings should maximise the alignments between molecules, the mapping with the smallest topological edit distance to the aggregated maps is chosen. An example of how the topological edit distance is calculated is shown in Figure 4-6. For each node-node pairing the topological distance between both nodes is compared. This number indicates how many bonds need to be added or removed to make the bond structure overlap. The total of all these comparisons is found and becomes the topological edit distance. The topological edit distance for the topological distance maps shown in Figure 4-6 is 12. If there are multiple examples with the smallest topological edit distance, then the one with the lowest average distance compared to all the aggregated maps is selected.

If there is more than one mapping with the smallest topological edit distance, then the substituent topological edit distance maps are compared. For each node the number of insertions and deletions are calculated, Figure 4-7 illustrates an example of how the substituent topological edit distance is calculated. The lower the total number, the larger the overlap. The example with the lowest total is selected, unless there are multiple examples with the same lowest total, then the lowest average across all the aggregated substitution maps is used. If there is only one match this is the RG core mapping that is used. However, if there are multiple matches these all go forward onto the next stage or if there are no matches all the potential mappings that had previously not been filtered out go forward onto the next stage.

Topological Distance Map 1	Topological Distance Map 2	Topological Distance Map Edit Distance
{1-2: 1,	{1-2: 1,	{1-2: 0,
1-3: 5,	1-3: 5,	1-3: 0,
1-4: 7,	1-4: 7,	1-4: 0,
1-5: 12,	1-5: 8,	1-5: 4,
2-3: 1,	2-3: 1,	2-3: 0,
2-4: 3,	2-4: 3,	2-4: 0,
2-5: 8,	2-5: 4,	2-5: 4,
3-4: 1,	3-4: 1,	3-4: 0,
3-5: 6,	3-5: 2,	3-5: 4,
4-5: 1}	4-5: 1}	4-5: 0}
		Total = 12

Figure 4-6: Comparison of the edit distance between two topological distance map, each node to node distance is compared to generate an overall total edit distance of the topological distance maps

Topological Substitution Distance Map 1	**Topological Substitution Distance Map 2**	**To Add**	**To Remove**	**Total Transformations**
1: [7, 8],	1: [3, 4, 7, 8],	3, 4	-	2
2: [5, 6],	2: [1, 1, 5, 6],	1, 1	-	2
3: [4, 5],	3: [3, 4, 4, 5],	3, 4	-	2
4 : [1, 1]	4 : [1, 1, 4, 5]	4, 5	-	2
				= 8 Total

Figure 4-7: Example of the how the topological substitution edit distance is calculated

If the multiple mappings are still not resolved, then the chemical graphs are checked to determine if the chemical graphs of the unresolved mappings are the same, if they are there is no way to distinguish between them and so one is chosen arbitrarily (the first instance encountered) Figure 4-8. When there are multiple chemical graphs then the heavy atom counts (HAC) are compared and the mapping with the largest HAC is used. The largest HAC is selected as this is likely to be a larger molecule and take up a larger space within a potential binding pocket. If the HACs are the same then molecular weights are considered and the mapping with largest molecular weight is used.

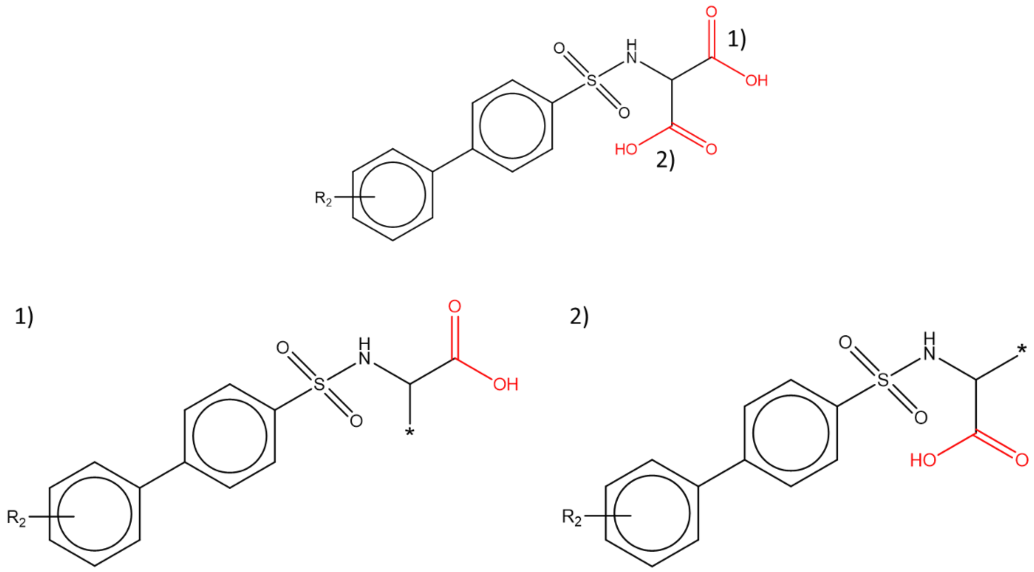


Figure 4-8: An example where the CG does not differ and is therefore indistinguishable

If this still does not distinguish between the mappings, then the Tanimoto maximum common substructure (tMCS) of the chemical substructures of the potential mappings are found with all the substructure in the unique mappings. The mappings with the largest tMCS is used, if there are multiple mappings with the same largest tMCS then the highest average tMCS across all the unique mappings is found for all the potential mappings, Figure 4-9. When there are examples with the highest average tMCS the chemical graphs are compared again.

There are very few examples where this workflow is unable to resolve multiple mapping, and for those cases where it does not identify a single mapping then an arbitrary one is chosen from the remaining examples.

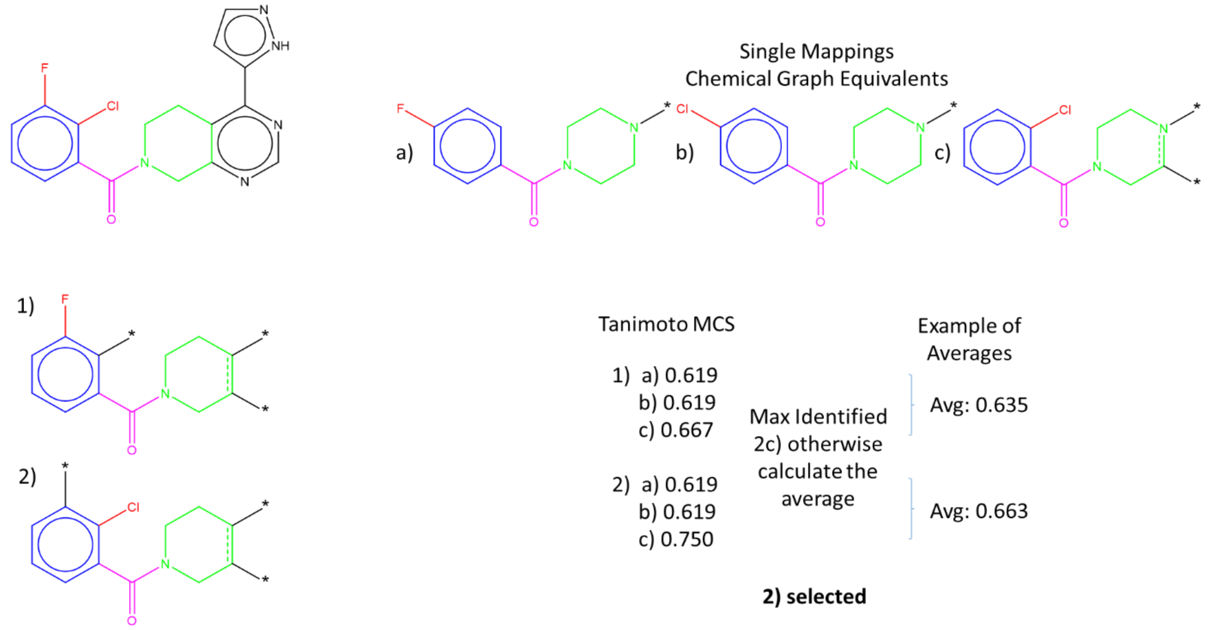
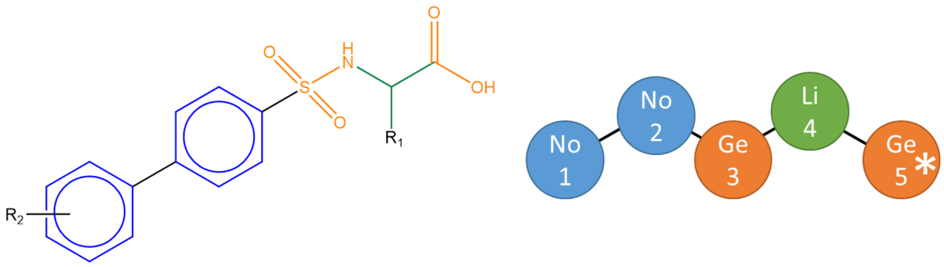


Figure 4-9: An example of how the Tanimoto MCS is calculated where 1) and 2) are potential mappings within the molecule and a) b) and c) are chemical graph RG core equivalents from the single mappings

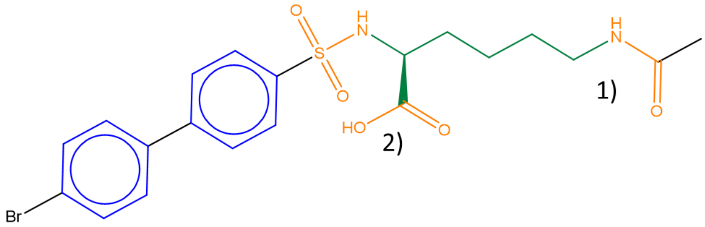
There are some instances where there are no molecules with unique mappings to the RG core. In this case, the node topological distance maps are collected from all molecules, along with the number of times that they appear. This enables the most common node topological distance map to be found. For a given molecule, if there is only one mapping that contains the most frequent node topological distance map then this mapping is used. If there are no examples, then the next most common topological distance map is used. Also, if there are multiple examples then some of the same procedures as before are followed with the chemical graph checked, then the HAC, then the MW and finally the chemical graph again. Finally, an arbitrary mapping is selected if none of these steps identify a preferred mapping.

Some examples are given below.

The example in Figure 4-10 is from the MMP12 dataset. 1900 molecules have a unique mapping to the core and identical topological distance maps. Molecule A06B02 has two potential mappings to the RG core due to the presence of two Ge, hydrogen bond acceptor and donor aliphatic nodes, at the right side of the molecule labelled 1) and 2). The node topological distance maps for these two alternative mappings are shown and just one of these is present in the aggregated maps (representing substructure 2) and so this mapping is selected.



{1-2: 1, 1-3: 5, 1-4: 7, 1-5: 8, 2-3: 1, 2-4: 3, 2-5: 4, 3-4: 1, 3-5: 2, 4-5: 1} -> 1900



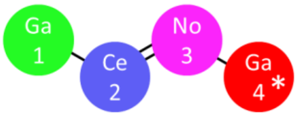
1) {1-2: 1, 1-3: 5, 1-4: 7, 1-5: 12, 2-3: 1, 2-4: 3, 2-5: 8, 3-4: 1, 3-5: 6, 4-5: 1} -> 0
2) {1-2: 1, 1-3: 5, 1-4: 7, 1-5: 8, 2-3: 1, 2-4: 3, 2-5: 4, 3-4: 1, 3-5: 2, 4-5: 1} -> 1900

2) is selected

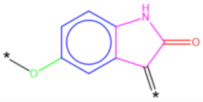
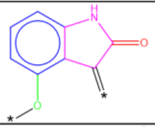
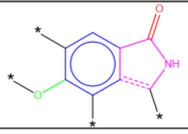
Figure 4-10: Molecule with ID A06B02 within the MMP12 dataset containing multiple examples and how the selection process occurred.

Figure 4-11 shows an RG core [Ga][Ce]=[No][Ga] for the CDK2 dataset. 28 molecules have single mappings to the core and these give rise to three topological distance maps due to different substituent positions on the rings. Examples are given in the table: variant 1) represents 24 molecules; variant 2) represents five molecules; and variant 3) represents a single molecule. The molecule 50415235 has multiple mappings to the RG core and two topological distance maps consistent with variant 1) and variant 3), respectively. Mapping 1) is chosen as this is more prevalent in the maps aggregated from the molecules with unique mappings.

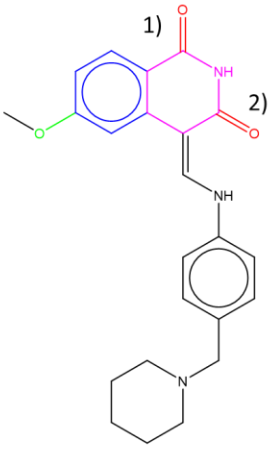
Reduced Graph Core



Single Mappings

Topological Distance Map	Number of Single Mapping Occurrences	Example
{1-2: 1, 1-3: 3, 1-4: 6, 2-3: 0, 2-4: 3, 3-4: 1}	22	
{1-2: 1, 1-3: 3, 1-4: 5, 2-3: 0, 2-4: 2, 3-4: 1}	5	
{1-2: 1, 1-3: 2, 1-4: 6, 2-3: 0, 2-4: 3, 3-4: 1}	1	

Multi Mapping Molecule



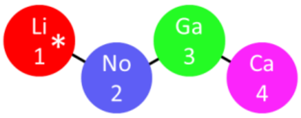
Potential Mappings
1) {1-2: 1, 1-3: 2, 1-4: 6, 2-3: 0, 2-4: 3, 3-4: 1} -> 1
2) {1-2: 1, 1-3: 3, 1-4: 6, 2-3: 0, 2-4: 3, 3-4: 1} -> 22

2) is selected

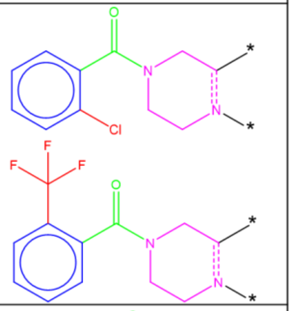
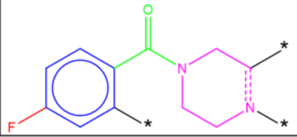
Figure 4-11: Workflow for a multiple mapping molecule, 50415235, within CDK2 dataset

These two examples are representative of the approaches that resolve many cases, however, there are two instances when neither approach is sufficient to differentiate between the potential mappings. The first is when a molecule has multiple topological distance maps that match to maps in the aggregated set that are equivalent and also most frequent. The second is when none of the topological distance maps for the molecule are present in the single mapping molecules. For these instances, further action needs to be taken and examples of each are shown below.

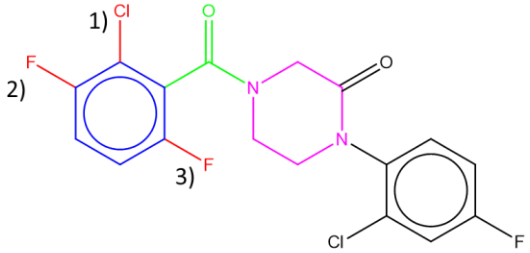
Reduced Graph Core



Single Mappings

Topological Distance Map	Number of Single Mapping Occurrences	Examples
{1-2: 1, 1-3: 3, 1-4: 4, 2-3: 1, 2-4: 2, 3-4: 1}	10	
{1-2: 1, 1-3: 5, 1-4: 6, 2-3: 1, 2-4: 2, 3-4: 1}	1	

Multi Mapping Molecule

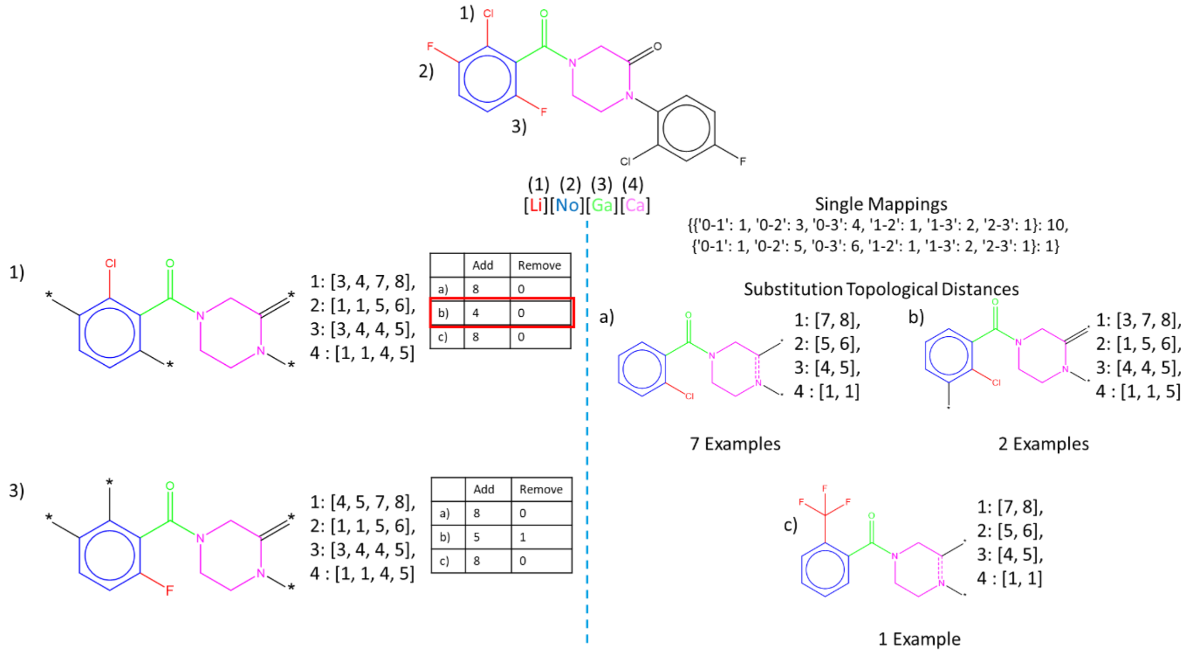


Potential Mappings
1) {1-2: 1, 1-3: 3, 1-4: 4, 2-3: 1, 2-4: 2, 3-4: 1} -> 10
2) {1-2: 1, 1-3: 4, 1-4: 5, 2-3: 1, 2-4: 2, 3-4: 1} -> 0
3) {1-2: 1, 1-3: 3, 1-4: 4, 2-3: 1, 2-4: 2, 3-4: 1} -> 10

1) Or 3) to be selected

Figure 4-12: Molecule CHEMBL2218143 within P2x7 Subset dataset with multiple mappings with the same number of examples

Figure 4-12 shows molecule CHEMBL2218143 which is in the P2x7 Subset and the P2x7 dataset and is represented by the RG core [Li][No][Ga][Ca]. There are three potential mappings for the molecule to the RG core due to the presence of the three halogens on the ring (labelled 1), 2) and 3)), each of which matches to the first Li node of the RG core (starred). The three topological distance maps for the molecule are shown. The topological distance maps for 1) and 3) are identical and occur 10 times in the aggregated set, whereas the map for 2) is not present. These two potential mappings are further analysed through the topological substitution edit distance. Figure 4-13 shows the molecule CHEMBL2218143 in full where it can be seen that there are two further substituents on the ring. Each of the potential node mappings gives rise to a different substituent topological map as shown on the left. Similarly, each of the molecules in the unique mappings to the RG core may have a different substituent pattern and therefore substituent topological map, and examples of these are shown on the right along with the number of molecules that have each substituent topological map. Each of the substituent maps for the molecule is compared against each of the aggregated substituent maps, shown on the right hand side, using the edit distance approach, and the mapping is chosen with the smallest edit distance to one of the aggregated maps. In the example shown this is mapping 1) which has edit distance 4 to the aggregated map b).



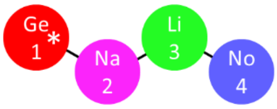
	Add	Remove
a)	8	0
b)	4	0
c)	8	0

	Add	Remove
a)	8	0
b)	5	1
c)	8	0

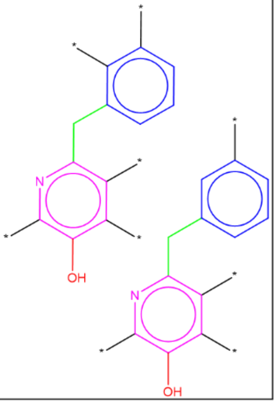
Figure 4-13: Molecule CHEMBL2218143 within P2x7 Subset dataset topological substitution edit distance calculations

Figure 4-14 is molecule CHEMBL3091612 in the P2x7 dataset which is represented by the RG core [Ge][Na][Li][No]. There are two potential mappings for the first node, Ge, of the molecule. Within the figure, the node has been starred and the different substructural fragments have been labelled. Neither of the potential mappings map to the single mapping topological distance maps. The two potential mappings are then compared to the single mapping topological distance maps (only one in this instance). Mapping 1 is selected as the smallest number of changes would be required to make it match to the single mapping example.

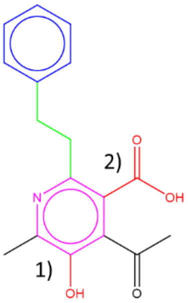
Reduced Graph Core



Single Mappings

Topological Distance Map	Number of Single Mapping Occurrences	Examples
{1-2: 1, 1-3: 5, 1-4: 6, 2-3: 1, 2-4: 2, 3-4: 1}	6	

Multi Mapping Molecule



Potential Mappings
1) {1-2: 1, 1-3: 5, 1-4: 7, 2-3: 1, 2-4: 3, 3-4: 1} -> 0
2) {1-2: 1, 1-3: 3, 1-4: 5, 2-3: 1, 2-4: 3, 3-4: 1} -> 0

1)	Single Topological Distance Map	Comparison
{1-2: 1,	{1-2: 1,	{1-2: 0,
1-3: 5,	1-3: 5,	1-3: 0,
1-4: 7,	1-4: 6,	1-4: 1,
2-3: 1,	2-3: 1,	2-3: 0,
2-4: 3,	2-4: 2,	2-4: 1,
3-4: 1}	3-4: 1}	3-4: 0}
		Total = 2

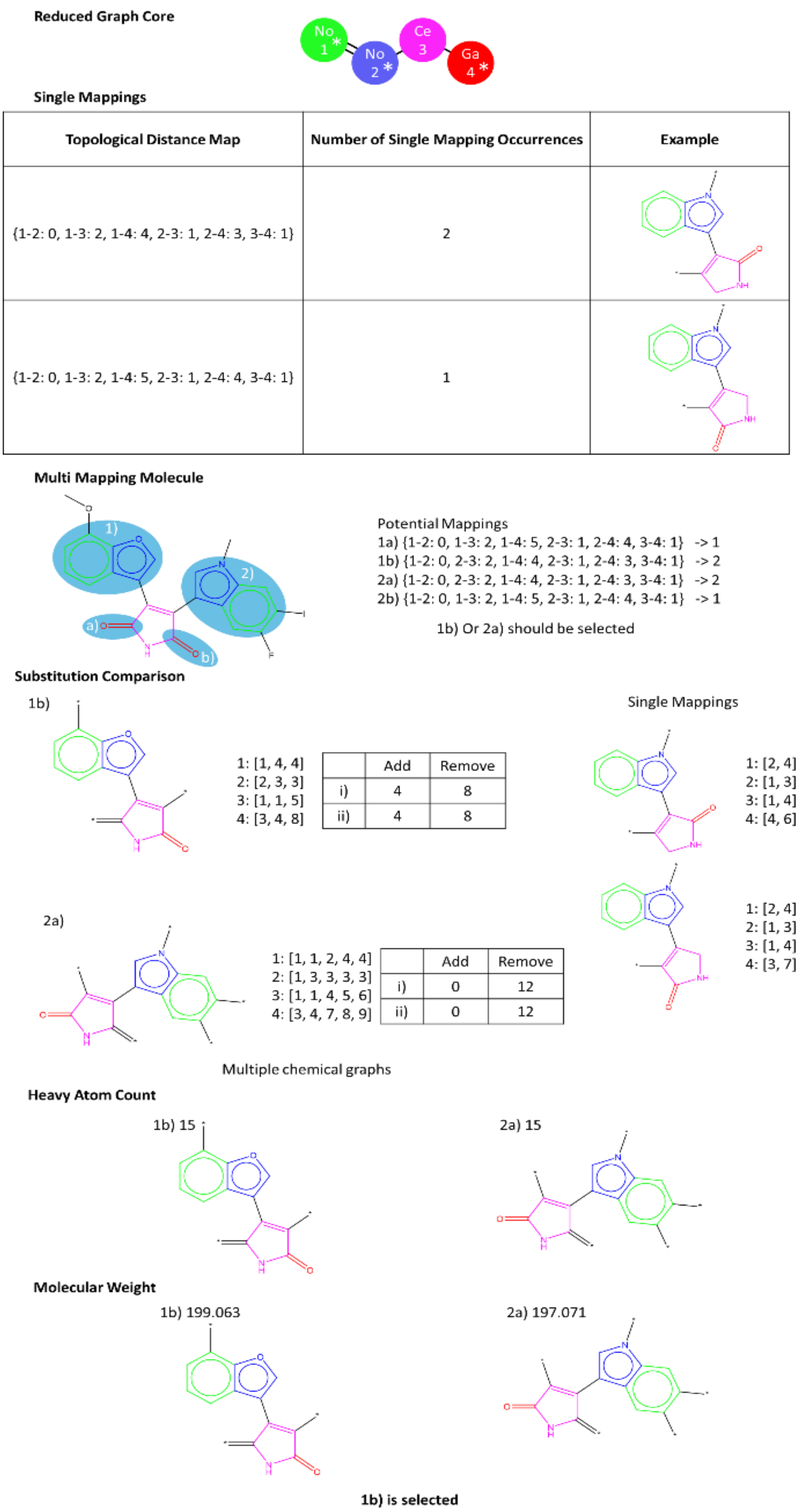
2)	Single Topological Distance Map	Comparison
{1-2: 1,	{1-2: 1,	{1-2: 0,
1-3: 3,	1-3: 5,	1-3: 2,
1-4: 5,	1-4: 6,	1-4: 1,
2-3: 1,	2-3: 1,	2-3: 0,
2-4: 3,	2-4: 2,	2-4: 1,
3-4: 1}	3-4: 1}	3-4: 0}
		Total = 4

1) Selected

Figure 4-14: Molecule CHEMBL3091612 within P2x7 dataset, solved by comparing potential topological distance maps to existing topological distance maps

Figure 4-15 shows molecule 50457130 within the CDK2 dataset which maps to the RG core [No]=[No][Ce][Ga]. This illustrates the molecular weight step in the workflow. There are three nodes that have several different variants, the first, No, second, No, and fourth, Ga, nodes. There are only two possibilities for the fused ring, the first two nodes, and have been labelled 1) or 2). The fourth node has been labelled a) or b). There are then four possible mappings of the RG core onto the molecule. These are made up of a combination of the substructures 1) or 2) and a) or b). The topological distance maps are calculated for each. 1b) and 2a) match with the most frequent map in the aggregated set. When examining the substitution patterns for these two mappings, both have 12 minimum changes using the edit distance approach, therefore, both are taken forward for further analysis. The next step is to compare the number of heavy atoms for the underlying structures. Both 1b) and 2a) have 15 heavy atoms so the molecular weight is checked next. 1b) is selected as it has a larger molecular weight of 199.063, compared to 2a)'s molecular weight of 197.071.

Figure 4-16 shows an RG core [Li][Na][No][Li] for the P2x7 dataset together with the set of topological distance maps for molecules with a single mapping to the core and some example molecules for each map. Molecule CHEMBL21028, shown in Figure 4-17, has two possible mappings to the core as both of the Cl substructures labelled 1) and 2) can map to the Li node labelled 4 in the RG core. Figure 4-17 demonstrates the steps taken within the workflow to resolve the RG core mapping for molecule CHEMBL21028. Both of the potential mappings match to the same number of existing examples (the final two rows in the table in Figure 4-16), therefore, they cannot be distinguished at this first step. When comparing the substitution patterns, both potential mappings have edit distances of four, the heavy atom counts and molecular weights of the underlying chemical graphs are also the same, 13 and 194.036 respectively. A tMCS is calculated for the chemical graph of every molecule which has a unique mapping to the RG core, and there are 25 of these. The largest tMCS is shown for each potential mapping, and the second option is selected as it has the highest tMCS overall. If these two were the same, the average tMCS would be found across the 25 examples in order to achieve the maximum overlap, however, this step is not necessary in this case.



Single Mappings

Topological Distance Map	Number of Single Mapping Occurrences	Example
{1-2: 0, 1-3: 2, 1-4: 4, 2-3: 1, 2-4: 3, 3-4: 1}	2	
{1-2: 0, 1-3: 2, 1-4: 5, 2-3: 1, 2-4: 4, 3-4: 1}	1	

Potential Mappings
1a) {1-2: 0, 1-3: 2, 1-4: 5, 2-3: 1, 2-4: 4, 3-4: 1} -> 1
1b) {1-2: 0, 2-3: 2, 1-4: 4, 2-3: 1, 2-4: 3, 3-4: 1} -> 2
2a) {1-2: 0, 2-3: 2, 1-4: 4, 2-3: 1, 2-4: 3, 3-4: 1} -> 2
2b) {1-2: 0, 1-3: 2, 1-4: 5, 2-3: 1, 2-4: 4, 3-4: 1} -> 1

1b) Or 2a) should be selected

1b)

1: [1, 4, 4]
2: [2, 3, 3]
3: [1, 1, 5]
4: [3, 4, 8]

	Add	Remove
i)	4	8
ii)	4	8

Single Mappings

1: [2, 4]
2: [1, 3]
3: [1, 4]
4: [4, 6]

1: [2, 4]
2: [1, 3]
3: [1, 4]
4: [3, 7]

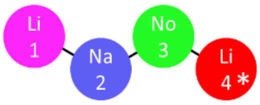
2a)

1: [1, 1, 2, 4, 4]
2: [1, 3, 3, 3, 3]
3: [1, 1, 4, 5, 6]
4: [3, 4, 7, 8, 9]

	Add	Remove
i)	0	12
ii)	0	12

Figure 4-15: Molecule 50457130 within CDK2 dataset with multiple mappings with the same number of examples that is resolved using molecular weight

Reduced Graph Core



Single Mappings

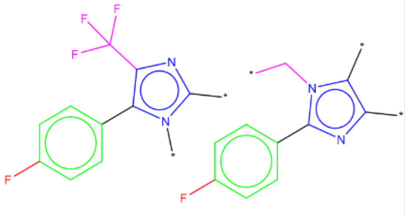
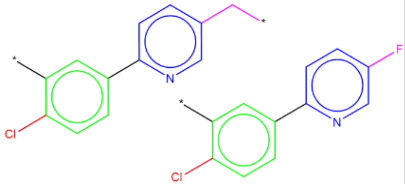
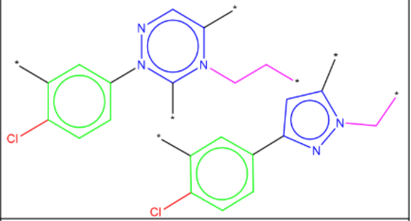
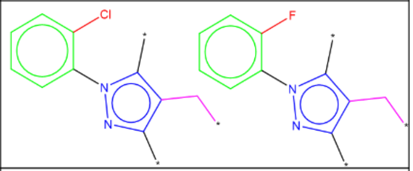
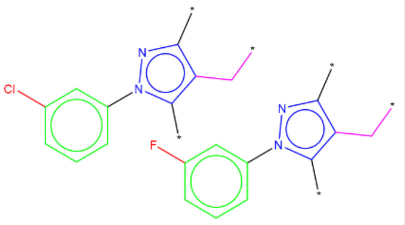
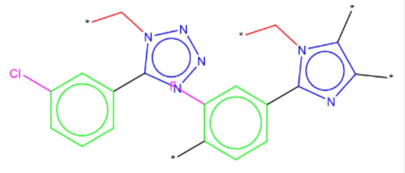
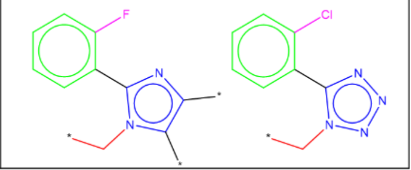
Topological Distance Map	Number of Single Mapping Occurrences	Examples
{1-2: 1, 1-3: 3, 1-4: 7, 2-3: 1, 2-4: 5, 3-4: 1}	25	
{1-2: 1, 1-3: 5, 1-4: 9, 2-3: 1, 2-4: 5, 3-4: 1}	13	
{1-2: 1, 1-3: 4, 1-4: 8, 2-3: 1, 2-4: 5, 3-4: 1}	7	
{1-2: 1, 1-3: 4, 1-4: 6, 2-3: 1, 2-4: 3, 3-4: 1}	2	
{1-2: 1, 1-3: 4, 1-4: 7, 2-3: 1, 2-4: 4, 3-4: 1}	2	
{1-2: 1, 1-3: 3, 1-4: 6, 2-3: 1, 2-4: 4, 3-4: 1}	2	
{1-2: 1, 1-3: 3, 1-4: 5, 2-3: 1, 2-4: 3, 3-4: 1}	2	

Figure 4-16: Molecule CHEMBL210284 within P2x7 dataset initial single mapping examples

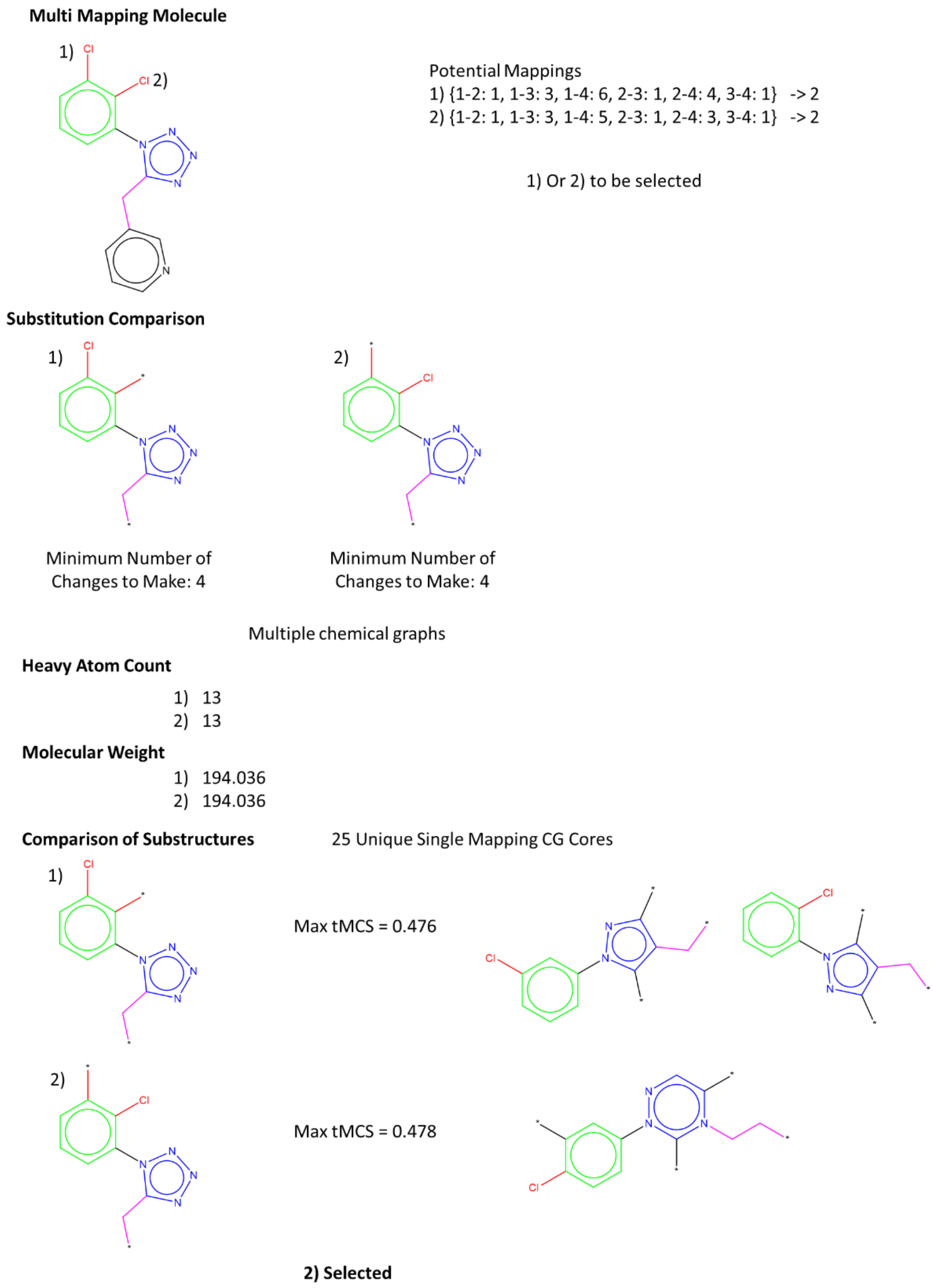


Figure 4-17: Molecule CHEMBL210284 within P2x7 dataset resolution steps of the RG core mapping

Figure 4-18 illustrates how two molecules CHEMBL1371125 and CHEMBL31184 within the Neurokinin dataset can be mapped to the RG core [No]=[No]1=[No]=[No]=1 when there are no existing single mapping examples. There are three possible mappings to the core for CHEMBL1371125 and two possible mappings for CHEMBL31184, respectively. All five topological distance maps are found which leads to three unique maps with two having two examples. The potential maps are then examined for each molecule to see which maps to the topological distance map with the largest number of examples. As there are two that have two examples this step only rules out option b) for molecule CHEMBL1371125. The chemical graphs of each of the potential maps are then examined. These are different within the molecule so the number of heavy atoms is calculated, which allows the two options to be resolved. In both instances mapping a) is selected as this contains one more heavy atom at a count of 17.

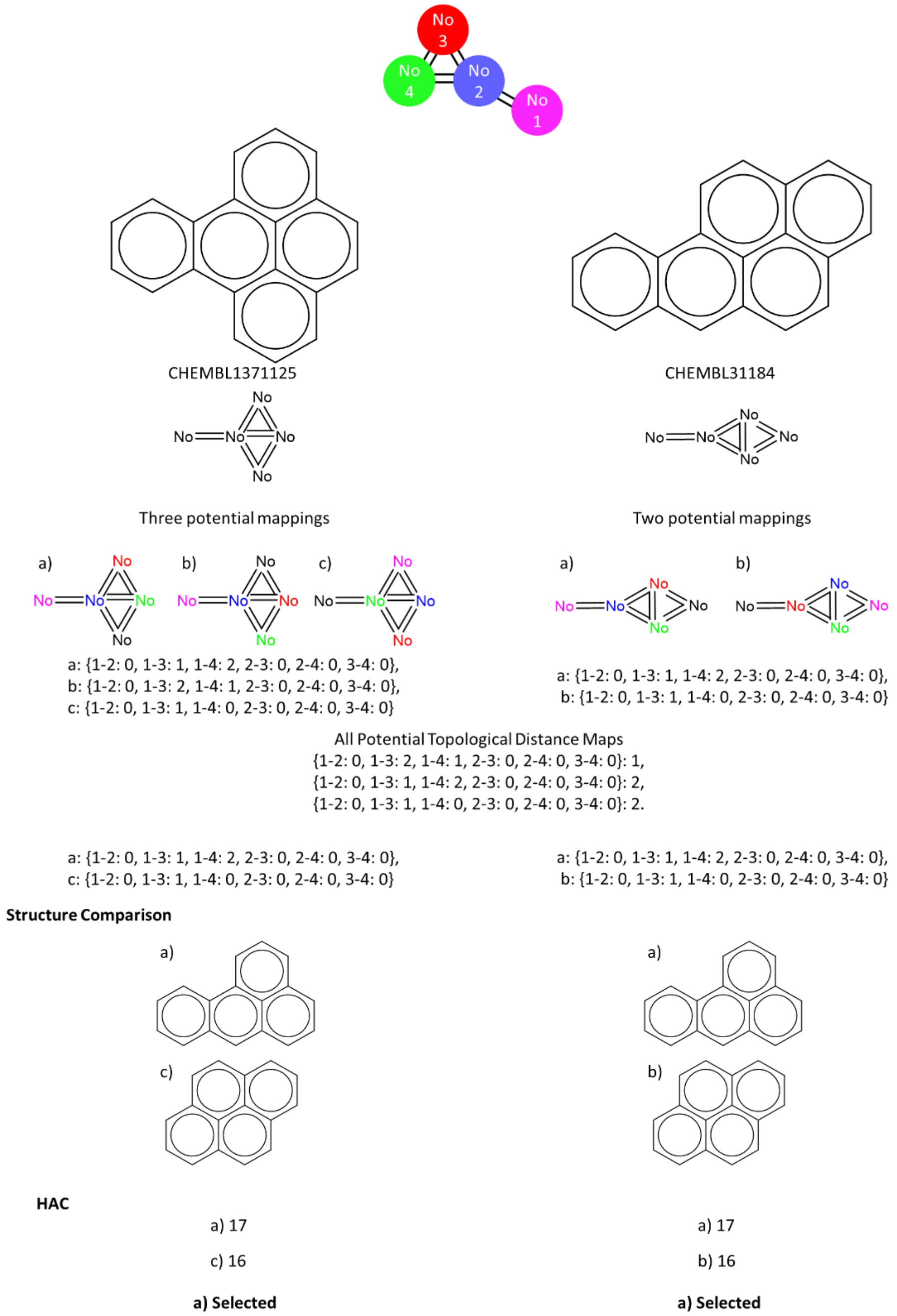

No 3
No 4
No 2
No 1
CHEMBL1371125
CHEMBL31184
Three potential mappings
Two potential mappings
a)
b)
c)
a: {1-2: 0, 1-3: 1, 1-4: 2, 2-3: 0, 2-4: 0, 3-4: 0},
b: {1-2: 0, 1-3: 2, 1-4: 1, 2-3: 0, 2-4: 0, 3-4: 0},
c: {1-2: 0, 1-3: 1, 1-4: 0, 2-3: 0, 2-4: 0, 3-4: 0}
a: {1-2: 0, 1-3: 1, 1-4: 2, 2-3: 0, 2-4: 0, 3-4: 0},
b: {1-2: 0, 1-3: 1, 1-4: 0, 2-3: 0, 2-4: 0, 3-4: 0}
All Potential Topological Distance Maps
{1-2: 0, 1-3: 2, 1-4: 1, 2-3: 0, 2-4: 0, 3-4: 0}: 1,
{1-2: 0, 1-3: 1, 1-4: 2, 2-3: 0, 2-4: 0, 3-4: 0}: 2,
{1-2: 0, 1-3: 1, 1-4: 0, 2-3: 0, 2-4: 0, 3-4: 0}: 2.
a: {1-2: 0, 1-3: 1, 1-4: 2, 2-3: 0, 2-4: 0, 3-4: 0},
c: {1-2: 0, 1-3: 1, 1-4: 0, 2-3: 0, 2-4: 0, 3-4: 0}
a: {1-2: 0, 1-3: 1, 1-4: 2, 2-3: 0, 2-4: 0, 3-4: 0},
b: {1-2: 0, 1-3: 1, 1-4: 0, 2-3: 0, 2-4: 0, 3-4: 0}
Structure Comparison
HAC
a) 17
c) 16
a) Selected
a) 17
b) 16
a) Selected

Figure 4-18: Example from Neurokinin dataset of how to resolve RG core mapping issue when there are no initial single mappings

4.4 Results of Mapping Process

Table 4-4 demonstrates how many molecules are resolved at each stage of the workflow. The majority of the multiple matches are resolved using the topological distance maps with the more complex steps required only for a relatively small number of molecules, with the exception of the Cyto and Neurokinin dataset. This indicates that the RG cores probably do not work effectively enough for these two datasets as they cannot easily distinguish the relationships in the dataset, also indicated by small RG cores.

Table 4-4: Table showing the number of molecules that are resolved within each step of the workflow

Resolved By…		**Dataset**									
		Bajorath	**CDK2**	**Chk1**	**Cyto**	**FactorXa**	**MMP12**	**Neurokinin**	**P2x7**	**P2x7 Subset**	**p38α**
Molecules with multiple mappings		377	355	2	15192	668	600	9316	961	290	1814
Existing Match to Topological Node Map	Single Match to Topological Distance Map	123	67	0	586	140	600	119	74	149	201
	Separated by the number of Topological Node Distance Map previously seen	11	200	2	2144	307	0	692	434	107	1050
	Topological substitution edit distance	46	45	0	1175	114	0	671	113	1	176
	CG	6	9	0	4189	21	0	3625	173	33	273
	HAC	13	0	0	4301	17	0	2382	30	0	30
	MW	2	7	0	1482	0	0	558	28	0	38
	Largest tMCS	0	2	0	132	31	0	83	9	0	9
	Highest Average tMCS	0	0	0	198	15	0	25	39	0	3
	Second CG	0	0	0	705	0	0	1092	1	0	0
	Arbitrary	0	0	0	0	0	0	0	0	0	2

No Existing Match to Topological Node Map	Min Edit Distance Topological Node Distance Map	62	6	0	22	19	0	3	5	0	1
	Avg Low Edit Distance Topological Node Distance Map	58	5	0	47	2	0	9	0	0	1
	Topological substitution edit distance	4	2	0	96	0	0	15	40	0	2
	CG	43	7	0	35	1	0	18	6	0	27
	HAC	9	2	0	3	1	0	5	2	0	0
	MW	0	0	0	2	0	0	0	1	0	1
	Largest tMCS	0	0	0	2	0	0	1	3	0	0
	Highest Average tMCS	0	0	0	2	0	0	1	0	0	0
	Second CG	0	0	0	0	0	0	3	0	0	0
	Arbitrary	0	0	0	0	0	0	5	3	0	0
No Existing Single Mappings	Max Overlap	0	0	0	0	0	0	0	0	0	0
	CG	0	0	0	0	0	0	0	0	0	0
	HAC	0	0	0	0	0	0	2	0	0	0
	MW	0	0	0	0	0	0	0	0	0	0

4.5 Conclusion

Some molecules can be mapped onto an RG core multiple times; a prioritisation process has been implemented to select one mapping that will be used for the further analysis. The prioritisation involves first examining the mappings of molecules for which there is a unique mapping and then exploring which of the multiple mappings aligns best with the existing examples. This process uses a topological distance map and associated workflow that allows the different mappings to the RG core to be distinguished. These mappings are then be utilised in the RG core visualisation.

5 Using Reduced Graphs to Visualise Lead Optimisation Series

5.1 Introduction

The previous chapter described how RG cores can be mapped onto a RG core to allow metadata to be extracted. This chapter aims to illustrate how these RG cores can be visualised, the final step of the workflow Figure 5-1.

An RG core consists of connected nodes that are common to multiple molecules within the dataset. The nodes contain the substructural information collected from the molecules represented by the RG core. The visualisation should provide an understanding of the relationship between molecules and be easily interpretable. Also, the visualisation should demonstrate the chemical space that has been investigated. The level of exploration is described by examining the variety of substructural groups each node represents.

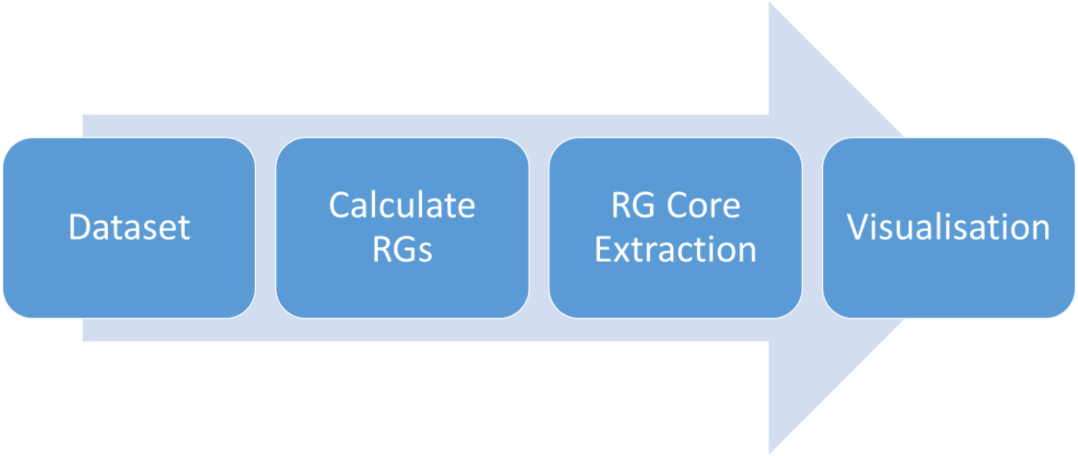


Figure 5-1: Optimised workflow:

5.2 Generation of the Visualisation

Data visualisation is important as it provides a way to communicate the relationships between data. These visualisations can then indicate patterns within the data that might have been missed or overlooked. Visualisations also provide a way of representing a large amount of information in an effective way. Therefore, a visualisation that brings together the molecules within a dataset and indicates areas of chemical space that these molecules cover, can allow chemists to identify relationships between the different structures and their properties.

The visualisation tool is designed to be interactive to allow a chemist to view the data at different levels of detail, such as the RG core level or RG node level and by the various substructures represented by the RG nodes. By moving between the different levels, the relationships and areas in chemical space that have been over- or under-explored should be revealed. The visualisation extends and complements R-group SAR tables and Markush structures. The RG cores are the most central part of the visualisation as they aim to summarise all the molecules and focus on the relationships between them all. An RG core should also demonstrate potential binding capabilities with the target.

Once the molecules represented by an RG core have been mapped to it, Chapter 4, the chemical fragments for each molecule are associated with the corresponding RG nodes of the core. These fragments become known as substructural group derivatives. Figure 5-2 shows an example of the output from this step for a dataset. As mentioned in Chapter 2, the reduced graph nodes of the individual molecules are annotated by SMARTS string representations of the underlying substructures. These substructures are collated for all molecules represented by a core and displayed as tables associated with each node. The substructures are associated with counts indicating the number of times they occur, i.e., the number of molecules that contain the substructure at that position. As demonstrated by the node No in Figure 5-2, this method differentiates between substructural group derivatives with different substitution patterns.

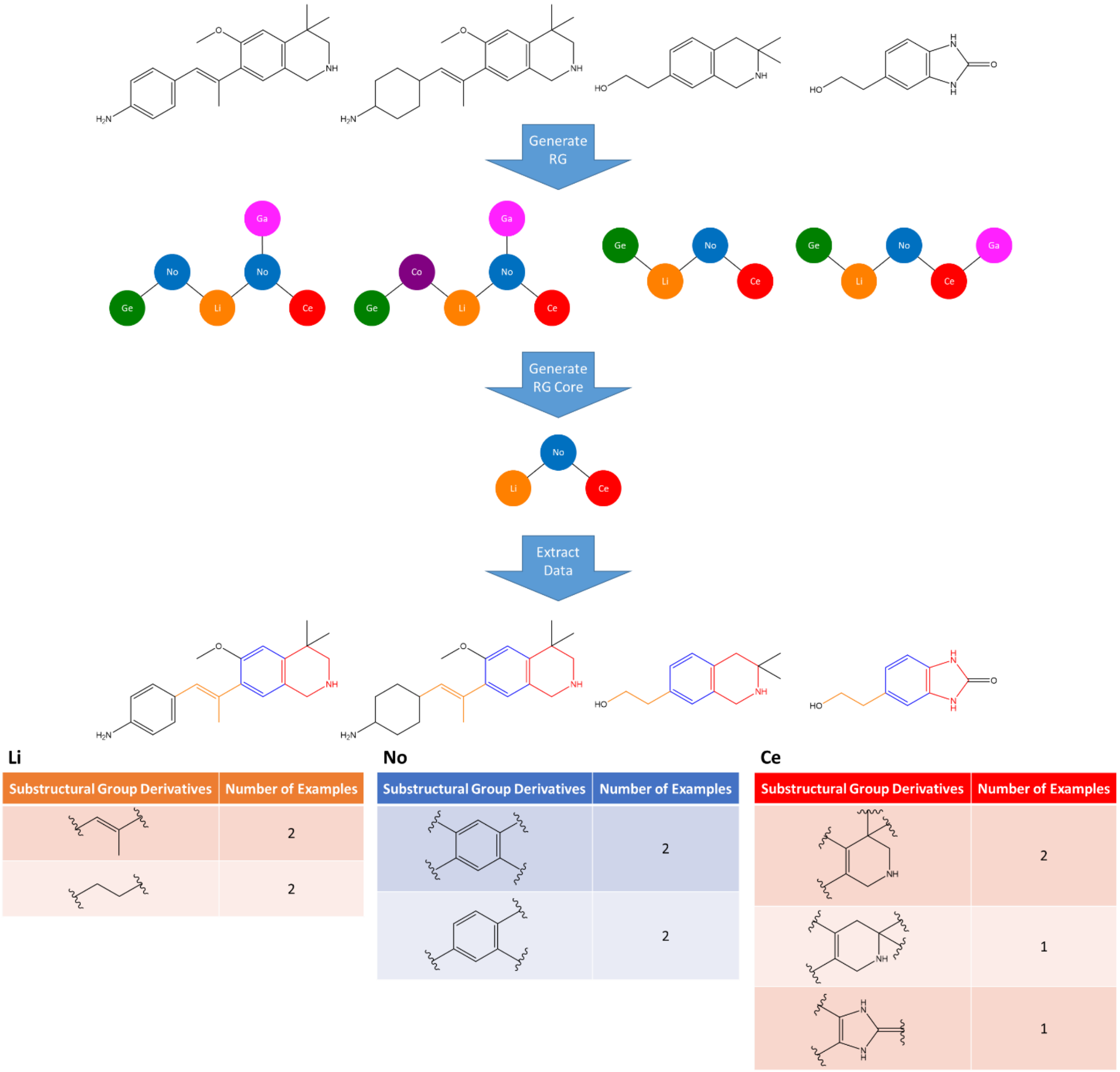
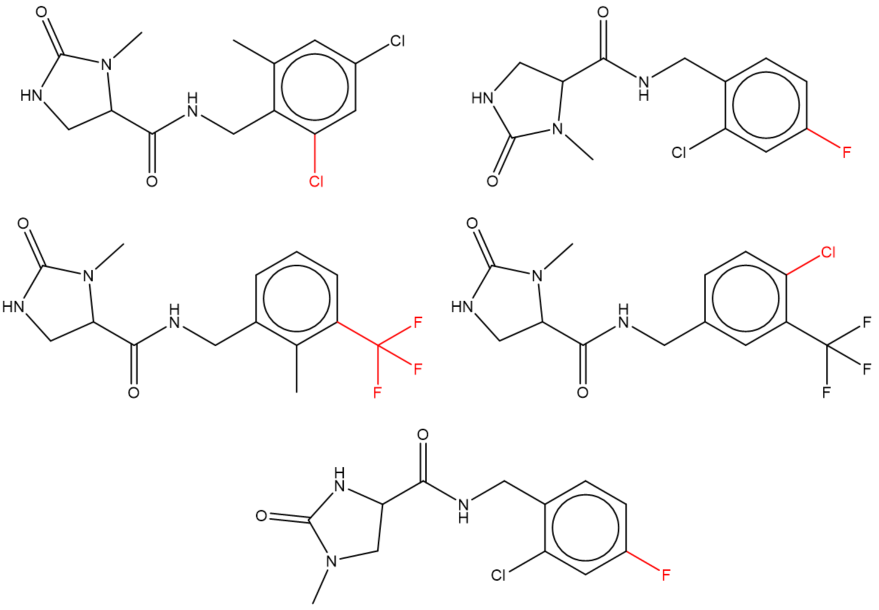


Figure 5-2: Representation of the substructure derivatives for a set of molecules that map to an RG core.

Figure 5-3 illustrates another example based on five molecules that all match the same RG core. Even though there are five molecules, there are only three unique substructures that are represented by the red node and the count for each substructure is displayed in the table.



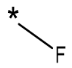
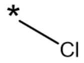
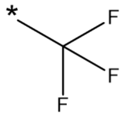
Node Structure	Count
	2
	2
	1

Figure 5-3: The substructure derivative table for the node shown in red

Each node of an RG core is displayed as a pie chart that shows the number of different substructures associated with the node. The overall size of the pie chart represents the number of unique substructures represented by the node. The segments within the pie chart represent the different substructures, with the segment size proportional to the frequency of the substructure. Figure 5-4 illustrates the node pie chart for the molecules in Figure 5-3. Using pie charts makes it easy for the user to see which substructures have been used frequently and which have been used rarely. For example, Figure 5-4, shows three substructural group derivatives have been used, where the trifluoro group (grey) has been explored less than the chloro (orange) and fluoro (blue) groups. The pie charts therefore indicate the level of exploration of a node and by examining the core as a whole, a user can identify chemical space that has been under- and over-explored. In Figure 5-5, nodes 1Na, 2Ca, and 3Ga are smaller pie charts as they have only one or two substructural group derivatives, respectively. Node 0Na is considerably bigger as it has 17 different substructural group derivatives. The large blue segment in node 0Na indicates that one of the substituents has been seen the most at 15 times, and the olive coloured segment is smaller as it has only been seen once.

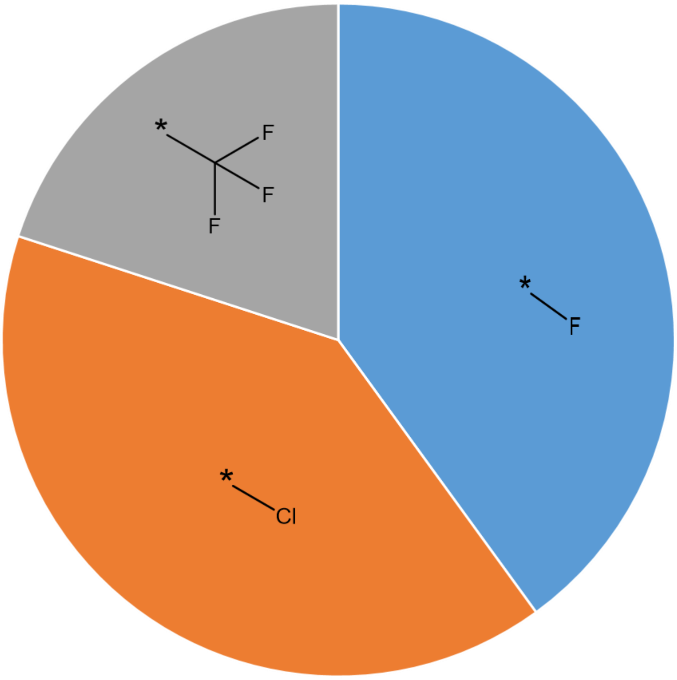


Figure 5-4: An example of a node pie chart

The pie charts that represent the nodes within a core are connected together to form a visualisation of a set of related molecules. The way they are connected depends on the bond between the nodes within the core structure. A thin line is used for a single bond with a thicker line used for a double bond. An example can be seen in Figure 5-5, where the bond between node 1Na and 2Ca is a double bond and the rest of the bonds are single bonds.

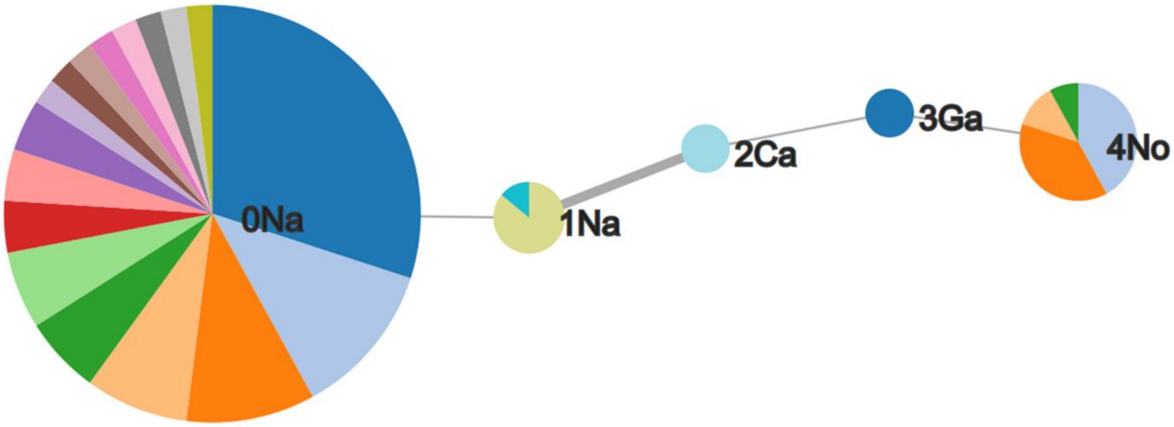


Figure 5-5: An example of a whole core within the visualisation tool

5.2.1 Implementation

The visualisation is an interactive tool that was implemented via a RESTful API that runs using Python 3, Flask, D3, JavaScript, HTML and CSS. The engineering of this interactive tool is highlighted in Figure

5-6. The interactive tool is a webpage that the user can easily interact with. These interactions cause a request to be posted to the flask server. If the user is looking at a dataset that has already been processed and exists within the visualisation, the data is extracted from the corresponding hierarchical data format (HDF) file. The HDF file allows several different datasets and the associated RG core information to be saved within the same file within separate sections so they can be accessed individually. The first page for each dataset is a principal component analysis (PCA) plot that is constructed in a D3 SVG object, where the data points are calculated using scikit-learn and M2FP descriptors. The RG core visualisation is also constructed in a D3 SVG object. The user can also import a new dataset as a set of SMILES strings with a corresponding IDs and pIC50 values. In this case, all the pre-processing steps are executed including creating the RGs, extracting the RG cores and then extracting the metadata to display within the RG visualisation.

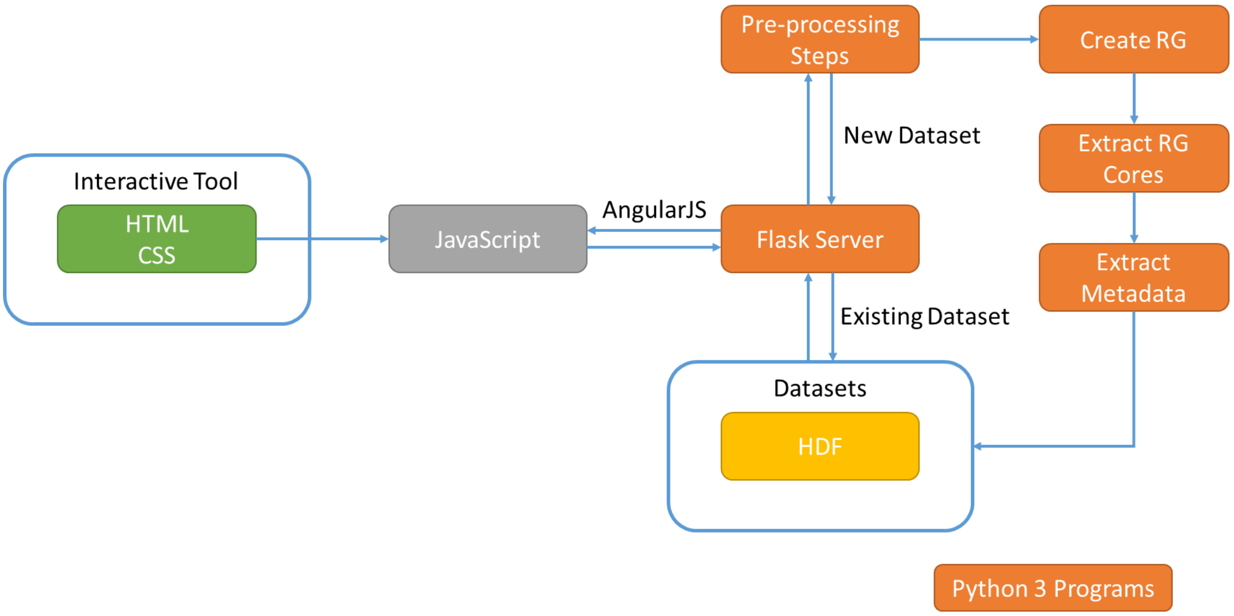


Figure 5-6: Engineering behind the visualisation tool

5.3 Results

The initial webpage of the interactive tool is depicted in Figure 5-7 where the blue boxes have been inserted to highlight some features. The interactive features allow different data to be hidden until needed so that the visualisation is clear, concise, and not an information overload. The user can select between existing datasets or exploring a new dataset that has not yet been run through the visualisation, by using the highlighted tabs. If the user wants to examine an existing dataset, the dataset is selected from the dropdown menu, the box labelled B in Figure 5-7. The boxes labelled C and D in Figure 5-7 show search bars where either individual RG cores or the molecule IDs can be searched. If the RG core or molecule ID is present within the current screen the corresponding data is highlighted, otherwise, if there are no matches, the user is alerted.

The following analysis was done for the P2x7 subset dataset when extracting the cores from the whole dataset, i.e., without any clustering.

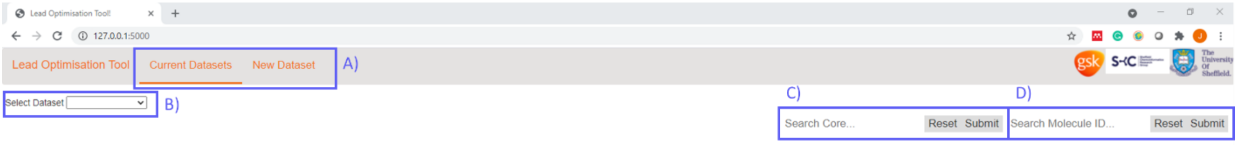


Figure 5-7: Initial view of the interactive tool. A) Tab to select between current datasets or new dataset. B) Dropdown menu that contains the current datasets. C) Core search bar. D) Molecular ID search bar.

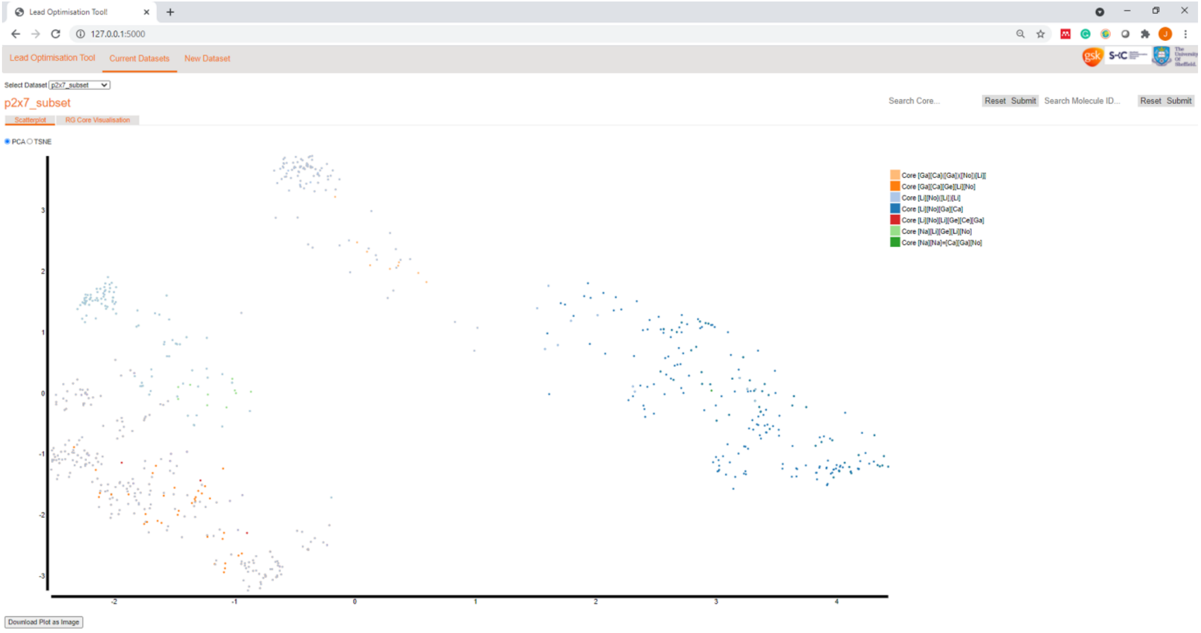
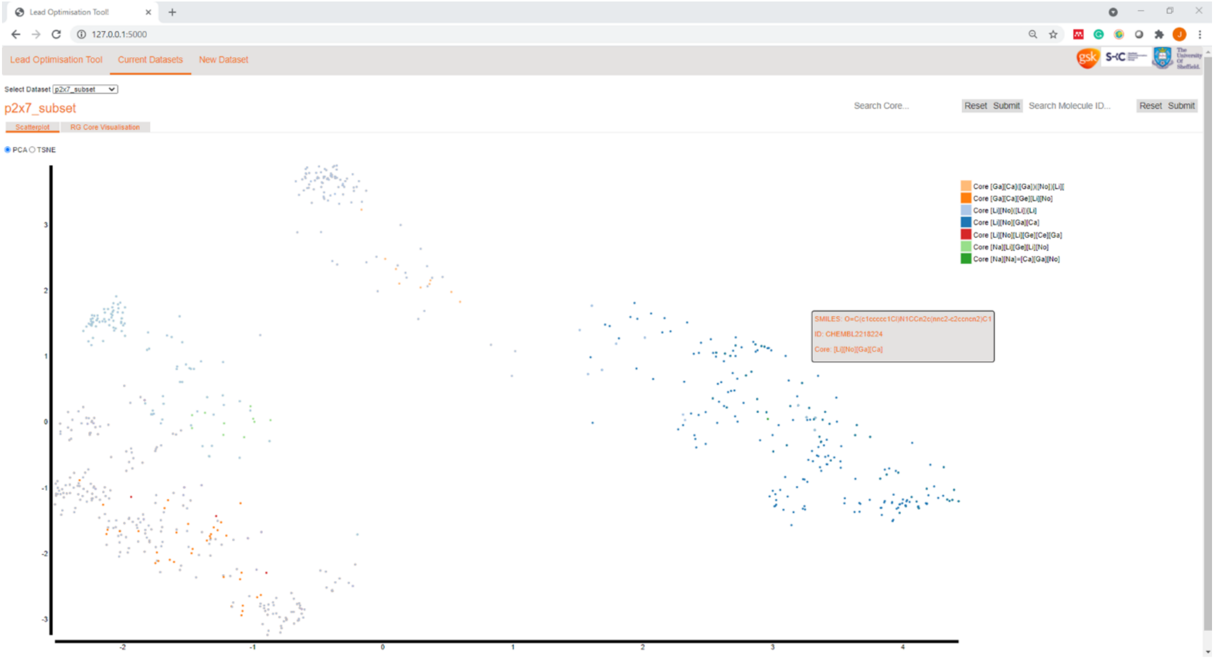


Figure 5-8: Visualisation screenshot of the PCA plot for P2x7 subset

The first step is for the user to select the dataset they would like to visualise. A PCA plot is then generated using M2FP descriptors calculated for the dataset, Figure 5-8. The display can also be switched to a t-distributed stochastic neighbour embedding (t-SNE) plot (Van Der Maaten & Hinton, 2008). The displayed plot can be saved as an image by clicking the '*Download Plot as Image*' button below the plot. Each data point is a molecule that can be coloured by the RG core that it belongs to. When the user hovers over a data point, they are presented with some information about that molecule, Figure 5-9. The reported metadata is in a grey hover box and provides details of the SMILES, ID and RG core for that data point. As mentioned previously, two search bars can be used to highlight molecules with either a specific RG core or molecule ID. An example of how this looks in the scatterplot is shown in Figure 5-10 where the molecules corresponding to RG core [Ga][Ca][Ge][Li][No] have been highlighted in red.



Lead Optimisation Tool!

127.0.0.1:5000

Select Dataset p2x7_subset

p2x7_subset

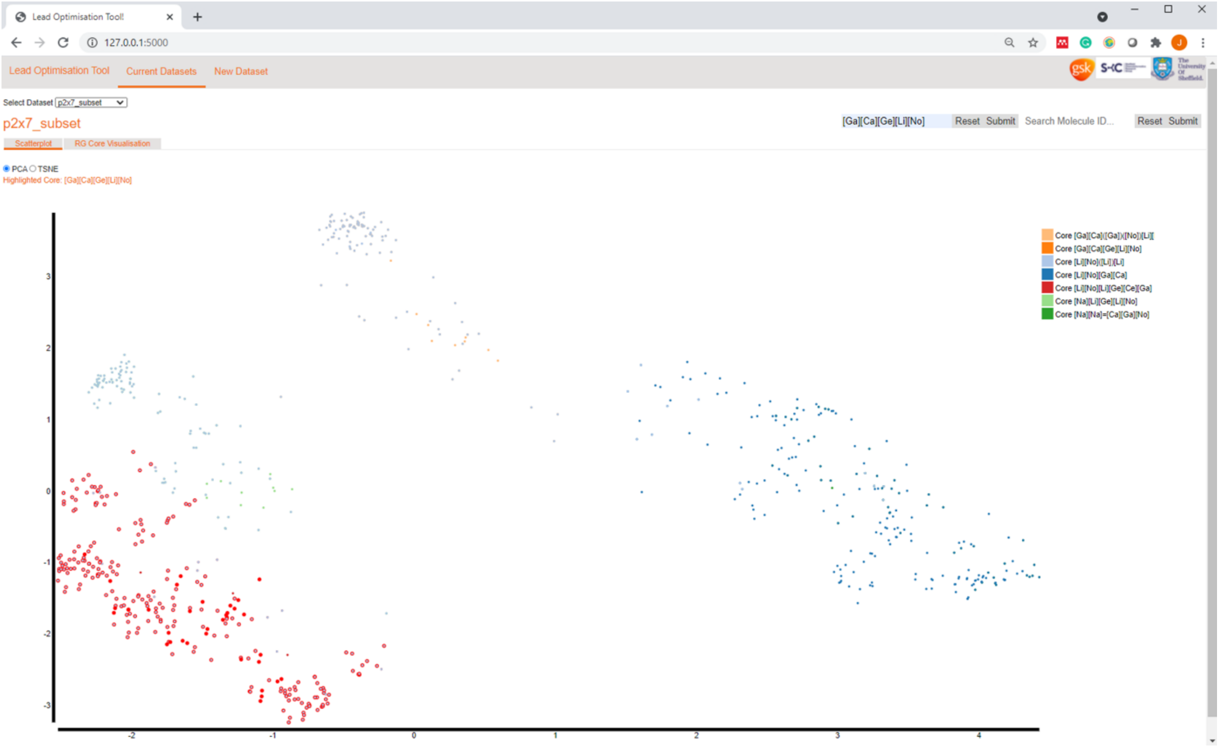
● PCA ○ TSNE

SMILES: O=C(c1ccccc1Cl)N1CCn2c(nnc2-c2ccncn2)C1

ID: CHEMBL2218224

Core: [Li][No][Ga][Ca]

Figure 5-9: Screenshot showing the hovered over information



Lead Optimisation Tool!

127.0.0.1:5000

Select Dataset p2x7_subset

p2x7_subset

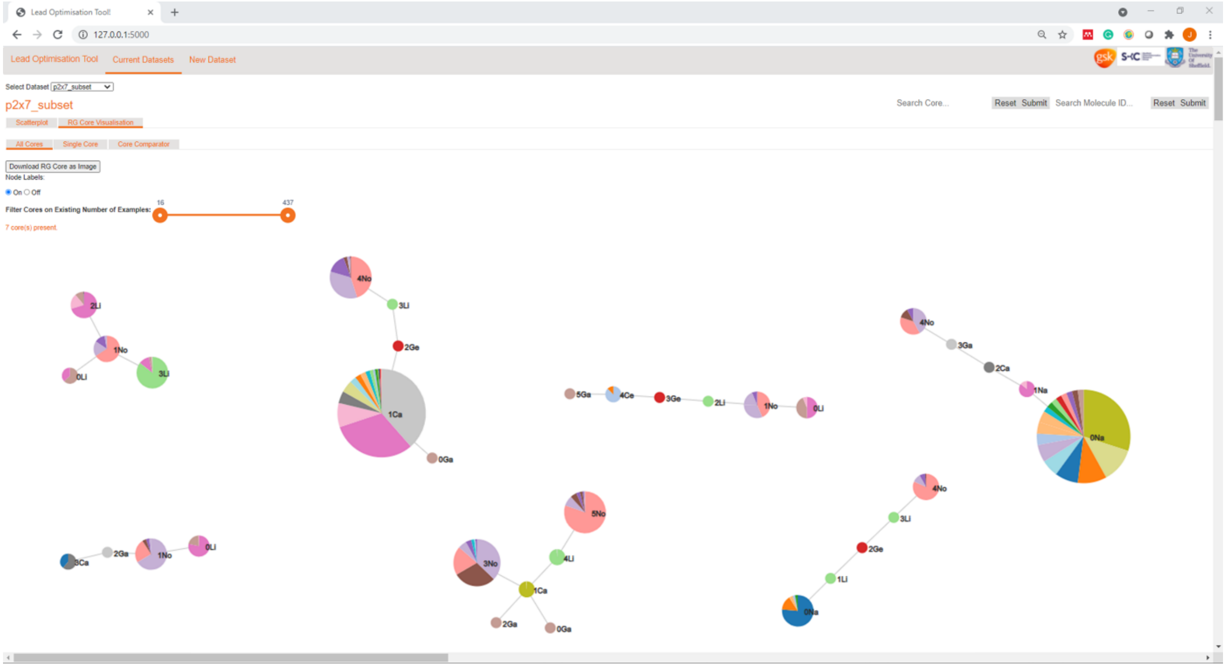
[Ga][Ca][Ge][Li][No]

● PCA ○ TSNE

Highlighted Core: [Ga][Ca][Ge][Li][No]

Figure 5-10: Highlighting RG core [Ga][Ca][Ge][Li][No]

To understand how an individual molecule relates to the dataset, the user can select the '*RG Core Visualisation*' tab or click on a specific data point to highlight the RG core related to that molecule. A new display is generated showing all of the RG cores for that dataset and if a data point is clicked, the corresponding core is highlighted in orange, Figure 5-11. The P2x7 subset dataset consists of 691 unique molecules that are represented by seven RG cores.



Select Dataset p2x7_subset

p2x7_subset

Node Labels:

● On ○ Off

Filter Cores on Existing Number of Examples: 16 437

7 core(s) present.

Figure 5-11: Visualisation screenshot of all cores within P2x7 subset, the top central RG is highlighted in orange

The RG cores are represented as graphs with the nodes displayed as pie charts. An image of the RG core representations can also be saved. An RG core from Figure 5-11 is captured in Figure 5-12. The number of molecules represented by this core is 249.

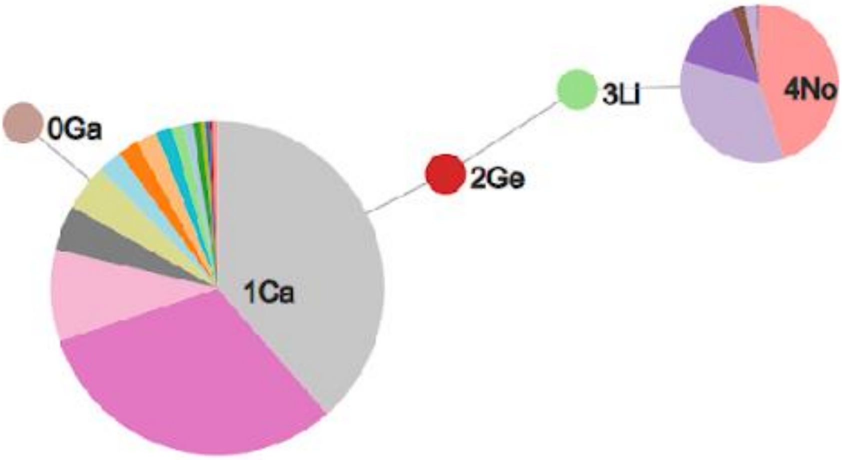
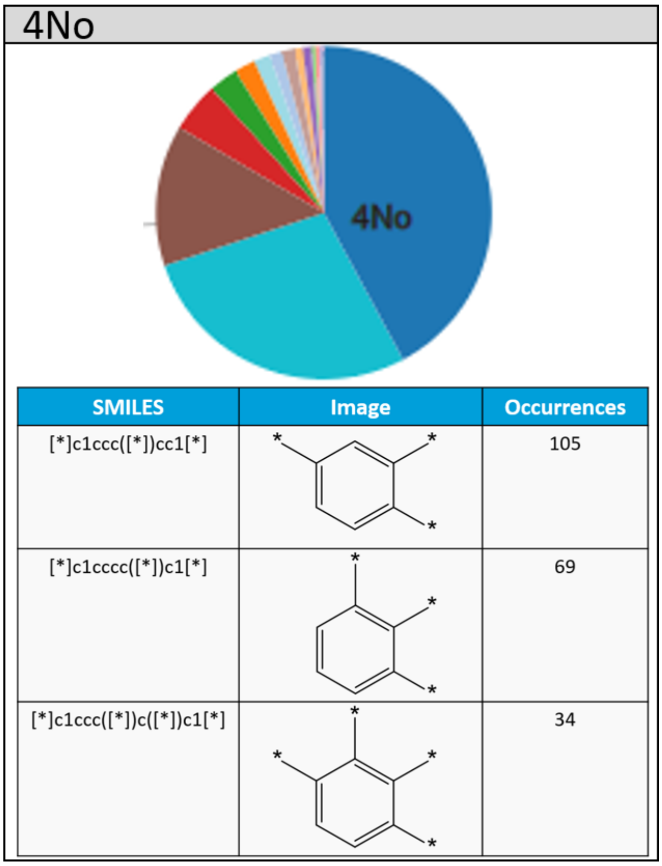


Figure 5-12: RG core [Ga][Ca][Ge][Li][No] from P2x7 subset is displayed in the visualisation

As the size of the pie charts indicates how many substructural fragments have been explored, it is easy to see that two nodes have had more fragments investigated. Three of the nodes, Ga, Ge and Li, only have one substructural group derivative and are therefore small pies charts of uniform colour. For the remaining two nodes, Ca and No, it can be seen that there are multiple substructural group derivatives. Node Ca is the largest because it has the highest number of different substructural group derivatives at 16, compared to No's seven and the rest of the nodes with just one. As discussed above, substructural group derivatives with different substitution patterns are distinguished as demonstrated in Figure 5-13 where the substructures are all benzene rings but with varying points of substitution. The pie charts can be selected and a pop up is displayed showing the pie chart and a table of the corresponding substructural group derivatives with the number of occurrences, as shown in Figure 5-13.



SMILES	Image	Occurrences
[*]c1ccc([*])cc1[*]		105
[*]c1cccc([*])c1[*]		69
[*]c1ccc([*])c([*])c1[*]		34

Figure 5-13: Pop up display of one of the core RG nodes

Within the main display window, underneath the node display, there is a table that represents each RG core. The columns represent the nodes in the RG core and each row represents a unique combination of the substructures that map to the nodes of the RG core. Therefore, a substructural group derivative could appear multiple times in a column, however, the combination of substructures across the nodes is unique. Figure 5-14 demonstrates an example of a single RG core within the table where it can be seen that for nodes 0Ga, 2Ge and 3Li there is no variation. However, for the other two nodes, there are variations, whether it is the substitution patterns or the atoms found within the rings. A combined column is also introduced to clearly demonstrate how the substructures are connected. The final column contains the number of molecules that have this combination as their chemical graph representative of that RG core. A screenshot of how the RG core table is displayed within the interactive visualisation is shown in Figure 5-15, which is an up scaled version of Figure 5-14 that shows the tables for all the RG cores within the dataset. A table can also be downloaded as an excel file to make it easier to convert this information into different formats or presentations.

[Ga][Ca][Ge][Li][No]

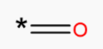
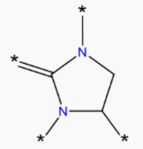
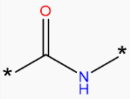
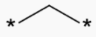
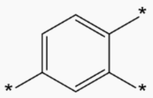
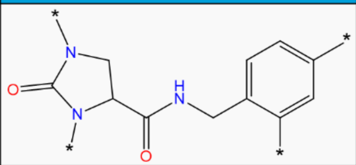
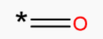
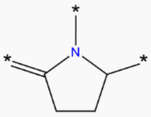
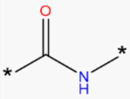
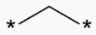
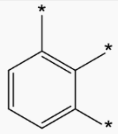
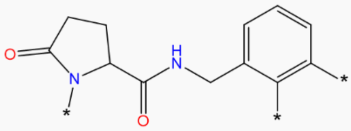
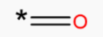
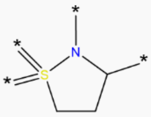
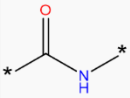
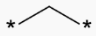
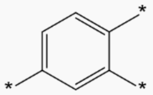
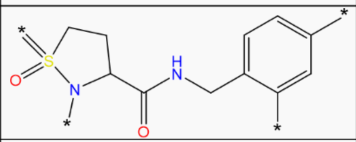
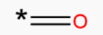
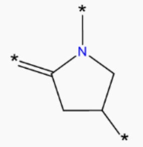
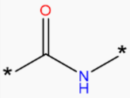
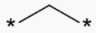
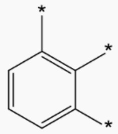
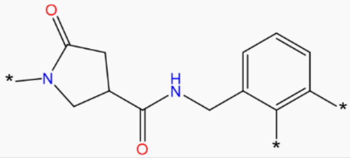
0Ga	1Ca	2Ge	3Li	4No	Combined	Number of Examples
						29
						19
						10
						3

Figure 5-14: A snippet of the large all cores table, Figure 5-15

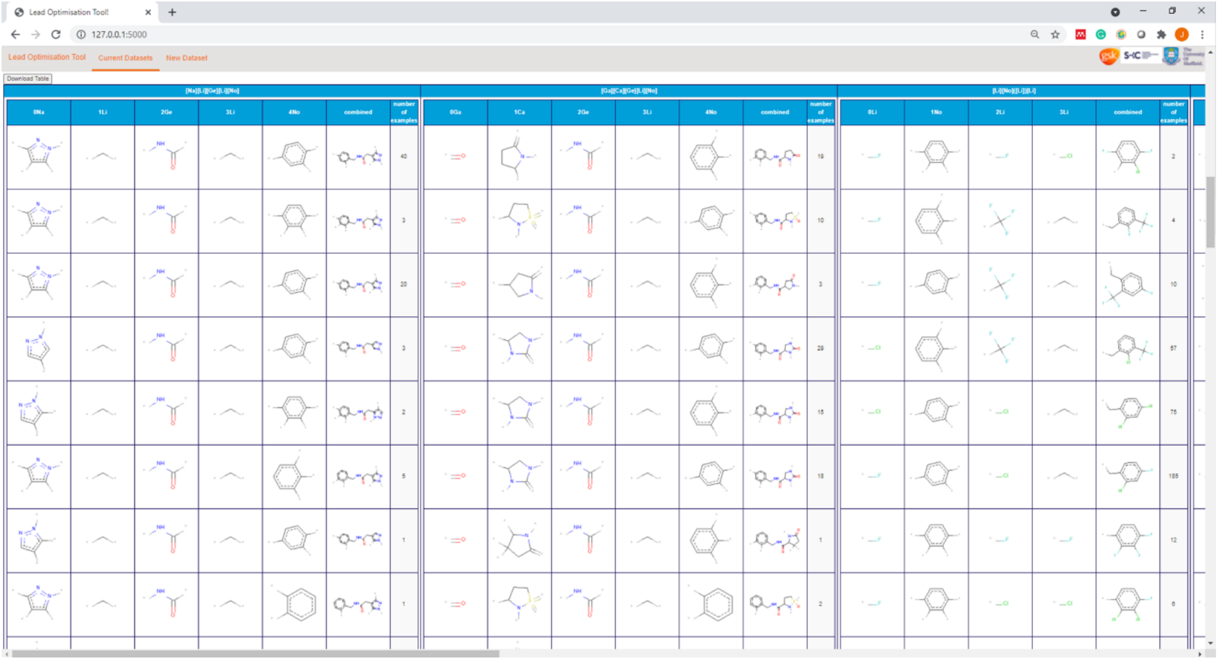
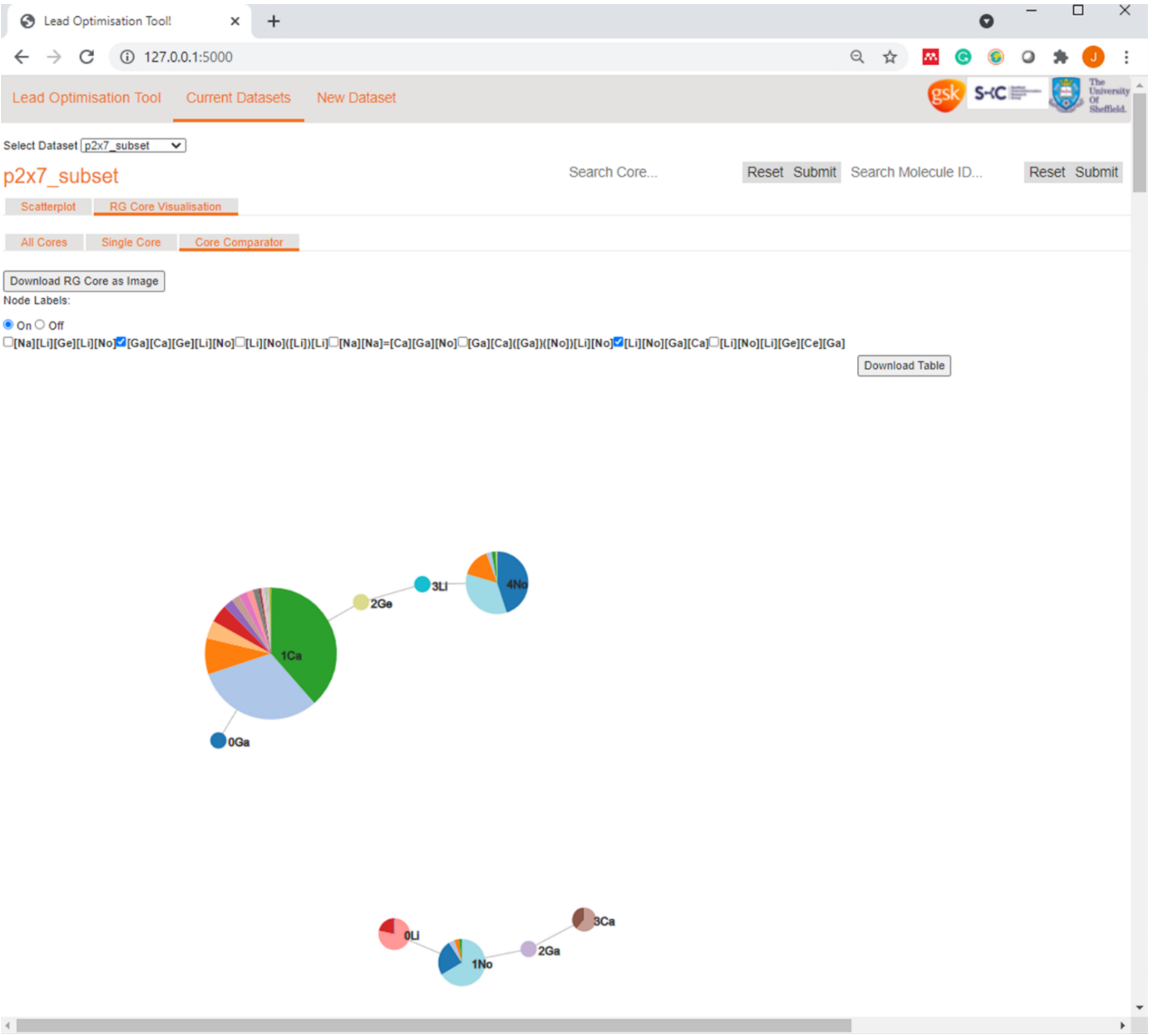


Figure 5-15: Screenshot demonstrating the RG core substituent table for all p2x7 subset cores

If a dataset has a large number of RG cores, the interactive visualisation can be overwhelming and it can be a little difficult to comprehend all the data initially. Therefore, strategies have been put in place to allow the data to be filtered. On the RG core visualisation '*All Cores*' tab, a slider filter option allows the shown RG cores to be filtered between the two limits. Additionally, to allow a less complicated display, a tab has been created that allows the user to select RG cores to compare. This tab has all the same functionality as before, however, it only displays the RG cores that the user has selected.



Select Dataset p2x7_subset

p2x7_subset

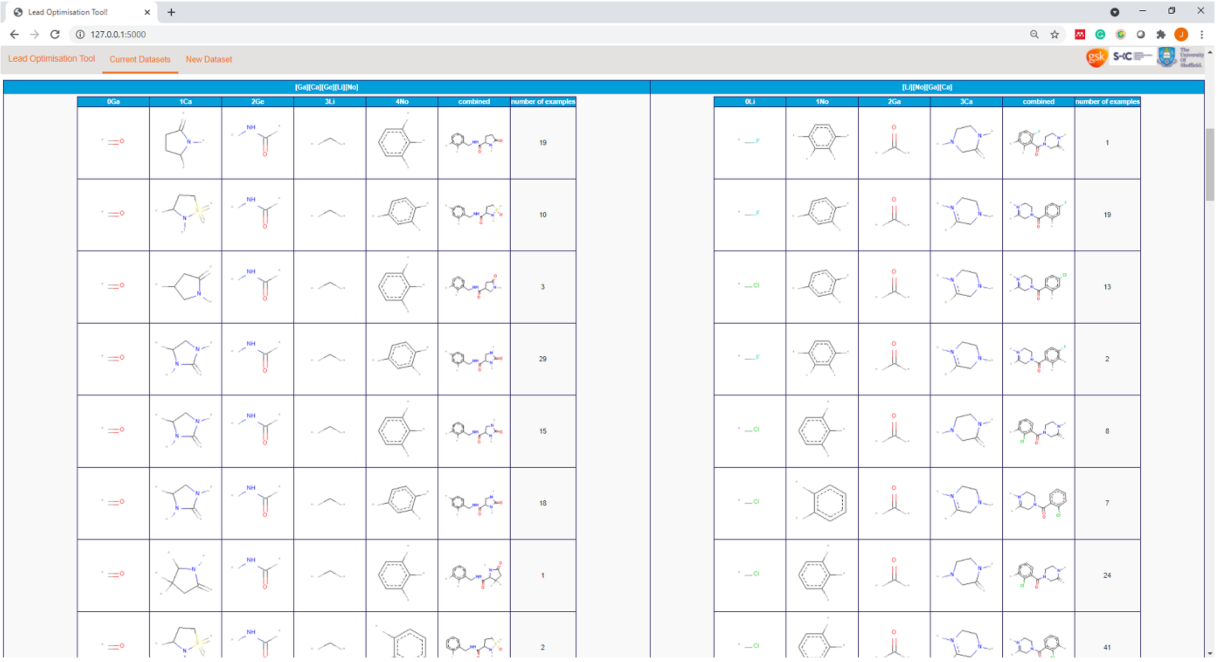
Node Labels:

● On ○ Off

□[Na][Li][Ge][Li][No]☑[Ga][Ca][Ge][Li][No]□[Li][No]([Li])[Li]□[Na][Na]=[Ca][Ga][No]□[Ga][Ca]([Ga])([No])[Li][No]☑[Li][No][Ga][Ca]□[Li][No][Li][Ge][Ce][Ga]

Figure 5-16: Core comparison screen within the visualisation

Like previously, underneath the core display, the substituent tables indicate the unique combinations of substructural group derivatives for each of the cores, Figure 5-17, and both tables have interactive features. Each of the RG cores can be viewed individually. There are two possible ways: clicking on the '*Single Core*' tab and choosing from the drop-down menu; or clicking on the header of the table of a core which takes the user to a page displaying only that core, Figure 5-18.



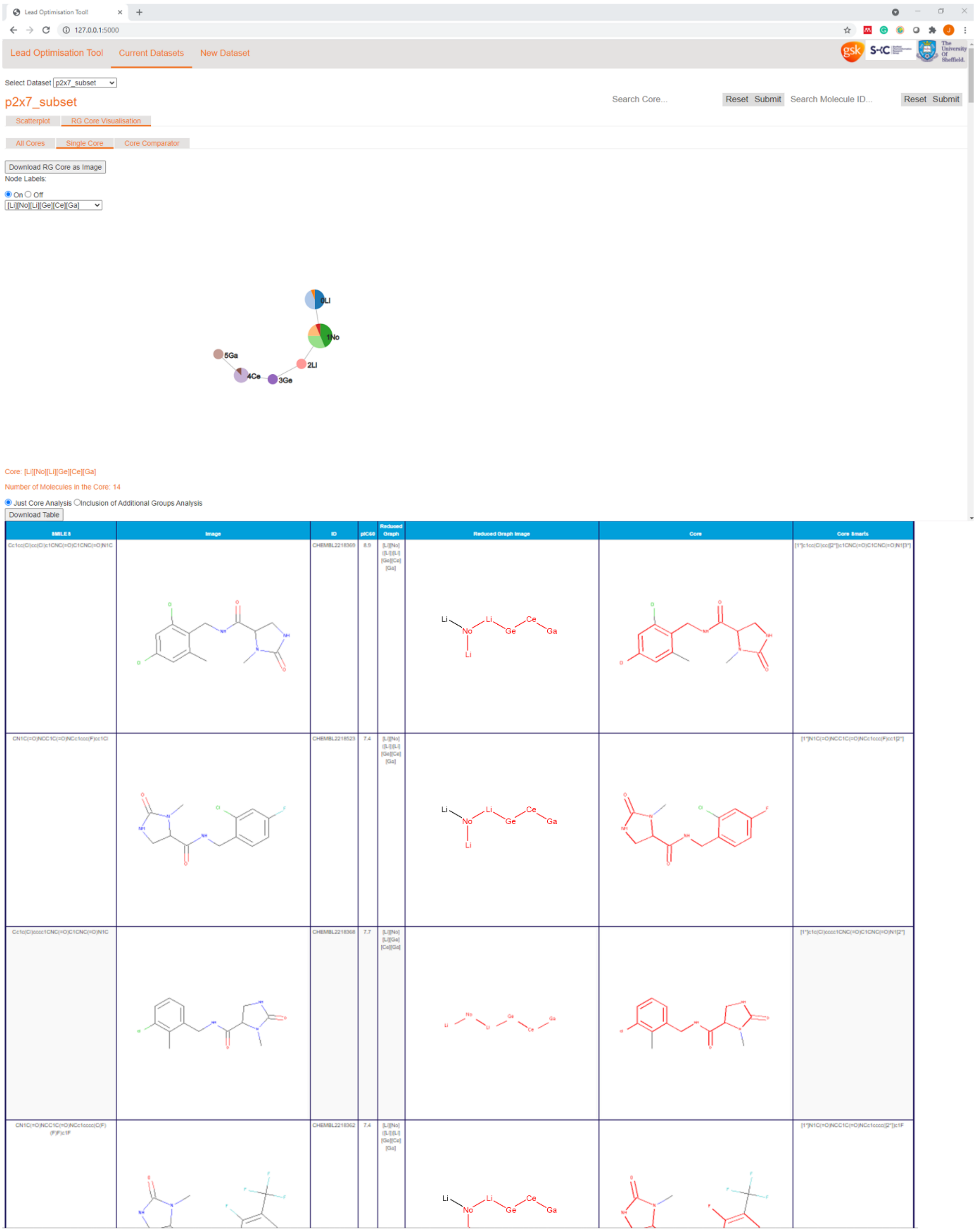
[Ga][Ca][Ge][Li][No]

0Ga	1Ca	2Ge	3Li	4No	combined	number of examples
						19
						10
						3
						29
						15
						18
						1
						2

[Li][No][Ga][Ca]

0Li	1No	2Ga	3Ca	combined	number of examples
					1
					19
					13
					2
					8
					7
					24
					41

Figure 5-17: Core breakdown within the core comparison



Lead Optimisation Tool | Current Datasets | New Dataset

Select Dataset p2x7_subset

p2x7_subset

Search Core... Reset Submit Search Molecule ID... Reset Submit

Scatterplot | RG Core Visualisation

All Cores | Single Core | Core Comparator

Download RG Core as Image

Node Labels:

● On ○ Off

[Li][No][Li][Ge][Ce][Ga]

Core: [Li][No][Li][Ge][Ce][Ga]

Number of Molecules in the Core: 14

● Just Core Analysis ○Inclusion of Additional Groups Analysis

Download Table

SMILES	Image	ID	pIC50	Reduced Graph	Reduced Graph Image	Core	Core Smarts
Cc1cc(Cl)cc(Cl)c1CNC(=O)C1CNC(=O)N1C		CHEMBL2218369	8.9	[Li][No]([Li])[Li][Ge][Ce][Ga]			[1*]c1cc(Cl)cc([2*])c1CNC(=O)C1CNC(=O)N1[3*]
CN1C(=O)NCC1C(=O)NCc1ccc(F)cc1Cl		CHEMBL2218523	7.4	[Li][No]([Li])[Li][Ge][Ce][Ga]			[1*]N1C(=O)NCC1C(=O)NCc1ccc(F)cc1[2*]
Cc1c(Cl)cccc1CNC(=O)C1CNC(=O)N1C		CHEMBL2218368	7.7	[Li][No][Li][Ge][Ce][Ga]			[1*]c1c(Cl)cccc1CNC(=O)C1CNC(=O)N1[2*]
CN1C(=O)NCC1C(=O)NCc1cccc(C(F)(F)F)c1F		CHEMBL2218362	7.4	[Li][No]([Li])[Li][Ge][Ce][Ga]			[1*]N1C(=O)NCC1C(=O)NCc1cccc([2*])c1F

Figure 5-18: Core analysis within the new visualisation, the highlighted atoms are part of the RG core

Finally, if the input dataset is organised according to lead optimisation rounds, that is, the molecules are labelled according to the LO iteration in which they were made, the progression of the data in each iteration can be analysed. The progress of each RG core is then observed. A table is provided that shows the RG cores that exist within each iteration along with the number of molecules that map to each RG core in each round of the lead optimisation process. A simplistic view of what is achieved is illustrated in Table 5-1, where ‘*RG Core 1*’ represents one of the RG cores and ‘*+n*’ is the number of molecules that have been added to the RG core in each round. Additionally, within the pop up display of the RG core node, Figure 5-13, new substructural fragments are indicated along with the numbers of each substructure that have been added in this iteration.

Table 5-1: Simplistic view of the RG core progression table when round data is introduced

Round 1		Round 2		Round 3	...	Round N
RG Core 1	+ n	RG Core 1	+ n	RG Core 1	...	RG Core 1
RG Core 2	+ n	RG Core 2	+ n	RG Core 2	...	RG Core 2
RG Core 3	+ n	RG Core 3	+ n	RG Core 3	...	RG Core 3
		RG Core 4	+ n	RG Core 4	...	RG Core 4
		RG Core 5	+ n	RG Core 5	...	RG Core 5
					...	RG Core 6

5.4 Conclusion

An interactive visualisation tool has been created to complement the chemists existing knowledge and to aid their decision-making process. It allows them a more profound understanding of the lead optimisation dataset that is being researched. Once the molecules have been mapped to the RG cores, the corresponding substructural node fragments are extracted from the datasets, and can be visualised to understand the chemical space that had been over- and under-explored. This is done through a combination of RG cores, a new graphical representation, and node tables. The substructural fragments for each of the nodes within an RG core were extracted for each molecule associated with the RG core.

The nodes of an RG core are represented as pie charts. The pie charts indicate the levels of chemical exploration at the node. When a pie chart is small and has a low number of segments then the chemical space in this region of the molecule has not been explored extensively, however, if a pie chart is large and has a high number of segments then the chemical space has been explored more extensively.

The work carried out in the rest of this thesis will further explore and exploit the data extracted from this visualisation and create molecules based on this information.

6 Applying Reduced Graphs For Molecular Exploration

6.1 Introduction

A lead optimisation (LO) project consists of an iterative process in which existing structure-activity relationship (SAR) knowledge is used to suggest modifications to an existing molecule. These new molecules are synthesised, tested and the SAR updated to repeat the design process. The functional groups are chosen based on existing knowledge, or by exploring areas of chemical space that have previously not been observed. In chemoinformatics, when existing knowledge is used to create a molecule, this concept is known as exploitation. Whereas, when previously unseen functional groups are incorporated into a new molecule, new chemical space is explored. Characteristically, when pharmaceutical companies design new molecules, exploration and exploitation tend to be balanced so that further knowledge can be discovered while existing knowledge of active chemical space is utilised.

In recent years, machine learning algorithms have been employed to suggest potential new molecules to synthesise. Some of these machine learning techniques are referred to as black-box algorithms. A machine learning method is called a black box when it learns from the input data and creates a function that is not transparent to the user. Therefore, although the algorithms can generate output molecules there is little understanding of why the molecules are being recommended. However, it is essential for the medicinal chemists to understand and interpret why the molecules are of interest and why a molecule should be prioritised to be synthesised over other molecules. If medicinal chemists can understand why a model is suggesting a molecule, they are more likely to trust and use it.

The visualisation created in the previous chapter allows medicinal chemists to see areas of chemical space that are over- and under-explored. It does this by showing the substructural group derivatives used for each node together with the number of molecules in the series that they occur in. It is hoped that the information gathered in the creation of the visualisation can be utilised to determine if particular substructural group derivatives are crucial for particular nodes or whether any substructural group is satisfactory as long as it has the same binding capabilities.

Even though the concepts of exploration and exploitation are essential features in the drug discovery process, none of the current techniques explicitly maps molecules onto the existing LO dataset and scores them on both the exploration and exploitation. The next chapter will investigate the creation of an exploitation score.

Molecules are categorised as “exploring” if they represent new areas of chemical space and can, therefore, generate new knowledge. Contrastingly, molecules are classified as exhibiting exploitation if they are comprised of functional groups that have been seen before and the functional groups are associated with good properties, albeit in different arrangements.

6.2 Methodology

Chapter 2 presented a representation that has been developed to align compounds in a LO project into a single object, referred to as a reduced graph (RG) core, or simply a core. Each node within a core is annotated according to the number of different substituents/ substructures that map to that node. An example RG core that represents 334 molecules is shown in Figure 6-1. The visual representation illustrates the different chemical substructures associated with each node, Figure 6-2. Each node contains the metadata that is related to that node for a particular LO dataset. The metadata consists of the different substructures along with the frequency of occurrence of each. A node’s metadata indicates the extent to which the chemical space at this position has been explored. For example, the blue node in the core in Figure 6-1 is shown in Figure 6-2. The node has four substructures which are distributed as follows: 165 of the 334 molecules have F at this node position; 158 are Cl; ten are CF_3, and one example is Br.

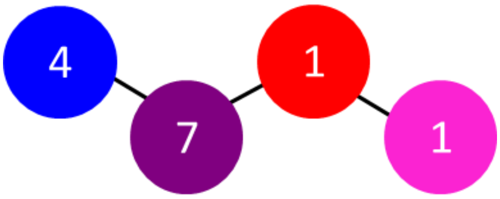


Figure 6-1: Reduced Graph Core. The number within the node is how many substructures that node contains

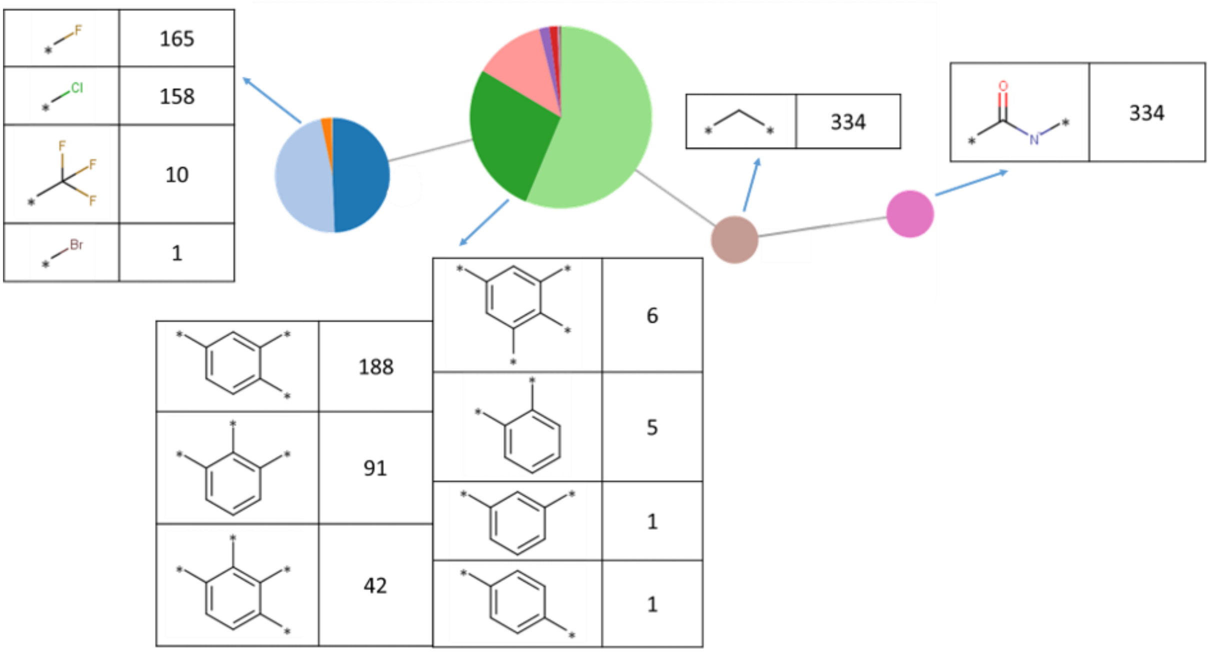


Figure 6-2: Visual representation of core and node breakdown

This chapter describes the development of an exploration score based on this representation. The exploration score will assess the extent to which a new molecule expands the chemical space currently explored. Therefore, the frequency of occurrence of each of the substructural fragments for each node is of importance.

6.2.1 Exploration Score

Given a new molecule, the aim is to map the new molecule onto the core and generate a score to reflect how much information would be added to the LO series. A molecule which adds the most information should have the highest exploration score. A score is calculated for each node that maps to a node in the core. Ideally the score will lie between zero and one to easily identify an exploration scale for a node, whether it has a high level of exploration or a low level of exploration. The node scores for all nodes that map to the core will be combined to generate an overall molecule score. The higher the score, the higher the level of exploration. This score is known as the exploration score since it should reflect the extent to which the new molecule explores new chemical space.

Consider a new molecule that maps to the grey segment of the node shown in Figure 6-3a) which represents three different substructures (shown by the grey, orange and blue segments) with different frequencies of occurrence. The exploration score should be highest if the molecules maps to the least frequent substructure shown by the grey segment, followed by orange, followed by blue. The exploration score should also be able to distinguish between nodes with different distributions of substructures. For example, the node in Figure 6-3b) also represents three different substructures but the examples are distributed differently across the three substructures. The exploration score for mapping to the grey segment in b) should be higher than that for the grey segment in a) since the distribution across the substructures in b) is more skewed and the molecule maps to a substructure with relatively low occurrence. If the molecule maps to the blue segment then the exploration score for a) should be higher. If the molecule maps to the orange segment then b) should be higher.

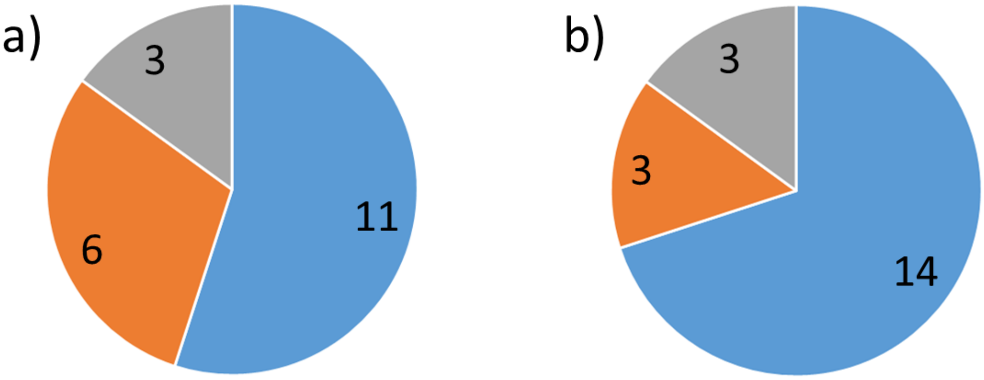


Figure 6-3: Two separate nodes. The exploration score should distinguish between adding a new molecule to the two different 3 grey substructures

The exploration score should take account of both the number of substructural fragments within a node and the distribution of examples across the substructures. Consider the example in Figure 6-4, which shows two nodes, one which represents five substructure fragments whereas the other represents just two. A new molecule which does not map to either node segments would have a higher score for node b). Also, if a new molecule maps to both the orange segments then node b) will score higher. An example where node a) would score a higher exploration score than node b) would be if a new molecule would map to the yellow segment in node a) and blue segment in node b).

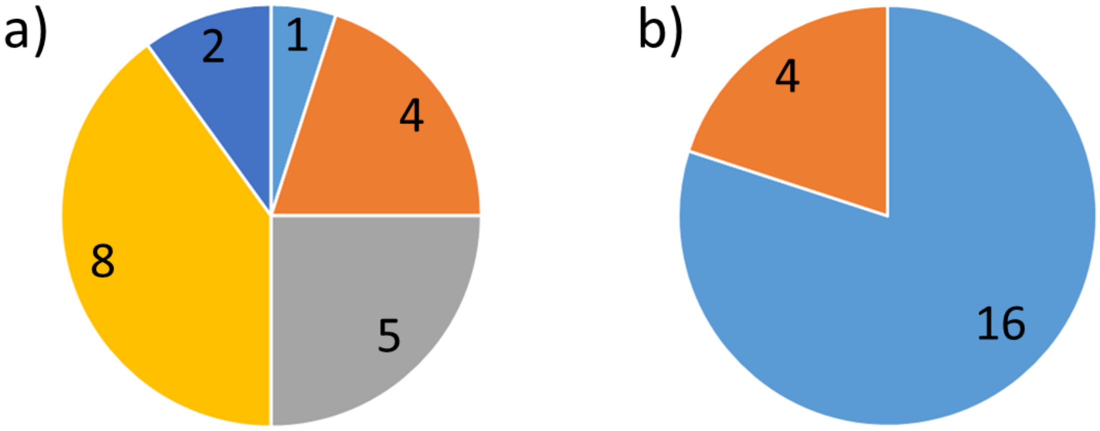


Figure 6-4: Example of two nodes with different number of substructural fragments but overall represent the same number of molecules

6.2.1.1 Hypothetical Examples

6.2.1.1.1 Node Scores

Some hypothetical examples were constructed to determine requirements for the exploration score. The examples represent nodes derived from theoretical LO series and are described in Table 6-1. There are a total of seven different nodes (A, B, C, D, E, F, G) each with a different underlying distribution of substructures. In all cases, the nodes represent 20 molecules. The second column shows the underlying distribution of substructures for each node as a set of frequency bins. Thus, Node A consists of a single substructure which is common to all 20 molecules. Node B consists of two substructures with one present in 19 molecules and the other present in only one molecule. The third column represents the corresponding part of the prior distribution that the new molecule would map to. If the new molecule has a substructure that is already present in the core, the bin number indicates which bin it maps to; if it is a new substructure that is not present in the node from the core this is shown as “New subst”. Thus, the first row in the table represents a new molecule mapped to node A where the molecule contains the same substructure that is represented by all 20 molecules in node A. The second row represents a new molecule which adds a new substructure to node A, etc.

Table 6-1: Hypothetical node examples

Row No.: Node No	Prior Distribution	New Molecule
1:A	(20) i.e. single substituent	Bin 1
2:A	(20) i.e. single substituent	New subst
3:B	(19,1)	Bin 1
4:B	(19,1)	Bin 2
5:B	(19,1)	New subst
6:C	(18,1,1)	Bin 1
7:C	(18,1,1)	Bin 2

8:C	(18,1,1)	New subst
9:D	(11,9)	Bin 1
10:D	(11,9)	Bin 2
11:D	(11,9)	New subst
12:E	(11,6,3)	Bin 3
13:F	(14,3,3)	Bin 3
14:G	(7,6,4,3)	Bin 4

Considering the two extreme cases first. The largest score should be row 2 and the lowest score should be row 1. There is only one existing substructure represented in node A and row 2 adds a new substructure, i.e., new information, whereas row 1 represents another example of the substructure seen in all existing molecules.

The next highest scoring molecules should be those that add new substructures that have not been seen before; rows 5, 8 and 11. Ideally the exploration score should differentiate between the different prior distributions in these nodes. Rows 5 and 8 are highly skewed distributions indicating that most molecules in the series have the same substructure at this node. The addition of a new substructure should therefore score highly in terms of exploration. This is most extreme for row 5 which should therefore have the highest score. Row 11 has a more even distribution. Adding a new substructure is beneficial in terms of exploration but the prior distribution is less extreme than for either row 5 or 8 and therefore ideally the exploration score would be less in this case.

The next level of molecules is represented by rows 4 and 7. Both these rows add a substructure that has only a single example in the core. Ideally row 4 would score slightly higher as there are only two substructures in the prior distribution compared to three.

The next three rows are 12, 13, 14. All three of these rows add to a substructure that currently has three examples, however, the prior distributions of substructures differ. The highest of the three rows should be row 13 as the prior distribution is the most skewed and the new substructure smooths this. Row 14 is the most even distribution and the new substructure does not, therefore, change this much. Row 12 is intermediate.

The next two rows, 9 and 10, are both from the same distribution, but row 10 adds to the substructure with a lower number of examples and therefore leads to a smoothing of the distribution compared to row 9 which increases the skewness. Therefore, row 10 should have a higher exploration score.

The final rows 3 and 6 both add to the substructure with the highest number of examples within their distributions. The final state for 3 is more skewed than the final state for 6, therefore, 6 should score higher.

Therefore, the ideal ordering is: 2 > 5 > 8 > 11 > 4 > 7 > 13 > 12 > 14 > 10 > 9 > 6 > 3 > 1.

Therefore, the requirements for the exploration score are: that the values should range between zero and one where one is the maximum information that can be added and should be assigned when a new substructure is being added; different scores are given for the same number of substructures but different distributions, like the orange segments in Figure 6-4; and the addition of a substructure that smooths a more skewed distribution should score higher than a less skewed one.

6.2.1.1.2 Molecule Scores

The node exploration scores should be combined to generate an overall molecular exploration score. A simple scenario has been constructed to identify the best way of combining the scores. This consists of a hypothetical core and various new molecules that map onto the core in different ways. The hypothetical core consists of three nodes and all three nodes, by definition, must have the same overall number of examples. The three nodes that comprise the hypothetical core are nodes, E, F and G, from Table 6-1. Their node distributions can be seen in Figure 6-5 (where the nodes are represented as histograms rather than pie charts). Three different methods are used to combine the node scores to molecule scores: summing the node scores; multiplying the node scores; and calculating the mean value of the node scores.

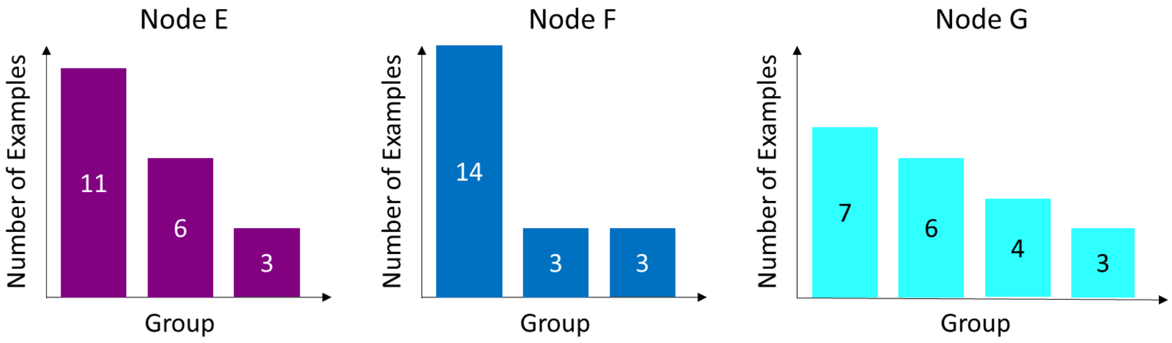


Figure 6-5: Node distribution of example core, with nodes E, F and G

Several new molecules have been imagined as shown in Table 6-2. Each row represents a new molecule that has been mapped to the core. If the molecule presents a new substructure at a node, this is indicated by “New subst” at the relevant node position in column three. If the new molecule has a substructure that is already present in the relevant node of the core, then the appropriate bin is highlighted in red in column two and the bin number is given in column 3.

Table 6-2: Imagined Core Scenarios

Row No.	Prior Distributions (E, F, G)	New Molecule (E, F, G)
15	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(New subst), (New subst), (New subst)
16	(11, 6, **3**), (14, 3, 3), (7, 6, 4, 3)	(Bin 3), (New subst), (New subst)
17	(11, 6, 3), (14, 3, **3**), (7, 6, 4, 3)	(New subst), (Bin 3), (New subst)
18	(11, 6, 3), (14, 3, 3), (7, 6, 4, **3**)	(New subst), (New subst), (Bin 4)
19	(11, 6, **3**), (14, 3, **3**), (7, 6, 4, **3**)	(Bin 3), (Bin 3), (Bin 4)
20	(11, 6, **3**), (14, 3, **3**), (7, 6, **4**, 3)	(Bin 3), (Bin 3), (Bin 3)
21	(11, **6**, 3), (14, 3, **3**), (7, 6, 4, **3**)	(Bin 2), (Bin 3), (Bin 4)
22	(11, 6, **3**), (14, 3, **3**), (7, **6**, 4, 3)	(Bin 3), (Bin 3), (Bin2)
23	(11, **6**, 3), (14, 3, **3**), (7, **6**, 4, 3)	(Bin 2), (Bin 3), (Bin2)
24	(**11**, 6, 3), (14, 3, **3**), (7, 6, 4, **3**)	(Bin 1), (Bin 3), (Bin 4)
25	(11, 6, **3**), (**14**, 3, 3), (7, 6, 4, **3**)	(Bin 3), (Bin 1), (Bin 4)
26	(**11**, 6, 3), (**14**, 3, 3), (7, 6, 4, **3**)	(Bin 1), (Bin 1), (Bin 4)
27	(**11**, 6, 3), (**14**, 3, 3), (**7**, 6, 4, 3)	(Bin 1), (Bin 1), (Bin 1)

An ideal ordering of the molecules in Table 6-2 was determined by manually considering the extent to which the molecules explore new chemical space. Row 15 should score the highest as this molecule introduces a new substructure to all three nodes. The next three rows (16, 17 and 18) all present new substructures to two of the nodes and an existing substructure to a third node, albeit a different node in each case. Although the existing substructure is added to a bin of occupancy three in each case, the underlying distributions are different and therefore the node exploration scores, and consequently the molecule exploration scores, should be different. The molecule adding an existing substructure to node F should score higher than that adding to node E which in turn should be higher than that adding to node G. This is because F has the most highly skewed distribution which is being smoothed followed by E and then G. Therefore, the ordering of the rows should be row 17 > row 16 > row 18. Row 19 should be ranked next; it adds previously seen substructures to each node but in each case this is to the bin with lowest occupancy.

The ideal ranking of the next six rows is not as clear cut, however, rows 20, 21 and 22 should all score higher than rows 23, 24 and 25 as in the first three cases the skewness of the distributions are smoothed whereas in the last three cases the skewness is increased.

Comparing rows 20 and row 21, the distributions for row 20 become less skewed than for row 21, therefore, the ideal ranking would rank 20 higher than 21 in terms of exploration.

Comparing rows 21 and 22, for both nodes E and G different segments are added to. Node E is initially a more skewed distribution. Both rows for node G smooth out the distributions row 22 only slightly more. However, as node E is initially a more skewed distribution and row 21 contributes more to smoothing out the distribution, overall row 21 should score a slightly larger exploration score.

Considering rows 23, 24 and 25, row 23 should have a larger exploration score than row 24, because adding to the two groups of six does not have as big an impact on the distribution as adding to the group with 11 examples. Row 24 should have a higher exploration score than row 25, since both add to the largest value for node F which makes both the distributions more skewed, however, this is more pronounced for row 25, therefore, it should have a lower exploration score than row 24. As row 26 adds to the highest number of examples for two of the nodes, this is the next expected row in the ranking. Row 27 should have the lowest score, as it adds to the substructure with the highest number of existing examples for all three nodes.

In summary, putting all the above information together the ideal ordering should be: row 15 > row 17 > row 16 > row 18 > row 19 > row 20 > row 21 > row 22 > row 23 > row 24 > row 25 > row 26 > row 27.

6.2.1.2 Real LO data

The examples shown previously are based on hypothetical data. Here, data from a real LO series was examined. RG cores were extracted from 90% of the P2x7 Subset dataset selected at random and one of the cores was selected. Six of the molecules in the remaining 10% of the dataset that matched this core were then analysed.

As shown in Chapter 3, more than one core can be generated for a given LO series. Therefore, it would be desirable to have a score that could be used to compare exploration scores across different cores. The final experiment compared the scores generated when molecules are mapped to different cores. Comparing molecules with differing cores is referred to as cross core comparison. In this case, the cores can have different numbers of nodes with different numbers and distributions of substructures. The cores can also represent different numbers of molecules. Six molecules across four other cores were used from the P2x7 Subset test set. The six molecules were examined to understand if molecules that map to different cores can be scored and ranked effectively.

6.2.2 Different Implementations of an Exploration Score

A number of different node scores were investigated. These are prior probability, information entropy, Kullback-Leibler divergence and an adaptation to RG cores of a score developed for screening collection design, designated here as Collection Score. Each of these methods was used to develop a node score as described below.

As mentioned above, three different methods were explored to generate a molecule score from the node scores: summing; multiplying; or taking the mean of the node scores.

6.2.2.1 Prior Probability

This score is based on the prior probability of the substructure that is mapped onto a node of the core. The prior probability of substructure *i* is given by: $p_i = x/N$ where *x* is the frequency of the bin the substructure *i* maps to, and *N* is the total number of molecules in the LO series. The exploration score is then formed by subtracting the prior probability from one (Ep= 1- p_i). Therefore, a higher value means the substructure in the new molecule has a low likelihood and therefore scores highly with respect to exploration.

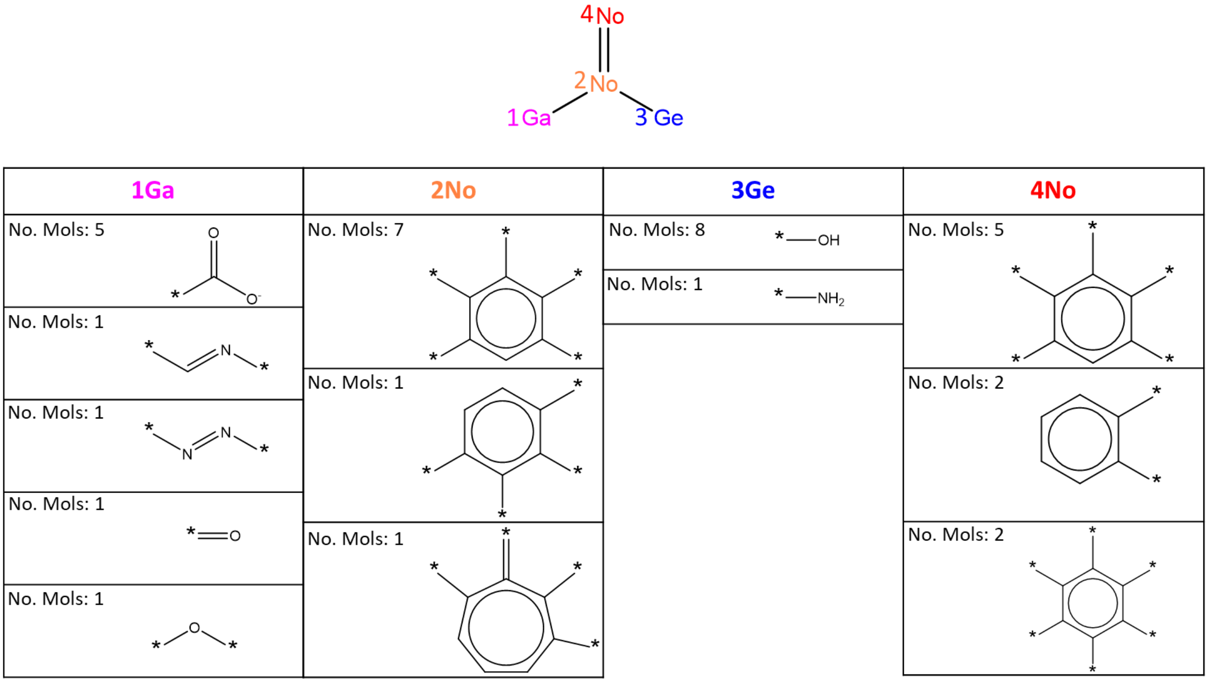


Figure 6-6: RG core breakdown

An example is shown in Figure 6-6 to demonstrate how this score is calculated. The RG core represents nine molecules. Each column indicates the substructures represented by each node within the core shown at the top of the table along with the number of molecules in the core that contain the substructure. Considering node four, and a new molecule which contains the last substructure (the fully substituted benzene) shown in the table then:

$$\begin{gathered} E_P = 1 - \frac{x}{N} \\ E_P = 1 - \frac{2}{9} \\ E_P = 0.778 \end{gathered} \tag{6.1}$$

Where *x/N* equals 2/9 as the substructure has two previously seen examples out of a total of nine.

The prior probability score is the simplest of the scores investigated as it relies only on knowing the prior probability of the substructure that is being added to the series. However, a limitation of the prior probability score is that it does not take into account the distribution of substructures within the node.

6.2.2.2 Change in Information Entropy

Information entropy, or Shannon entropy, is an approach used to calculate how much information, order or uncertainty, is within a system or event (Shannon, 1948). The more certain and ordered a system is, the less information it contains, and the lower entropy it has. A highly probable event within a system does not provide much information as this event is unsurprising, compared to an improbable event that provides a large amount of information as this event's occurrence is surprising and rare. Information entropy takes the distribution of the events into account and not just the total number. As entropy is a measure of uncertainty, a high entropy score indicates that there is a large amount of uncertainty and there is no order in any potential outcome (Shannon, 1948). When examining two different systems, Figure 6-7a shows a highly ordered system with a lower information entropy than a highly disordered system such as Figure 6-7b. A new event corresponding to the third bin in each case would be less surprising for case a) compared to b) and would therefore add less information to case a). It is important to note that if there are two entropy scores, for example, 1.5 and 3, that there is more uncertainty in the larger value, however, the values are not scalable, meaning that there is not twice as much uncertainty in the larger value.

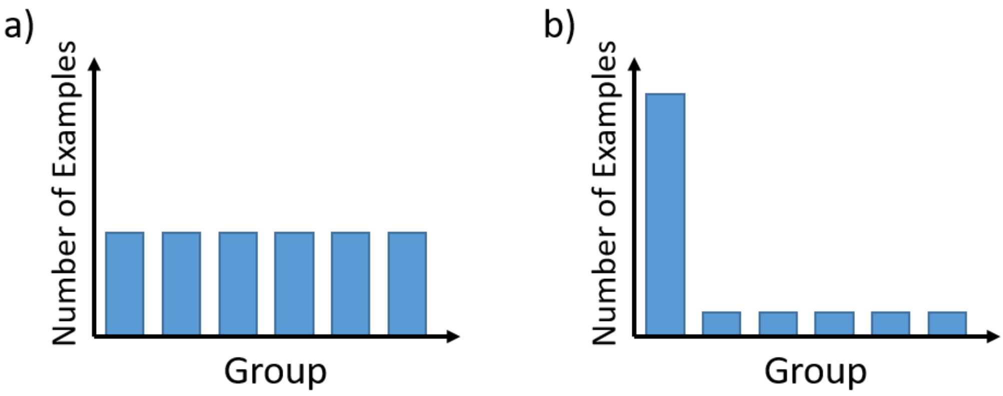


Figure 6-7: Information Entropy Graphs a) highly ordered b) highly disordered

The information entropy for a node is calculated before any new information is added into the system, using the following equation.

$$H = -\sum_i p_i ln\, p_i \tag{6.2}$$

Where p_i is the proportion of substructure i within the node, this can be calculated as $\frac{x}{N}$ where *x* is the number of times substructure *i* appears and *N* is the number of molecules related to that core and the summation is over the different substructures represented by the node. This equation is used as all events are independent.

The minimum entropy value is zero. A value of zero occurs when there is only one state or event within a system as it is highly probable that this event will occur. The maximum entropy is when all events are equally as likely. Therefore, the maximum entropy is equal to the natural log of the number of substructure derivatives present for that node, i.e. for a node that has three substructure values $H_{max} = ln\,(3) = 1.099$.

The exploration score is defined as the change in entropy when a new molecule is added which is calculated by subtracting the entropy before from the entropy afterwards.

Considering the example in Figure 6-6, the entropy is calculated for the node 4No prior to considering the new molecule, *H_B*.

$$\begin{gathered} H_B = -\sum_i \left(\frac{5}{9} ln\, \frac{5}{9}\right) + \left(\frac{2}{9} ln\, \frac{2}{9}\right) + \left(\frac{2}{9} ln\, \frac{2}{9}\right) \\ H_B = -\sum_i (-0.327) + (-0.334) + (-0.334) \\ H_B = 0.995 \end{gathered} \tag{6.3}$$

The information entropy is then calculated with the new molecule added, H$_A$, when the number of molecules is increased by 1 to 10 and the distribution of substructures is now: 5;2;3 (compared to 5:2:2). The calculation is as follows.

$$H_A = -\sum_i \left(\frac{5}{10} ln\ \frac{5}{10}\right) + \left(\frac{2}{10} ln\ \frac{2}{10}\right) + \left(\frac{3}{10} ln\ \frac{3}{10}\right)$$
$$H_A = -\sum_i (-0.347) + (-0.322) + (-0.361) \qquad (6.4)$$
$$H_A = 1.030$$

The exploration score is then calculated as the change in entropy:

$$\Delta H = H_A - H_B \qquad (6.5)$$
$$= 1.030 - 0.995 = 0.035$$

If a new substructure is introduced in the new molecule, then a new term is added to the calculation, as shown on the left of Figure 6-8. A negative score can sometimes occur when adding a substructure with a large number of previously seen examples, as shown on the right of Figure 6-8. This indicates that a negative score is always generated when the new state is more disordered.

New Substructure Example

$$H_B = -\sum_i \left(\frac{5}{9}\ln\frac{5}{9}\right) + \left(\frac{2}{9}\ln\frac{2}{9}\right) + \left(\frac{2}{9}\ln\frac{2}{9}\right)$$
$$H_B = -\sum_i (-0.327) + (-0.334) + (-0.334)$$
$$H_B = 0.995$$
$$H_A = -\sum_i \left(\frac{5}{10}\ln\frac{5}{10}\right) + \left(\frac{2}{10}\ln\frac{2}{10}\right) + \left(\frac{2}{10}\ln\frac{2}{10}\right) + \left(\frac{1}{10}\ln\frac{1}{10}\right)$$
$$H_A = -\sum_i (-0.347) + (-0.322) + (-0.322) + (-0.230)$$
$$H_A = 1.221$$
$$= 1.221 - 0.995 = 0.226$$

Negative Example

$$H_B = -\sum_i \left(\frac{5}{9}\ln\frac{5}{9}\right) + \left(\frac{2}{9}\ln\frac{2}{9}\right) + \left(\frac{2}{9}\ln\frac{2}{9}\right)$$
$$H_B = -\sum_i (-0.327) + (-0.334) + (-0.334)$$
$$H_B = 0.995$$
$$H_A = -\sum_i \left(\frac{6}{10}\ln\frac{6}{10}\right) + \left(\frac{2}{10}\ln\frac{2}{10}\right) + \left(\frac{2}{10}\ln\frac{2}{10}\right)$$
$$H_A = -\sum_i (-0.307) + (-0.322) + (-0.322)$$
$$H_A = 0.950$$
$$= 0.950 - 0.995 = -0.045$$

Figure 6-8: Examples of a new substructure calculation and an example where a negative value is found

6.2.2.3 Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence is a way of quantifying and comparing two probability distributions. An exploration score based on KL divergence is calculated using the probabilities of the node substructures before and after a new molecule is added to the series.

$$KL\ divergence = -\sum p_i ln\left(\frac{q_i}{p_i}\right) \qquad (6.6)$$

Where p_i is the probability of the substructure before; q_i is the substructure's probability after a new molecule has been added into the system; and the summation is over the different substructures (or bins) represented by the node.

An example of how the KL divergence is calculated is shown below using the same example as previously seen, where the new molecule is being added to the last substructure for node 4No in Figure 6-6. p_i is 5/9 for the first substructure; and 2/9 for the last two. As the new molecule maps to the third substructure, the q_i values are, 5/10, 2/10 and 3/10 as the denominator for all the substructures is incremented by one to represent the new molecule being added into the core. The third numerator is also incremented as this is the substructure that the new molecule possesses. These values are input to equation 6.6 as follows (equation 6.7)

$$\begin{aligned} KL\ divergence \\ &= -\sum_i \left(\frac{5}{9} ln\left(\frac{5/10}{5/9}\right)\right) + \left(\frac{2}{9} ln\left(\frac{2/10}{2/9}\right)\right) \\ &+ \left(\frac{2}{9} ln\left(\frac{3/10}{2/9}\right)\right) \\ = -\sum_i (-0.059) + (-0.023) + (0.067) \\ = 0.015 \end{aligned} \tag{6.7}$$

A limitation of KL divergence for this problem is that a score cannot be calculated for a new substructure because the prior probability of that substructure, p_i, is zero which would lead to a division by zero. To avoid this problem alpha smoothing is used. Alpha smoothing is a technique used to eliminate zero values while retaining the original ratios. When calculating p_i and q_i an alpha value is added to both the denominator and the numerator so that $\frac{x}{N}$ becomes $\frac{x+\alpha}{N+\alpha}$. Where *x* is the number of times substructure *i* appears and *N* is the number of molecules related to that core and alpha is a value that is set by the user. In the experiments carried out within this chapter alpha equals 0.01.

6.2.2.4 Collection Model

The collection model score was developed to select a balanced set of compounds for high-throughput screening (HTS) (Harper, Pickett, & Green, 2004). The score is used to relate chemical similarity (as defined by clusters) to potential biological activity to estimate the number of clusters that contain lead compounds in HTS datasets. A higher value is more favoured as this means more clusters are likely to contain active molecules. The equation used for to calculate a collection model score is:

$$E = \sum_{i=1}^{p} \pi_i [1 - (1 - \alpha_i)^{N_i}] \qquad (6.8)$$

Where *p* is the number of clusters, *i* is a cluster, alpha is 0.3 (a property of the similarity method), *N_i* is the number of compounds in cluster *i,* π is the probability of there being a molecule active towards a particular assay within a set of molecules. π can be varied as some datasets could have a higher hit rate. By summing over the clusters, the collection model score takes account of the distribution of molecules across clusters. It was believed that this would be a good equation to adapt as the score as the distribution of molecules across the clusters can be considered similar to the distribution of molecules across the different substructures represented by a node.

To adapt the collection score to an exploration score, *p* is the number of distinct substructural fragments for the node and N_i is the frequency of occurrence of substructure *i*. π is set to 1 as there is no equivalent term to hit rate, and the same value of α is used as in the HTS experiments (0.3).

The first adaption of *E*, referred to as E1, is the change in the *E* scores following the addition of the new molecule to the core, and the summation is over the different substructures (or bins), p, represented by the node. A higher score indicates a higher exploration.

$$E1 = \sum_{after}^{p} [1 - (1 - \alpha)^{N_i}] - \sum_{before}^{p} [1 - (1 - \alpha)^{N_i}] \qquad (6.9)$$

E1 considers the absolute counts of the substructures, rather than proportions, and therefore it only accounts for the bin that changes due to the addition of the given substructure. All other bin values are unchanged and therefore cancel out of the equation.

A number of adaptions were then made in order to take the full distributions into account. The second method, E2, is based on E1, however, this is divided by the before E value, which therefore differentiates between different distributions.

$$E2 = \frac{\sum_{after}^{p}[1-(1-\alpha)^{N_i}] - \sum_{before}^{p}[1-(1-\alpha)^{N_i}]}{\sum_{before}^{p}[1-(1-\alpha)^{N_i}]} \qquad (6.10)$$

In the third method, E3, the count of each substructure is replaced by the probability. Where p_i is the probability before and q_i is the probability after.

$$E3 = \sum_{after}^{p} [1 - (1 - \alpha)^{q_i}] - \sum_{before}^{p} [1 - (1 - \alpha)^{p_i}] \qquad (6.11)$$

For the fourth method, E4, the *N_i* is replaced by the difference in the probability for substructure derivative *i*, p_i-q_i. This is to avoid negative powers since the before probability minus the after probability is a positive value for all the substructures that are unchanged.

$$E4 = -\sum_{i=1}^{p}[1-(1-\alpha)^{p_i-q_i}] \tag{6.12}$$

The final method, E5, is based on E4, however, it scales the score between zero and one, by dividing the sum by the maximum value E4 can take. The maximum value E4 can take is when all the substructures represented by a node are the same bin (i.e. there is only one bin) and the new molecule presents a new substructure. The maximum value is then dependent on the value of *N*, that is, the number of molecules represented by the core.

$$E5 = \frac{-\sum_{i=1}^{p}[1-(1-\alpha)^{p_i-q_i}]}{E4_{max}} \tag{6.13}$$

The above five equations are applied to the example shown in Figure 6-6 where a new molecule is added to the last substructure in node 4No in Figure 6-6. There are three different substructures with prior distribution (5, 2, 2) and the after distribution (5, 2, 3).

$$\begin{aligned} E1 = ([1-(1-0.3)^5] + [1-(1-0.3)^2] + [1-(1-0.3)^3]) \\ -\ ([1-(1-0.3)^5] + [1-(1-0.3)^2] + [1-(1-0.3)^2]) \\ = (0.832 + 0.510 + 0.657) - (0.832 + 0.510 + 0.510) \\ E1\ = 1.999 - 1.852 = 0.147 \end{aligned} \tag{6.14}$$

$$\begin{aligned} &E2 \\ &= \frac{([1-(1-0.3)^5] + [1-(1-0.3)^2] + [1-(1-0.3)^3]) -\ ([1-(1-0.3)^5] + [1-(1-0.3)^2] +}{([1-(1-0.3)^5] + [1-(1-0.3)^2] + [1-(1-0.3)^2])} \\ &= \frac{(0.832 + 0.510 + 0.657) - (0.832 + 0.510 + 0.510)}{(0.832 + 0.510 + 0.510)} \\ &E2 = \frac{1.999 - 1.852}{1.852} = 0.079 \end{aligned} \tag{6.15}$$

$$\begin{aligned} E3 = \left(\left[1-(1-0.3)^{5/10}\right] + \left[1-(1-0.3)^{2/10}\right] + \left[1-(1-0.3)^{3/10}\right]\right) \\ -\ \left(\left[1-(1-0.3)^{5/9}\right] + \left[1-(1-0.3)^{2/9}\right] \right. \\ \left. + \left[1-(1-0.3)^{2/9}\right]\right) \\ E3 = (0.163 + 0.069 + 0.101) + (0.180 + 0.076 + 0.076) \\ E3 = 0.334 - 0.332 = 0.002 \end{aligned} \tag{6.16}$$

$$\begin{aligned} E4 = -\left[\left(\left[1-(1-0.3)^{\left(\frac{5}{9}-\frac{5}{10}\right)}\right]\right) + \left(\left[1-(1-0.3)^{\left(\frac{2}{9}-\frac{2}{10}\right)}\right]\right) \right. \\ \left. + \left(\left[1-(1-0.3)^{\left(\frac{2}{9}-\frac{3}{10}\right)}\right]\right)\right] \\ E4 = -[(0.020) + (0.008) + (-0.028)] \end{aligned} \tag{6.17}$$

$$E4 = 0.0006$$

$$E5 = \frac{-\left[\left(\left[1-(1-0.3)^{\left(\frac{5}{9}-\frac{5}{10}\right)}\right]\right)+\left(\left[1-(1-0.3)^{\left(\frac{2}{9}-\frac{2}{10}\right)}\right]\right)+\left(\left[1-(1-0.3)^{\left(\frac{2}{9}-\frac{3}{10}\right)}\right]\right)\right]}{-\left[\left(\left[1-(1-0.3)^{\left(\frac{9}{9}-\frac{9}{10}\right)}\right]\right)+\left(\left[1-(1-0.3)^{\left(\frac{0}{9}-\frac{1}{10}\right)}\right]\right)\right]} \tag{6.18}$$

$$E5 = \frac{-[(0.020)+(0.008)+(-0.028)]}{-[(0.035)+(-0.036)]}$$

$$E5 = \frac{0.0006}{0.0013} = 0.483$$

6.3 Results and Discussion

All of the exploration scores were calculated for the hypothetical cases represented in Table 6-1 and Table 6-2.

6.3.1 Node Score

The simple node examples were examined first. As reported above, the ideal ordering is: 2 > 5 > 8 > 11 > 4 > 7 > 13 > 12 > 14 > 10 > 9 > 6 > 3 > 1.

Table 6-3: Single node prior probability, change in entropy and KL divergence scores

Row No.:Node No	Prior Distribution	New Molecule	Prior Prob	Change in Entropy	KL Divergence
1:A	(20) i.e. single substituent	Bin 1	0	0	0
2:A	(20) i.e. single substituent	New subst	1	0.191	0.046
3:B	(19,1)	Bin 1	0.05	-0.007	8.58 E-05
4:B	(19,1)	Bin 2	0.95	0.116	0.014
5:B	(19,1)	New subst	1	0.182	0.047
6:C	(18,1,1)	Bin 1	0.1	-0.014	0.0001
7:C	(18,1,1)	Bin 2	0.95	0.501	0.014
8:C	(18,1,1)	New subst	1	0.567	0.047
9:D	(11,9)	Bin 1	0.45	-0.005	0.0009
10:D	(11,9)	Bin 2	0.55	0.004	0.001
11:D	(11,9)	New subst	1	0.159	0.047
12:E	(11,6,3)	Bin 3	0.85	0.038	0.006
13:F	(14,3,3)	Bin 3	0.85	0.045	0.006
14:G	(7,6,4,3)	Bin 4	0.85	0.021	0.006

Table 6-3 demonstrates the results that were generated for the prior probability, change in entropy and KL divergence methods.

The prior probability method generates scores in the range zero to one. However, this method cannot distinguish between distributions with different skewness and distributions consisting of a different number of substituents or bins. For example, the same scores are derived for rows 12 and 13 despite the prior distribution of row 13 being more skewed. Similarly, the same score is derived for row 14 despite there being four substructures or bins in this case. Furthermore, the addition of a new substructure always results in a score of one regardless of the prior distribution, as shown for rows 2, 5, 8 and 11.

In contrast, the change in information entropy method is able to distinguish distributions with different skewness's and with different numbers of variables/substituents. This includes cases when new substructures are added, shown by the different values produced for rows 2, 5, 8 and 11. The ability to discriminate between these rows is important as it allows the identification between different distributions. Therefore, the score is not just based upon the substructure being added but on the whole system. However, the entropy scores do not scale well, as there are some negative values and it is therefore not possible to create a score between zero and one. The negative scores occur when adding a new substructure increases the skewness of the distribution, e.g., for rows 3, 6 and 8. These values are problematic as ideally the exploration score for these cases should be larger than for row 1. Row 1 should be the lowest score as it represents the case where all the existing molecules in the LO series have the same substructure at this position (node) and the new molecule presents another example of the same substructure, i.e., there has been no exploration of this region of the chemical space. The negative values could also cause issues when combining the nodes to generate a molecular exploration score.

KL divergence does not have the issue of negative scores. However, the range of values is extremely small with the maximum value of 0.046, and, as for the prior probability score, it does not have the ability to differentiate between different distributions skewness's and number of variables/substituents.

Table 6-4: Single node collection model based scores

Row No.: Node No	Prior Distribution	New Molecule	E1	E2	E3	E4	E5
1:A	(20) i.e. single substituent	Bin 1	0.0002	0.00024	0	0	0
2:A	(20) i.e. single substituent	New subst	0.300	0.300	0.005	0.00029	1

3:B	(19,1)	Bin 1	0.0003	0.0003	-0.0002	7.21E-07	0.003
4:B	(19,1)	Bin 2	0.210	0.162	0.004	0.00026	0.902
5:B	(19,1)	New subst	0.300	0.231	0.004	0.00028	0.953
6:C	(18,1,1)	Bin 1	0.0005	0.0003	-0.0004	2.16E-06	0.008
7:C	(18,1,1)	Bin 2	0.210	0.131	0.004	0.00025	0.858
8:C	(18,1,1)	New subst	0.300	0.188	0.004	0.00026	0.908
9:D	(11,9)	Bin 1	0.006	0.003	-0.0003	5.84E-05	0.202
10:D	(11,9)	Bin 2	0.012	0.006	0.0002	8.73E-05	0.302
11:D	(11,9)	New subst	0.300	0.155	0.003	0.0002	0.755
12:E	(11,6,3)	Bin 3	0.103	0.041	0.001	0.00016	0.559
13:F	(14,3,3)	Bin 3	0.103	0.045	0.002	0.00018	0.618
14:G	(7,6,4,3)	Bin 4	0.103	0.032	0.0005	0.00014	0.489

Table 6-4 shows the results for the different collection model score variations.

As discussed above, the E1 score produces results that are similar to the prior probability method as it does not discriminate between different distributions. Furthermore, the values range from 0 to 0.3 (α).

E2 differentiates between different distributions, such as rows 2, 5, 8 and 11, due to the division by the before E value. Furthermore, none of the scores are negative values. As for E1 the values range from 0 to 0.3.

E3 uses the probabilities of the substructures before and after the new molecule is added. This method is therefore able to differentiate between, for example, rows 12, 13 and 14 and ranks these rows in the desired order. However, it does not differentiate between rows 4, 5 and 7. Furthermore, the E3 scores are very low due to the power term being small, and for this example, it has a maximum value of 0.005. Also, there are some negative values. Negative values are caused when the skewness is increased by adding a substructure to a bin with a large percentage of the previously seen examples.

To overcome the negative values, E4 replaces N_i by the difference in the probabilities and, therefore, a comparison did not have to be drawn between E4 before and E4 after. Using the probabilities solved the issue of having negative results and the score still differentiates between different distributions. However, this score has even smaller power terms than seen with E2 and so the scale is even smaller, and for this example, the maximum value was 0.00029.

E5 combats the very small values associated with E4 by dividing by the maximum value E4 could be to scale the score between zero and one. The maximum score E4 is when all previously seen substructures are in only one existing group and the new substructure has not been seen before.

The maximum value varies depending on the number of previously seen examples. There are no negative values and E5 also discriminates between the distributions. Two key results to note are row 4 and 8, where the E4 scores appear to be the same, however, the E5 scores are slightly different. The difference in scores is due to the scaling and if the results are shown to another decimal place then row 4 would be 0.000260 and row 8 0.000262. Therefore, the scaling allows these two cases to be differentiated more easily.

The orderings of the rows for each of the methods in Table 6-3 and Table 6-4 are seen below. If more decimal places were examined, some equal values may have had slight variations, however, it was felt the number of decimal places that have been reported was sufficient.

Prior Probability: 2 = 5 = 8 = 11 > 4 = 7 > 12 =13 =14 > 10 > 9 > 6 > 3 > 1.

Change in entropy: 8 > 7 > 2 > 5 > 11> 4 > 13 > 12 > 14 > 10 > 1 > 9 > 3 > 6.

KL divergence: 5 = 8 = 11 > 2 > 4 = 7 > 12 = 13 = 14 > 10 > 9 > 6 > 3 > 1.

E1: 2 = 5 = 8 = 11 > 4 = 7 > 12 = 13 = 14 > 10 > 9 > 6 > 3 > 1.

E2: 2 > 5 > 8 > 4 > 11 > 7 > 13 > 12 > 14 > 10 > 9 > 6 > 3 > 1.

E3: 2 > 4 = 5 = 7 = 8 > 11 > 13 > 12> 14 > 10 > 1 > 3 > 9 > 6.

E4: 2 > 5 > 4 = 8 > 7 > 11 > 13 > 12 > 14 > 10 > 9 > 6 > 3 > 1.

E5: 2 > 5 > 8 > 4 > 7 > 11 > 13 > 12 > 14 > 10 > 9 > 6 > 3 > 1.

Unfortunately, none of the above methods achieves the ideal ordering. However, the orderings for E2 and E5 are close to the ideal. They differ slightly from the ideal as row 11 is expected to score higher than row 4 and 7, however, it scores lower than one or both of them. Row 11, is where a new substructure is being added to a relatively even distribution consisting of two substructures with eleven and nine examples. The other rows, 4 and 7, add another instance to an existing substructure with just one previous example. This is, therefore, acceptable if one or both of row 4 and 7 score higher than row 11 as it promotes the exploration of groups with a low number of examples and new substructures. Therefore, when new molecules are suggested it will not always constantly promote the introduction of new substructures, i.e. a full enumeration and allow the substructure with lower number of examples to be further analysed too.

6.3.2 Molecule Score

The molecule score investigation examines the effects of combining node scores together. The ideal ordering of the molecules should be: row 15 > row 17 > row 16 > row 18 > row 19 > row 20 > row 21 > row 22 > row 23 > row 24 > row 25 > row 26 > row 27.

Three methods were considered for combining the node scores to form molecule scores. These are to sum them, to multiple them, and to find the average of them. Results are shown in Table 6-5 and Table 6-6 where the rows have been switched around so that they are in the ideal order so that a scoring method that is able to reproduce the ideal ordering would have values that decrease going down the tables. Table 6-5 shows each node score for each of the methods and different additions. Table 6-6 shows the calculated overall exploration scores for the molecules by summing, multiplying and calculating the mean node score for all the nodes. The other remaining scores not shown can be found in the Appendix.

Table 6-5: Results for each node for this experiment for the Collection Model Score Variations

Row No.	Prior Distributions (E, F, G)	E2	E5
15	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.119), (0.130), (0.093)	(0.710), (0.769), (0.640)
17	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.041), (0.130), (0.093)	(0.710), (0.618), (0.640)
16	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.119), (0.045), (0.093)	(0.559), (0.769), (0.640)
18	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.119), (0.130), (0.032)	(0.710), (0.769), (0.489)
19	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.041), (0.045), (0.032)	(0.559), (0.618), (0.489)
20	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.041), (0.045), (0.022)	(0.559), (0.618), (0.439)
21	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.014), (0.045), (0.032)	(0.408), (0.618), (0.489)
22	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.041), (0.045), (0.011)	(0.559), (0.618), (0.338)
23	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.014), (0.045), (0.011)	(0.408), (0.618), (0.338)
24	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.002), (0.045), (0.032)	(0.158), (0.618), (0.489)
25	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.041), (0.0009), (0.032)	(0.559), (0.068), (0.489)
26	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.002), (0.0009), (0.032)	(0.158), (0.068), (0.489)

27	(**11**, 6, 3), (**14**, 3, 3), (**7**, 6, 4, 3)	(0.002), (0.0009), (0.008)	(0.158), (0.068), (0.288)

Table 6-6: Combined Overall Scores for this experiment for the Collection Model Score Variations

Row No.	**E2**			**E5**		
	Sum	**Multiplied**	**Mean**	**Sum**	**Multiplied**	**Mean**
15	0.342	0.001	0.114	2.119	0.350	0.706
17	0.264	0.0005	0.088	1.968	0.281	0.656
16	0.257	0.0005	0.086	1.968	0.275	0.656
18	0.281	0.0005	0.094	1.968	0.267	0.656
19	0.117	5.82E-05	0.039	1.666	0.169	0.555
20	0.108	4.08E-05	0.036	1.616	0.152	0.539
21	0.091	1.99E-05	0.030	1.515	0.123	0.505
22	0.096	2.00E-05	0.032	1.515	0.117	0.505
23	0.070	6.85E-06	0.023	1.365	0.085	0.455
24	0.079	3.36E-06	0.026	1.265	0.048	0.422
25	0.074	1.15E-06	0.025	1.115	0.018	0.372
26	0.035	6.64E-08	0.012	0.714	0.005	0.238
27	0.011	1.59E-08	0.004	0.513	0.003	0.171

The order of the rows using the E2 and E5 node scores are as follows:

E2: Sum and Mean: 15 > 18 > 16 > 17 > 19 > 20 > 22 > 21 > 24 > 25 > 23 > 26 > 27.

E2: Multiplied: 15 > 18 > 16 > 17 > 19 > 20 > 22 > 21 > 23 > 24 > 25 > 26 > 27.

E5: Sum and Mean: 15 > 17 = 16 =18 > 19 > 20 > 21 = 22 > 23 > 24 > 25 > 26 > 27.

E5: Multiplied: 15 > 17 > 16 > 18 > 19 > 20 > 21 > 22 > 23 > 24 > 25 > 26 > 27.

It can be seen that the sum and mean methods give the same rankings and only E5 is able to reproduce the ideal ordering and this is just for the multiplication method. The E5 scores combined using sum and mean are very similar apart from rows 16, 17 and 18 which generate the same values, as do rows 20 and 21. These scores being equal is not considered to be too detrimental as they are very similar and the individual node scores are as to be expected, however, when combined they return the same results.

Although E2 looked promising at the node level it did not work as well at the molecule level because row 16, 17 and 18 generated the reverse ordering.

The three methods used to combine the node scores into molecule scores have different advantages and disadvantages. If one of the node scores is zero, then the molecule score will be zero using the multiplication method. Summing the scores is more effective in this case, however, this score may not be appropriate when comparing different cores, which may consist of different numbers of nodes. Therefore, the mean value may be preferred as it averages the score over the nodes. This is explored later in the chapter.

6.3.3 Applying the Scores to Real Molecules

For all the datasets the molecules without pIC50 values were filtered out. A random selection of 90% of molecules were selected and the RG cores found. For each dataset, the remaining 10% of molecules were scored against all of the RG cores they matched to. Table 6-7 shows the split of data for each dataset and the number of RG cores extracted from the 90% of molecules.

Table 6-7: Table showing the data of the split datasets

Dataset	Number of Molecules in Whole Dataset with pIC50 Value	Number of Molecules in Subset of Dataset with pIC50 Value		Number of RG Cores Extracted from 90% Dataset
		10%	90%	
Bajorath	2084	208	1876	24
CDK2	1367	137	1230	105
Chk1	105	10	95	8
Cyto	6310	635	5675	189
FactorXa	1956	196	1760	40
MMP12	1704	170	1534	1
Neurokinin	1468	147	1321	27
P2x7	1786	179	1607	43
P2x7 Subset	691	69	622	7

P38α	3644	364	3280	120

One of the cores from the P2x7 Subset was selected as shown in Figure 6-9. Four molecules from the held out 10% that map to this core were selected (labelled 3, 4, 5 and 6 in Figure 6-10). In addition, two hypothetical molecules (1 and 2) were constructed to analyse how well the scores deal with extreme examples.

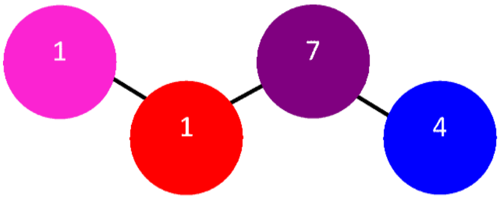


Figure 6-9: The core that these molecules have. The number within the node is how many substructures that node contains

The core, shown in Figure 6-9, is constructed from 334 molecules and consists of four nodes which are labelled according to the number of distinct substructures that each node represents. For example, the pink and red nodes each represent a single substructure so that there is no variation at these positions in the 334 molecules, whereas the purple node represents 7 different substructures. Figure 6-11 shows the distributions of substructures for each of the nodes.

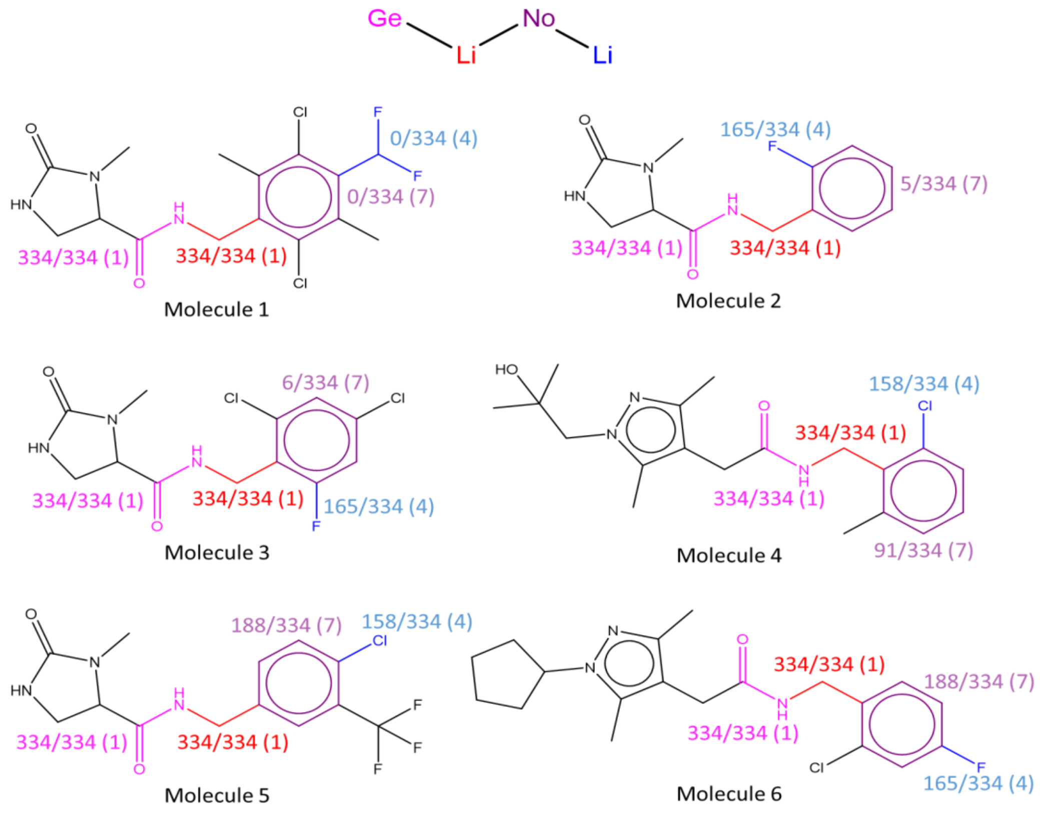


Figure 6-10: Six molecules that are being examined that match to the RG core in *Figure 6-9*

The molecules in Figure 6-10 represent new molecules that match to the core. The mapping of each molecule to the core is shown using colour coding, with the substructures coloured according to the respective node they match to. Additionally, each substructure in the molecule is labelled by the size of the bin it maps to and the number of bins is shown in brackets. For example, the pink and red substructures in molecule 2 are both labelled 334/334(1), indicating that there is just one substructure at each corresponding node in the core. The purple substructure is labelled 5/334(7), indicating that it is present in five out of 334 molecules and there are seven different substructures at this position in the core. A "0" value, e.g. the purple substructure in molecule 1, indicates that this substructure is not present in the core.

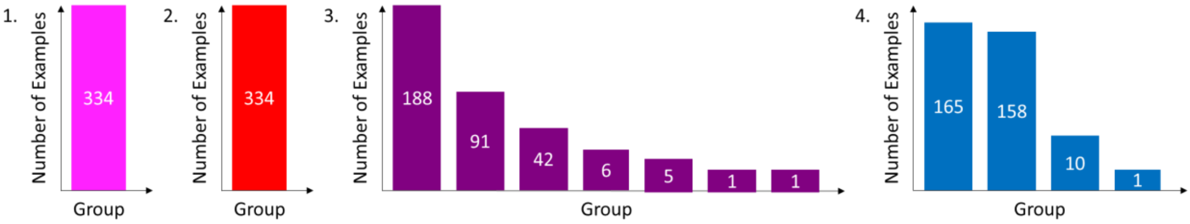


Figure 6-11: Representation of each of the nodes distributions

Before scoring the new molecules, they were inspected manually to determine the ideal ranking. Only two nodes need to be considered as in all cases the substructures that map to the first two nodes are identical to those in all molecules used to build the core.

Molecule 1 should have the highest exploration score as for both the purple and blue nodes it adds new substructures, that is, substructures that are not represented in the core. The rest of the molecules all add substructures that are already present in the core for all nodes.

Considering the blue node, the new molecules include one of the two most frequent substructures (either 165 or 158 occurrences in the core) and, of these, a slightly higher node exploration score should be obtained for the slightly less frequent substructure. For the purple node, the frequency of the group being added should influence the exploration score, with a substructure of lower frequency resulting in a higher exploration score.

As all molecules contain the same pink and red nodes then these are not taken into consideration when thinking about the overall molecule score rankings as they will be the same for all six molecules. As molecule 1 is the only molecule to introduce new substructures for the purple and blue node, this will score the highest exploration score. The next ranked molecule shall be molecule 2 closely followed by molecule 3. They both have a fluorine atom for the value node which has the highest number of seen examples for a substructure in that node, but they both have a substructure that has not been seen very often for the purple node, where molecule 2 has been seen one less time than molecule 3. Molecule 4 should be the next ranked molecule as it's blue node was observed only slightly fewer times than the previous examples but the purple node was seen a lot more times. Molecule 5 will score the second lowest exploration score as it has the same blue node as molecule 4 and a purple node that contains more observed examples. The last molecule is molecule 6 as for both the purple and blue node matches to the substructures with the highest number of examples.

Therefore, the exploration score should be highest for molecule 1 in descending order to molecule 6. There is only a marginal difference between molecule 2 and molecule 3, so it would be acceptable if these molecules scored the same. Similarly, for molecules 5 and 6 as they have the same substructure for the purple node and, although they have different substructures corresponding to the blue node, there is not much difference in the frequencies of these substructures in the core.

The molecules are presented in Figure 6-12 to Figure 6-17, respectively. For each molecule, the node breakdown and combinations for the collection model score E5 variations are shown in Table 6-8, Table 6-9, Table 6-10, Table 6-11, Table 6-12 and Table 6-13. All other scores can be found in the Appendix.

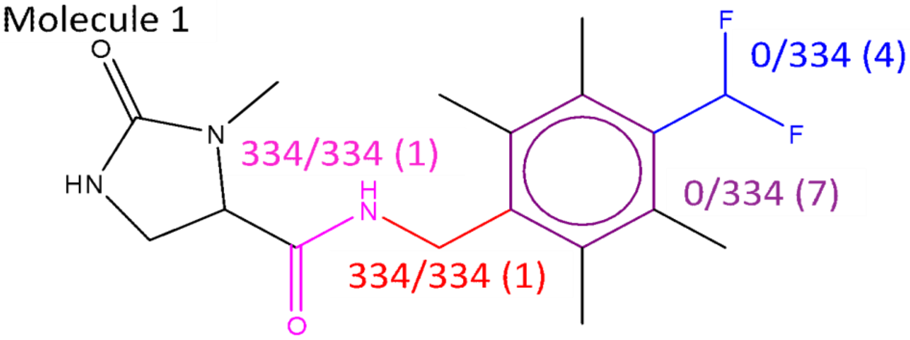


Figure 6-12: Molecule 1

Table 6-8: Node breakdown for Collection Model score Variations for Molecule 1

Node	E5
Pink	0
Red	0
Purple	0.704
Blue	0.735
Molecule Score	
Total Summed	1.438
Total Multiplied	0.000
Total Mean	0.360

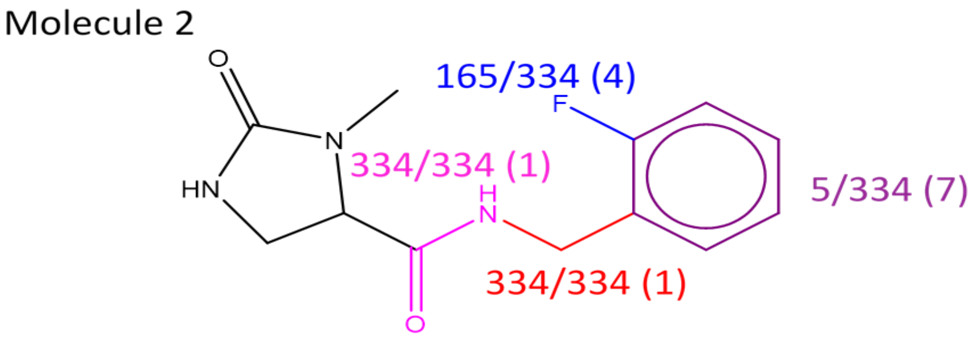


Figure 6-13: Molecule 2

Table 6-9: Node breakdown for Collection Model score Variations for Molecule 2

Node	**E5**
Pink	0
Red	0
Purple	0.689
Blue	0.240
Molecule Score	
Total Summed	0.929
Total Multiplied	0.000
Total Mean	0.232

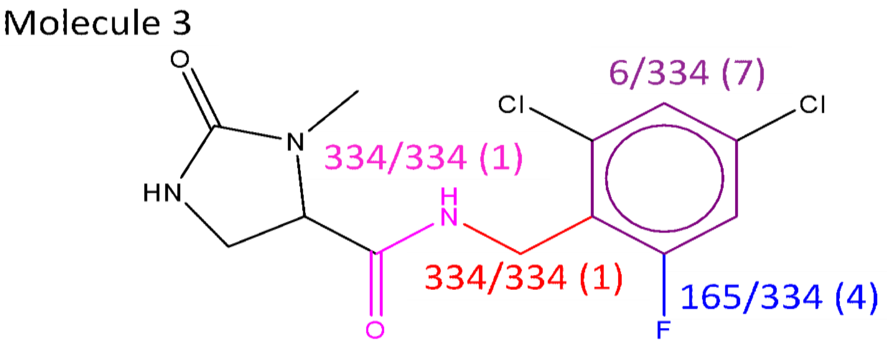


Figure 6-14: Molecule 3, CHEMBL2218567

Table 6-10: Node breakdown for Collection Model score Variations for Molecule 3

Node	**E5**
Pink	0
Red	0
Purple	0.686
Blue	0.240
Molecule Score	
Total Summed	0.926
Total Multiplied	0.000
Total Mean	0.232

Molecule 4

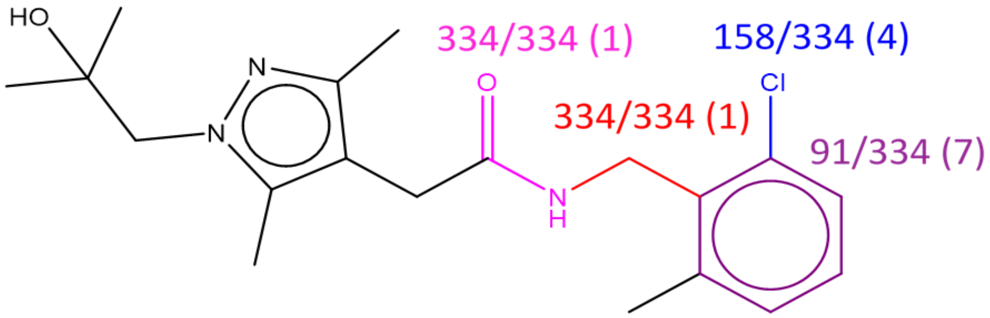


Figure 6-15: Molecule 4, CHEMBL2218612

Table 6-11: Node breakdown for Collection Model score Variations for Molecule 4

Node	**E5**
Pink	0
Red	0
Purple	0.431
Blue	0.261
Molecule Score	
Total Summed	0.693
Total Multiplied	0.000
Total Mean	0.173

Molecule 5

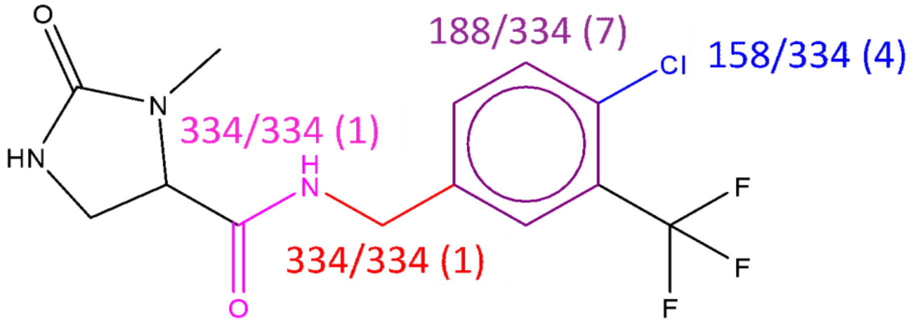


Figure 6-16: Molecule 5, CHEMBL2218364

Table 6-12: Node breakdown for Collection Model score Variations for Molecule 5

Node	**E5**
Pink	0
Red	0
Purple	0.141
Blue	0.261
Molecule Score	
Total Summed	0.402
Total Multiplied	0.000
Total Mean	0.101

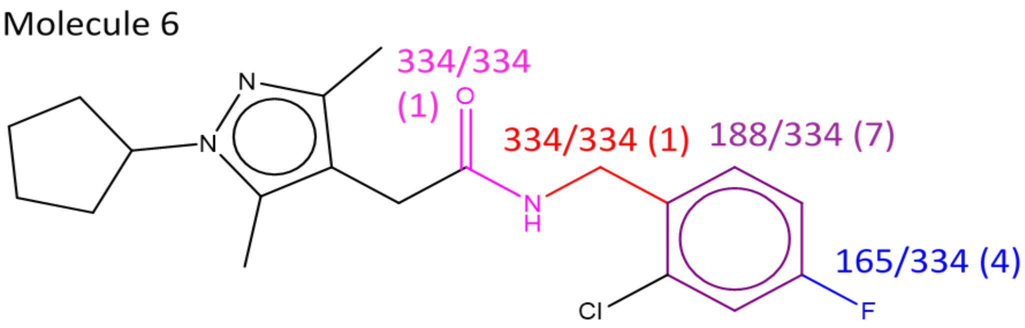


Figure 6-17: Molecule 6, CHEMBL2218425

Table 6-13: Node breakdown for Collection Model score Variations for Molecule 6

Node	**E5**
Pink	0
Red	0
Purple	0.141
Blue	0.240
Molecule Score	
Total Summed	0.381
Total Multiplied	0.000
Total Mean	0.095

Table 6-14: Displaying all the overall scores together for molecules within the same core

		Molecule 1	**Molecule 2**	**Molecule 3**	**Molecule 4**	**Molecule 5**	**Molecule 6**
1-Prior Prob	**Total Summed**	2	1.491	1.488	1.255	0.964	0.943
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.5	0.373	0.372	0.314	0.241	0.236
Change in Entropy	**Total Summed**	0.035	0.009	0.008	0.0004	-0.002	-0.002
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.009	0.002	0.002	0.0001	-0.001	-0.001
KL Divergence	**Total Summed**	0.006	0.0003	0.0002	1.76E-05	9.23E-06	8.83E-06
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.001	6.63E-05	5.63E-05	4.41E-06	2.31E-06	2.21E-06
E1	**Total Summed**	0.600	0.050	0.035	0.000	0.000	0.000
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.150	0.013	0.009	0.000	0.000	0.000
E2	**Total Summed**	0.148	0.009	0.007	0.000	0.000	0.000

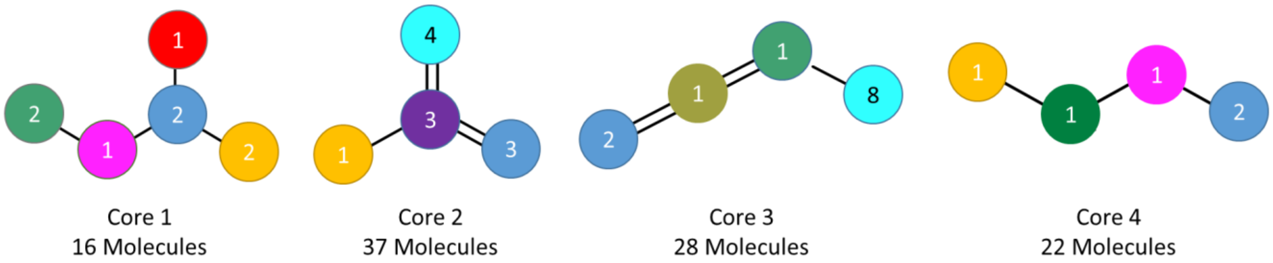
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.037	0.002	0.002	0.000	0.000	0.000
E3	**Total Summed**	0.0003	0.0001	0.0001	4.09E-05	-5.39E-05	-6.06E-05
	Total Multiplied	0	0	0	0	0	0
	Total Mean	7.60E-05	3.17E-05	3.14E-05	1.02E-05	-1.35E-05	-1.51E-05
E4	**Total Summed**	1.63E-06	1.05E-06	1.05E-06	7.85E-07	4.56E-07	4.32E-07
	Total Multiplied	0	0	0	0	0	0
	Total Mean	4.08E-07	2.63E-07	2.62E-07	1.96E-07	1.14E-07	1.08E-07
E5	**Total Summed**	1.438	0.929	0.926	0.693	0.402	0.381
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.360	0.232	0.232	0.173	0.101	0.095

The molecule scores for all molecules and all methods are shown in Table 6-14 to enable easy comparison with the ideal ranking of the molecules. The ideal ordering is molecule 1 > molecule 2 > molecule 3 > molecule 4 > molecule 5 > molecule 6 so that the scores should decrease going across the rows from left to right. Firstly, it can be seen that when the node scores are combined by multiplication all of the scores are zero. This is because two of the node scores are zero as they both only have one previously seen example which all of the molecules all add to it. When examining the other node score and molecule score combinations, the ones that replicate the ideal ordering are prior probability with summed and mean, KL divergence with summed and mean, E3 with mean, E4 with mean, and E5 with summed. Several other combinations produce rankings that are close to the ideal ordering (E3 with summed, E4 with summed and E5 with mean). The rest of the other methods do not produce the correct ordering.

Finally, it is interesting that methods E1 and E2 are unable to differentiate between molecules 4, 5 and 6. This is because all three molecules have large examples for the purple and blue node. When a large value is taken to the power of 0.7 (1-α) then this becomes a very small number, only significant at over 20 decimal places, and one minus an extremely small number becomes one.

6.3.4 Cross Core Comparison

The above molecules all match to the same core. In general, more than one core might be generated to represent a lead optimisation series, for example, this can arise if changes are made to a central scaffold. Thus, when considering new molecules to include in a LO project, different molecules may map to different cores. Therefore, it would be desirable to have a score that could be used to compare exploration scores across different cores. We refer to this as cross core comparison. In this case, the cores can have different numbers of nodes with different numbers and distributions of substructures. The cores can also represent different numbers of molecules, Figure 6-18.



Core 1
16 Molecules

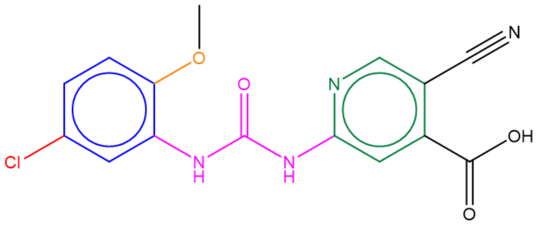
Core 2
37 Molecules

Core 3
28 Molecules

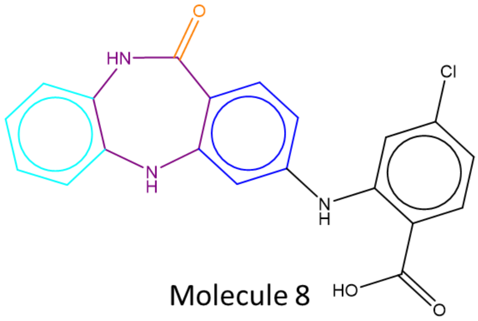
Core 4
22 Molecules

Figure 6-18: Cores being investigated

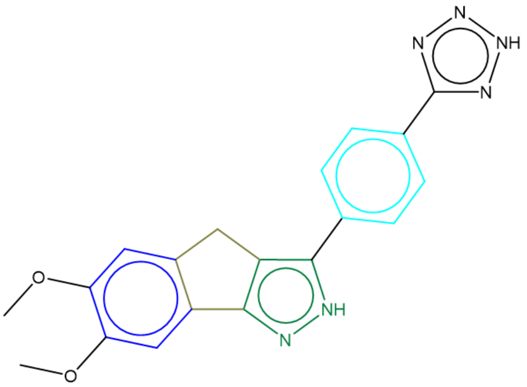
Below is an example from a LO project, 90% Chk1, where the existing data consists of four cores. Chk1 dataset was chosen as all four RG cores extracted are seen within the 10% of molecules and as it is a small dataset and it allows an easier cross-core comparison. Six new molecules, from 10% Chk1, are shown, Figure 6-19. Molecule 7 maps to core 1; molecule 8 maps to core 2; molecule 9 maps to core 3; molecule 10 maps to core 3; molecule 11 maps to core 2; molecule 12 maps to core 4. For each of the cores, the distributions of substructures represented by the nodes are shown in Figure 6-20, Figure 6-21, Figure 6-22 and Figure 6-23 to help with the analysis of formulating the expected ordering.



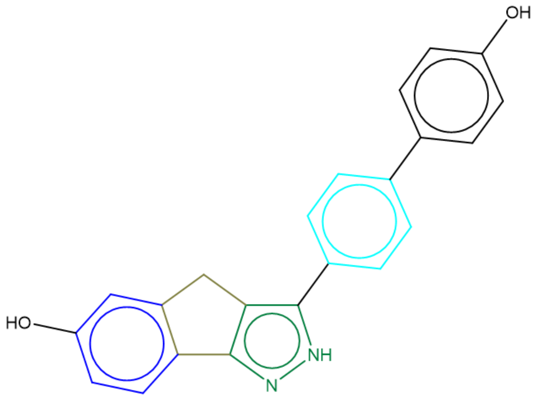
Molecule 7



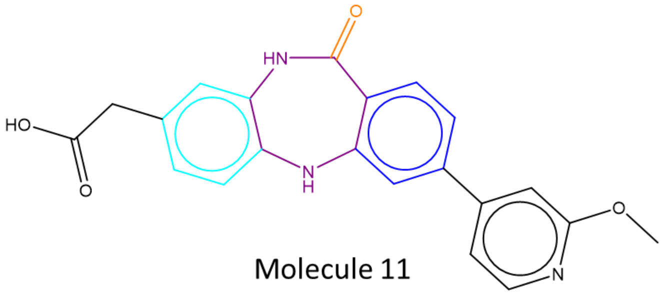
Molecule 8



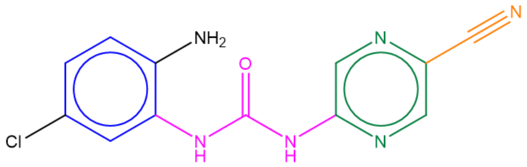
Molecule 9



Molecule 10



Molecule 11



Molecule 12

Figure 6-19: Molecules 7- 12 with their highlighted RG core mapping

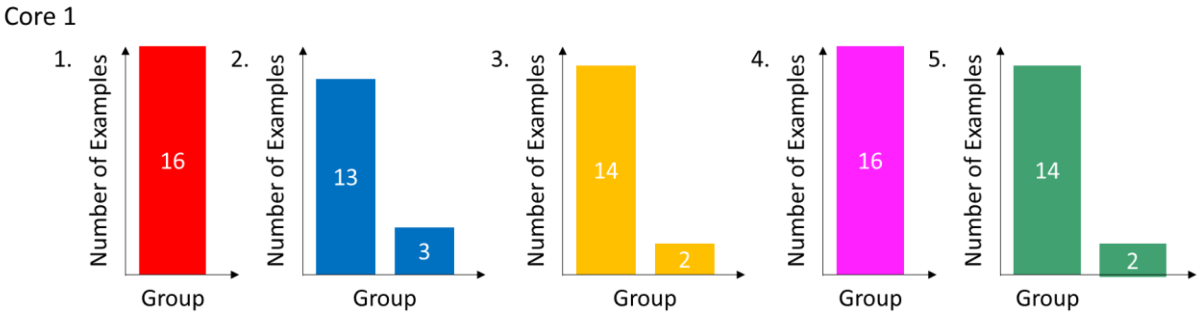


Figure 6-20: Core 1 nodes distributions

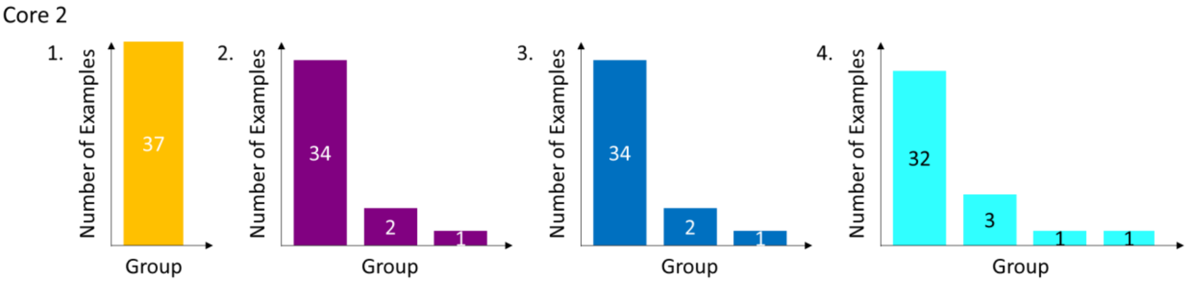


Figure 6-21: Core 2 nodes distributions

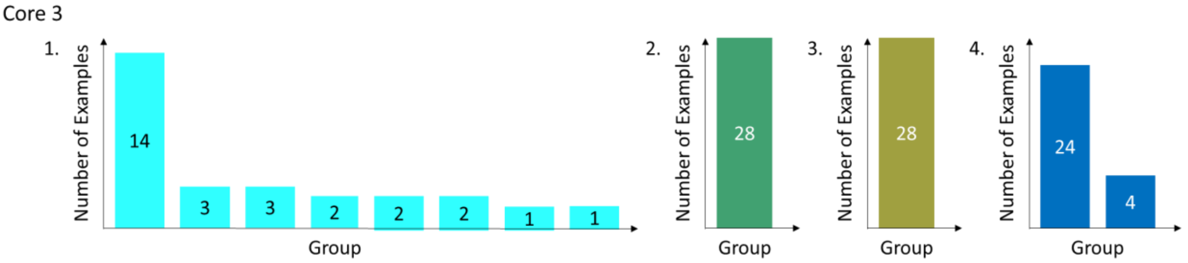


Figure 6-22: Core 3 nodes distributions

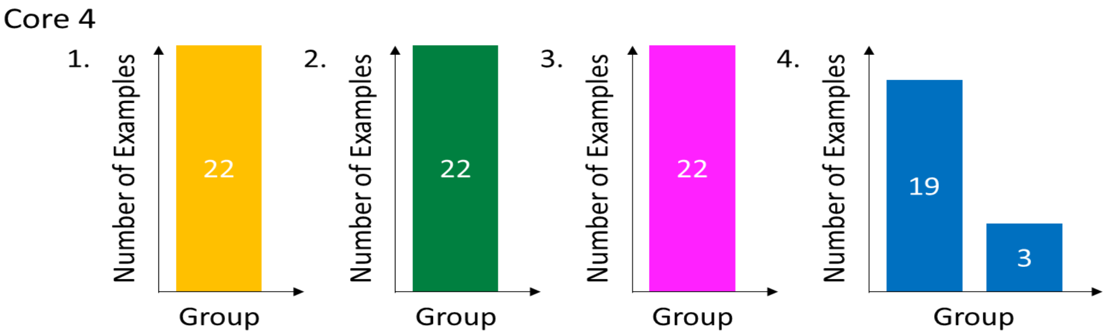


Figure 6-23: Core 4 nodes distributions

To determine the expected exploration score order, pairs of molecules were first inspected.

Initially, molecule 7 (Figure 6-24) and molecule 8 (Figure 6-25) were compared. These molecules map to core 1 and 2, respectively, and the substructures are coloured and annotated accordingly. Molecule 7 has a substructure (green) that is not present in the core; two substructures (red and purple) where there is no variation in the core; and two substructures (orange and blue) with a high number of previously seen examples. Molecule 8 has one substructure (cyan) that has a low number of seen examples; two substructures (purple and blue) that have a large number of previously seen examples; and one substructure (orange) for which there is no variation. Molecule 7 should have a slightly higher exploration score. This is because an entirely new substructure is added for one node, but the rest of the nodes have similar proportions. However, molecule 7 has two fully exploited nodes compared to just the one with molecule 8. Therefore, the order of these two molecules could be reversed or equal.

The next two molecules to be compared are molecules 8 and 9. Molecule 9 can be seen in Figure 6-26 and matches to core 3. Molecule 9 has one substructure with a low number of examples; one substructure which maps to half of the examples in the core; and two substructures where there is no variation. Ignoring the nodes where there is no variation, molecule 9 should therefore have one substructure which has a high node score and another with a medium score, compared to molecule 8 which should have one node that scores highly and three low scoring nodes. Overall, this suggests that molecule 9 should have a higher exploration score than molecule 8.

Comparing molecule 7 and molecule 9, as molecule 7 is expected to score similar if not slightly higher than molecule 8, molecule 9 would be expected to generate a higher exploration score than molecule 7. Even though molecule 7 has a new substructure, this should not outweigh the other factors.

Comparing molecule 8 and molecule 10, molecule 10 (Figure 6-27) has one substructure that is present in half of the examples in the core 3. Another substructure has many examples, and the final two substructures match nodes where there is no variation. Consequently, molecule 8 should generate a higher exploration score than molecule 10.

Molecule 10 and molecule 11 were the next pair to be compared. Molecule 11 is shown in Figure 6-28 and maps to core 2. Molecule 11 has three substructures that have a very high number of previously seen examples and one substructure where there is no variation. Molecule 10 should have a higher exploration score of these two molecules.

Finally, molecule 11 and molecule 12 are compared. Molecule 12 is demonstrated in Figure 6-29 and represents core 4. Molecule 12 has three substructures where there is no variation and one substructure that has a very high number of previously seen examples. So molecule 12 should have the lowest exploration score.

Ultimately, the molecule should, therefore, be ordered as follows: molecule 9 > molecule 7 ≥ molecule 8 > molecule 10 > molecule 11 > molecule 12.

The nodes are highlighted whilst displaying the number of examples currently with that substructure and within the parentheses the number of substructures for that node. For each molecule, the node breakdown and combinations for the collection model score E5 variations are shown in Table 6-15, Table 6-16, Table 6-17, Table 6-18, Table 6-19 and Table 6-20. All other scores can be found in the Appendix.

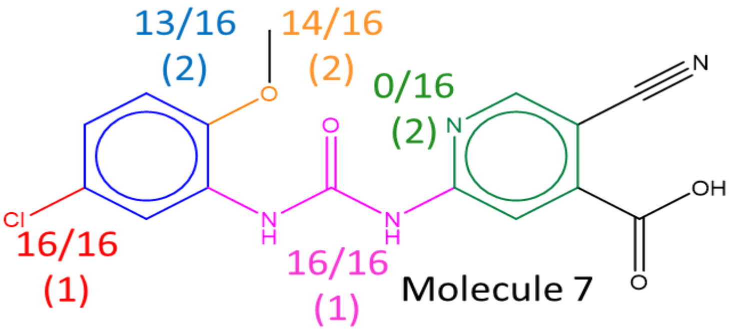


Figure 6-24: Molecule 7, Chk1N34

Table 6-15: Node breakdown for Collection Model score Variations for Molecule 7

Node	**E5**
Red	0
Blue	0.035
Orange	0.016
Pink	0
Green	0.892
Molecule Score	
Total Summed	0.943
Total Multiplied	0
Total Mean	0.189

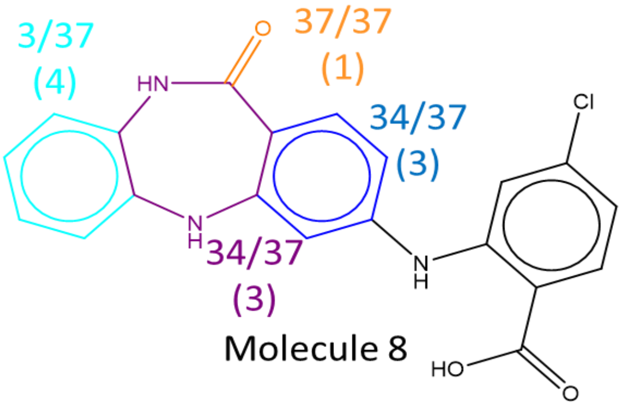


Figure 6-25: Molecule 8, Chk1N35:

Table 6-16: Node breakdown for Collection Model score Variations for Molecule 8

Node	**E5**
Orange	0
Purple	0.005
Blue	0.005
Cyan	0.797
Molecule Score	
Total Summed	0.807
Total Multiplied	0
Total Mean	0.202

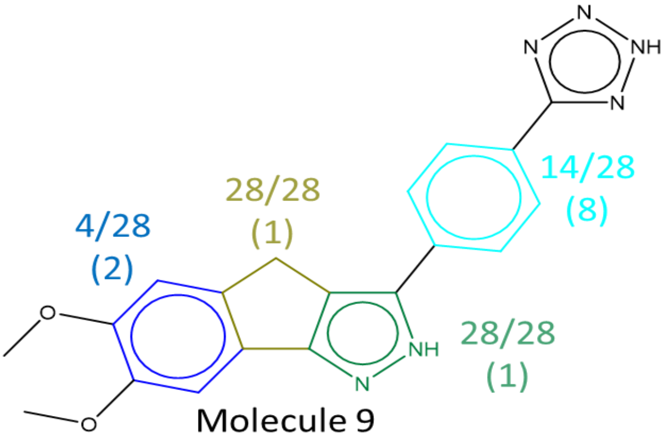


Figure 6-26: Molecule 9, Chk1N123

Table 6-17: Node breakdown for Collection Model score Variations for Molecule 9

Node	**E5**
Cyan	0.146
Green	0
Olive	0
Blue	0.735
Molecule Score	
Total Summed	0.880
Total Multiplied	0
Total Mean	0.220

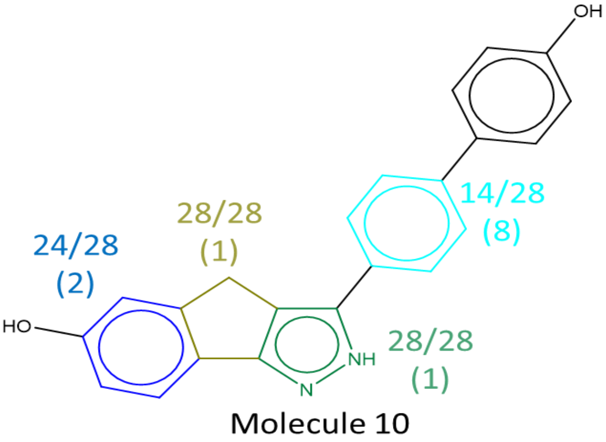


Figure 6-27: Molecule 10, Chk1N95

Table 6-18: Node breakdown for Collection Model score Variations for Molecule 10

Node	**E5**
Cyan	0.146
Green	0
Olive	0
Blue	0.020
Molecule Score	
Total Summed	0.166
Total Multiplied	0
Total Mean	0.042

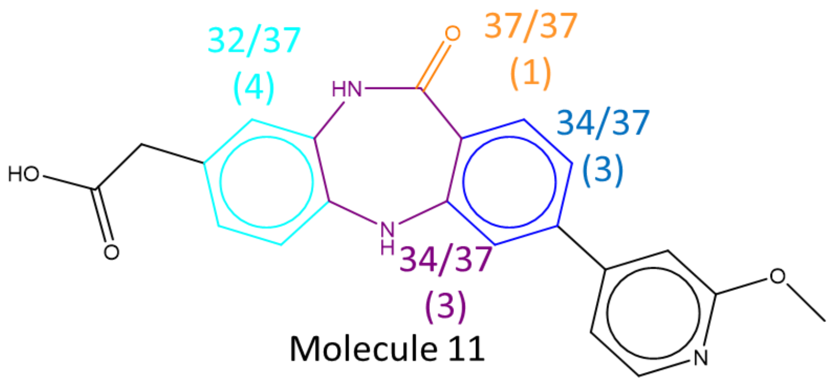


Figure 6-28: Molecule 11, Chk1N65

Table 6-19: Node breakdown for Collection Model score Variations for Molecule 11

Node	**E5**
Orange	0
Purple	0.005
Blue	0.005
Cyan	0.013
Molecule Score	
Total Summed	0.023
Total Multiplied	0
Total Mean	0.006

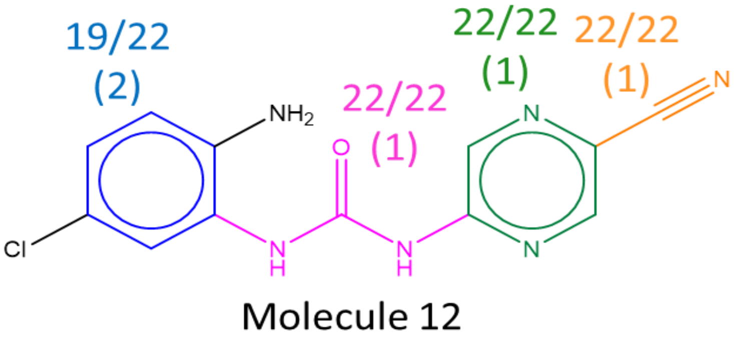


Figure 6-29: Molecule 12, Chk1N13

Table 6-20: Node breakdown for Collection Model score Variations for Molecule 12

Node	**E5**
Orange	0
Green	0
Pink	0
Blue	0.019
Molecule Score	
Total Summed	0.019
Total Multiplied	0
Total Mean	0.005

Table 6-21: Displaying all the overall scores together for molecules within the different cores

		Molecule 7	Molecule 8	Molecule 9	Molecule 10	Molecule 11	Molecule 12
1-Prior Prob	**Total Summed**	1.313	1.081	1.357	0.643	0.297	0.136
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.263	0.270	0.339	0.161	0.074	0.034
Change in Entropy	**Total Summed**	0.170	0.035	0.017	-0.042	-0.024	-0.011
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.034	0.009	0.004	-0.011	-0.006	-0.003
KL Divergence	**Total Summed**	0.059	0.003	0.004	0.0008	0.0002	0.0002
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.012	0.0008	0.001	0.0002	4.18E-05	5.00E-05
E1	**Total Summed**	0.307	0.103	0.074	0.002	7.12E-06	0.0007
	Total Multiplied	1.76E-12	1.51E-19	2.79E-14	2.22E-17	4.86E-24	5.52E-16
	Total Mean	0.061	0.026	0.019	0.0005	1.78E-06	0.0002
E2	**Total Summed**	0.307	0.103	0.074	0.002	7.12E-06	0.0007
	Total Multiplied	1.76E-12	1.51E-19	2.79E-14	2.23E-17	4.86E-24	5.52E-16
	Total Mean	0.061	0.026	0.019	0.0005	1.80E-06	0.0002
E3	**Total Summed**	0.003	0.001	0.001	-0.001	-0.0007	-0.0005
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.0007	0.0004	0.0003	-0.0003	-0.0002	-0.0001
E4	**Total Summed**	0.0004	7.11E-05	0.0001	2.51E-05	2.06E-06	4.47E-06
	Total Multiplied	0	0	0	0	0	0
	Total Mean	8.29E-05	1.78E-05	3.33E-05	6.28E-06	5.15E-07	1.12E-06
E5	**Total Summed**	0.943	0.807	0.880	0.166	0.023	0.019
	Total Multiplied	0	0	0	0	0	0
	Total Mean	0.189	0.202	0.220	0.042	0.006	0.005

The expected ordering of the molecules in this section and Table 6-21 is: molecule 9 > molecule 7 ≥ molecule 8 > molecule 10 > molecule 11 > molecule 12. The orderings observed for 1-Prior Prob, change in entropy, KL divergence and each E score for the summed, multiplied and mean combinations are as a follows. The orders that are the same have been combined.

1-Prior Probability summed: 9 > 7 > 8 > 10 > 11 > 12. Mean: 9 > 8 > 7 > 10 > 11 > 12.

Change in Entropy summed and mean: 7 > 8 > 9 > 12 > 11 > 10.

Kl divergence summed: 7 > 9 > 8 > 10 > 11 = 12. Mean: 7 > 9 > 8 > 10 > 12 > 11.

E1 summed and mean: 7 > 8 > 9 > 10 > 12 > 11. Multiplied: 7 > 9 > 12 > 10 > 8 > 11.

E2: summed and mean: 7 > 8 > 9 > 10 > 12 > 11. Multiplied: 7 > 9 > 12 > 10 > 8 > 11.

E3 summed: 7 > 8 = 9 > 12 > 11 > 10. Mean: 7 > 8 > 9 > 12 > 11 > 10.

E4 summed and mean: 7 > 9 > 8 >10 > 12 > 11.

E5 summed: 7 > 9 > 8 > 10 > 11 > 12. Mean: 9 > 8 > 7 > 10 > 11 > 12.

When multiplying the nodes together, the same issue occurred as previously that when one node was zero it affected the score as a whole. The method that reproduces the expected ordering is the node scoring method 1-Prior Prob with the node scores summed to give the molecule score. However, 1-Prior Prob mean and E5 mean produce orderings that are close to the expected ordering, as only molecules 7 and 8 are reversed and these could be considered close in terms of exploration. The other methods, do not give the correct ordering as they score molecule 7 higher than molecule 9 this is due to them placing more emphasis on new substructures.

From examining all these different scenarios E5 is the only method that across all three experiments generates the desired ordering or is close. It also has the desired properties of generating a score being zero and one, whilst also allowing the method to decipher between different distributions. The most appropriate way of combining is through the mean as this also provided the desired or close to order. By using the mean it also allows better cross-core comparison as otherwise RG cores with more nodes would naturally score more.

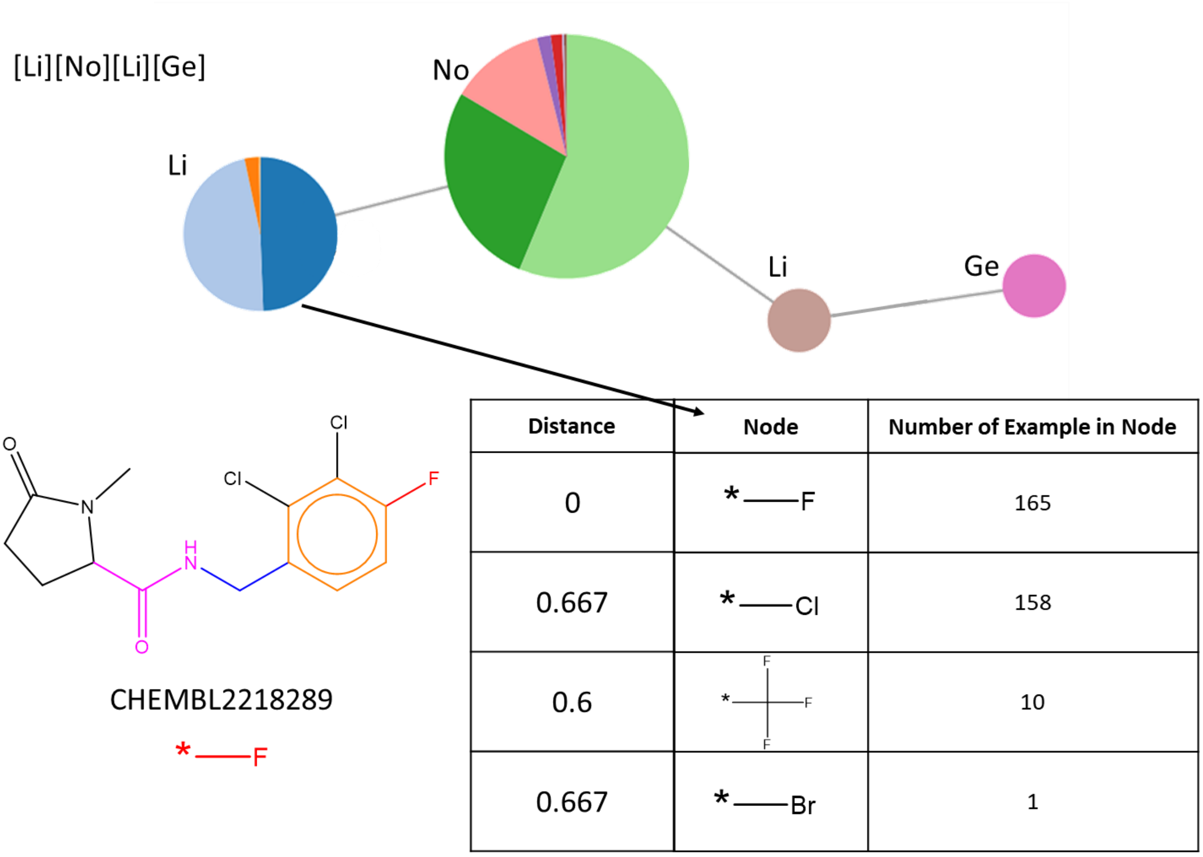
6.3.5 Exploration Score Validation

The exploration score that has been generated needs to be validated to ensure that this score is worthwhile. Unfortunately, there is not an existing method to compare the score to that looks at the level of exploration of chemical space that a molecule adds to. However, it is possible to see if it has advantages over just using Tanimoto distances, by comparing the ranking of molecules from both the exploration score and the Tanimoto distances. There are three levels that are examined: a node level, a core level and a whole molecule level.

All of the molecules from the 10% hold out set are compared to their 90% counterparts. The RG cores that each molecule can map to is identified. If a molecule matches no cores a score of zero is given. If a molecule matches multiple RG cores it is scored for each. The substructures can then be extracted for each of the nodes and the chemical graph that represents the RG core. The attachment points are all retained with changing the connecting atoms to a wild atom, *. The Tanimoto distance can then be calculated, by finding one minus the Tanimoto coefficient using the maximum common substructure, MCS, of the two structures or substructures (Maggiora & Shanmugasundaram, 2004). Where A is the number of atoms in the first molecule or substructure and B is the number of atoms in the second molecule or substructure.

$$Tanimoto\ Distance = 1 - \frac{MCS}{A + B - MCS} \tag{6.19}$$

For the node level, just the substructure extracted from a matched molecule is compared to all the existing substructural fragments for that node. The minimum distance to the nearest substructure is found. If the substructure already exists for that node, then a score of zero is assigned. Therefore, only substructural fragments that have not been seen before provide any score. For this method, the number of already seen examples does not affect this validation score. Figure 6-30 shows an example of how each of the node distances is calculated. This is molecule CHEMBL2218289 that matches to core [Li][No][Li][Ge] from the P2x7 Subset dataset. It can be seen that the node does already exist so a distance of zero is scored. However, if this was not present, the distance would be 0.6. For each node the distance is calculated. To compare to the exploration score, these distances are combined via adding and finding the mean for that core.



Distance	Node	Number of Example in Node
0	*—F	165
0.667	*—Cl	158
0.6	F *—F F	10
0.667	*—Br	1

Figure 6-30: Demonstrating the node distance extraction process

The next level is to examine the molecules on a core level. For this methodology, the whole core SMARTS are compared to the existing core SMARTS. This is done similarly to the node level. The minimum distance between this core SMARTS and the existing core SMARTS becomes this molecules' core score. Figure 6-31 demonstrates how the methodology works on a core level. The core from the molecule being examined is compared to all the existing unique cores present within this dataset. Only a proportion of the cores are shown in this example. As this core is already present a distance of zero is observed.

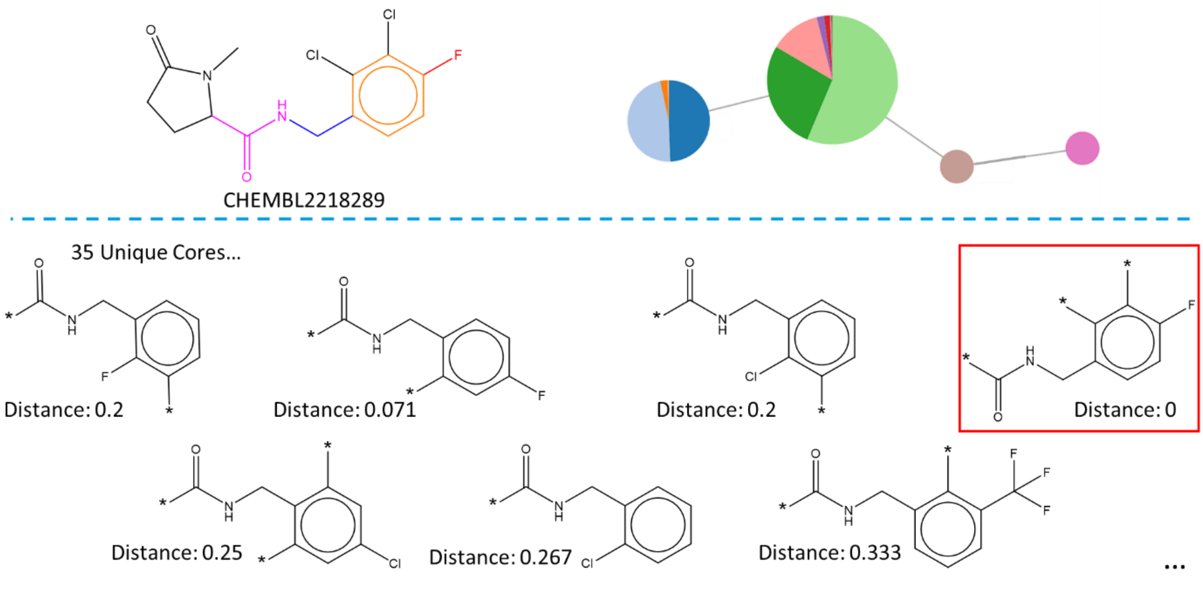


Figure 6-31: Example of core distance for molecule CHEMBL2218289

For the final level, the whole molecule is compared to all existing molecules represented by a core. The molecular level contains extra molecular information than the exploration score explored in this chapter as it contains the additional R-groups and not just the core scaffold. Molecule CHEMBL2218289 has a Tanimoto distance of 0.048. The whole molecule validation is an extra level of complexity than the exploration score generated in this chapter, however, it was done for comprehensiveness.

All of these scores were found for each of the molecules within the extracted 10%. As the rankings of these scores were to be compared to the rankings generated from the exploration score, the fractional ranking were found. These fractional rankings were found for the molecules that matched to each core level and for the whole 10% hold out set. The molecules are ordered in descending order of score. Fractional ranking then takes an average of the indexes if any of the values are the same. For example, Table 6-22 demonstrates how fractional ranking is found. When looking at 0.3 the two indexes it has in a sorted list are 4 and 5, therefore, the fractional ranking would be an average of these indexes.

Table 6-22: Fractional ranking example

Distance Score	0	0.3	0.2	0	0.7	0.4	0.9	0.3	0.7	0.1
Ordered Indexed	8	4	6	9	1	3	0	5	2	7
Fractional Ranking	8.5	4.5	6	8.5	1.5	3	0	4.5	1.5	7

Two statistics are then found. The statistics do not need to include a ranked bias comparison, just a comparison of agreement. Therefore, the Kendall tau and spearman rank coefficient are found. Both look at comparing the ranking within a list. For both *n* is the number of molecules.

$$Kendall\,Tau = \frac{(number\ of\ concordant\,pairs) - (number\ of\ discorant\,pairs)}{\left(\frac{n(n-1)}{2}\right)} \tag{6.20}$$

$$Spearman\,rank\,coefficient = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{6.21}$$

Where d_i is the difference in ranked orders.

6.3.5.1 Whole dataset

All the molecules are compared on a whole datasets level for all of the datasets. The comparison is drawn on two levels on a core level, where the average of these scores is taken and then on a whole dataset level. Table 6-23 show the results from the E5 Mean exploration score, p-values are recorded in brackets.

Table 6-23: E5 Mean Statistical Ranking Comparison

E5 Mean		Node Distances		Core Distances		Molecular Distance	
		Kendall	Spearman	Kendall	Spearman	Kendall	Spearman
Bajorath	Avg Core	0.619	0.640	0.487	0.534	0.171	0.203
	Dataset	0.141 (0.005)	0.170 (0.005)	0.258 (2.451e^{-7})	0.313 (1.22e^{-7})	0.019 (0.648)	0.037 (0.544)
CDK2	Avg Core	0.505	0.521	0.357	0.393	0.070	0.063
	Dataset	0.257 (1.66e^{-6})	0.315 (1.26e^{-6})	0.379 (2.10e^{-13})	0.480 (1.72e^{-14})	0.119 (0.009)	0.179 (0.007)
Chk1	Avg Core	1	1	1	1	0.825	0.864
	Dataset	0.461 (0.053)	0.523 (0.055)	0.461 (0.053)	0.523 (0.055)	0.333 (0.116)	0.483 (0.080)
Cyto	Avg Core	0.418	0.465	0.409	0.469	0.168	0.207
	Dataset	0.212 (1.68e^{-72})	0.262 (4.46e^{-74})	0.240 (3.09e^{-95})	0.301 (9.80e^{-99})	0.169 (6.27e^{-66})	0.251 (6.87e^{-68})
FactorXa	Avg Core	0.429	0.466	0.473	0.527	0.117	0.144

	Dataset	0.275 (9.50e^{-12})	0.341 (2.53e^{-12})	0.341 (4.20e^{-18})	0.433 (1.17e^{-19})	0.087 (0.012)	0.127 (0.011)
Neurokinin	Avg Core	0.401	0.439	0.429	0.478	-0.073	-0.040
	Dataset	0.290 (9.21e^{-9})	0.356 (3.87e^{-9})	0.430 (1.72e^{-18})	0.534 (2.03e^{-20})	0.145 (0.001)	0.208 (0.001)
P2x7	Avg Core	0.449	0.472	0.435	0.489	0.051	0.058
	Dataset	0.069 (0.094)	0.084 (0.095)	0.246 (9.05e^{-10})	0.304 (7.19e^{-10})	0.018 (0.592)	0.029 (0.562)
P2x7 Subset	Avg Core	0.524	0.576	0.642	0.670	0.072	0.102
	Dataset	0.090 (0.213)	0.106 (0.223)	0.199 (0.006)	0.237 (0.006)	0.118 (0.052)	0.170 (0.050)
P38a	Avg Core	0.374	0.413	0.328	0.372	0.156	0.191
	Dataset	0.209 (5.66e^{-15})	0.257 (3.26e^{-15})	0.255 (4.81e^{-23})	0.323 (1.30e^{-23})	0.134 (2.30e^{-9})	0.196 (2.44e^{-9})

Examining the results in Table 6-23 shows that very few datasets show cohesion between the exploration score and the distance as very few scores are one or close to one. The one dataset that generates the same ranking is dataset Chk1 and this is because there are very few molecules within the 10 percent, eleven molecules in fact. Four molecules match to two cores, therefore, there are only fourteen exploration scores to compare. Out of the fourteen only two provide a new substructure for one node, and these two cores had not been seen before. Therefore, it is very easy for the two ranked list to be the same. These scores in the majority of cases show the advantages of using the number of previously seen number of examples and it allows methods to be more discriminative.

6.4 Conclusion

Extensive work has been carried out to develop an exploration score. The exploration score should reflect the amount of information that would be added to a lead optimisation series by the addition of a new compound. It therefore should reflect the information being added into the system compared to how much current information is within the system. The score should be scaled between zero and one to allow to distinguish between a high and low score easily. Also, the score should be able to differentiate between different underlying distributions of substructures represented by the nodes in a reduced graph core. Finally, work was done to identify how best to combine the node scores for an overall molecule score.

Several scores have been developed from different theories. The different scenarios were manually ranked and assessed the scoring methods against these manual rankings, whilst understanding if these methods contained the desired properties. The majority of the scores had several advantages and disadvantages. Two scores seemed to have potential and were promising as they possessed all the sought after properties, E2 and E5. These two scores were promising because they were scaled between zero and one, they distinguish between different underlying distributions and between these distributions they reproduce the correct ordering.

All the scores were then further explored to identify if any of the scores generate the expected manual ordering. Several different experiments were constructed to evaluate them and to see if any consistently performed as hoped. None of the scores for all of the experiments got the desired ordering, however, some were close to the expected ordering, and there were justifications on why they could have got a slight variation to the desired results. For the simple node experiment, none of the variations exhibited the desired ordering. However, E2 and E5 nearly generate the expected ordering other than two rows being in a different order and this was down to the fact these two scores try and prevent the constant promotion of a new substructure being added. When a core was constructed of three nodes then two different techniques did generate the correct score. This experiment analysed how the node scores worked along with how best to combine the node scores. The two techniques that generated the correct score is E3 mean and E4 mean. Two scores generated nearly the expected ordering was E5 sum and mean.

The last two experiments were based upon real examples from P2x7 Subset dataset. When examining the molecules that came from the same core several scoring methods gave the desired output, 1-Prior prob sum and mean, KL divergence sum and mean, E3 mean, E4 mean and E5 sum. Four scores were close to generating this ordering, E3 sum, E4 sum and E5 mean, as they ranked two molecules equal instead of one being superior. When examining the molecules' results from different cores only one molecule gives the desired result, 1-Prior prob sum. However, two other scores nearly generate the correct results, 1-Prior prob mean and E5 mean.

After reviewing all of the experiments resulting and the desired properties that were hoped for within an exploration score, it was felt that the most appropriate score to use was E5 mean. E5 mean is the favoured score because it had all of the desired properties and in most instances, either gave the desired ordering or was close. If the ordering was not obtained, it was close and had valid reasons as to why this had been achieved and was not unreasonable.

7 Applying Reduced Graphs For Molecular Exploitation

7.1 Introduction

The Lead optimisation (LO) process is concerned with developing known active molecules to further improve absorption, distribution, metabolism, excretion and toxicity (AMDET) and potency properties. As stated previously, medicinal chemists adapt substituents on a core or scaffold that is shared by several molecules that are known to be active. Chemists typically adopt one of two strategies during lead optimisation: exploitation or exploration of the chemical space occupied by the compounds synthesised so far. In an exploitation strategy, they typically want to work in the areas of the chemical space that the most active compounds found so far occupy in order to improve potency.

The aim of the previous chapter was to assign an exploration score to a new molecule by comparing its RG representation with those of molecules already in the LO series. This was achieved by mapping the molecule to an RG core in the series and carrying out a node-by-node comparison. For each node, the substructure presented by the new molecule was compared with the frequency distribution of substructures in the RG core. A high exploration score was assigned to a substructure with low frequency in the RG core and vice versa.

The research undertaken within this chapter aims to assign an exploitation score to new molecules according to the extent to which they exploit existing knowledge on activity. First, substructure significance values are calculated for each substructure at each node position in the RG core based on the activity values of the molecules represented by the core. The substructure significance values are calculated using a method that is similar to the fragment significance method developed by Polishchuk et al. (Matveieva, Cronin, & Polishchuk, 2019; P. G. Polishchuk, Kuźmin, Artemenko, & Muratov, 2013; P. Polishchuk et al., 2016). An exploitation score for a new molecule is determined by mapping the new molecule onto a RG core and retrieving substructure significance scores based on the mapping.

The chapter first introduces the Polishchuk et al. method and then describes how it has been adapted in this work. Results are then presented based on the MMP12 dataset.

7.2 Methodology

The exploitation score is based on work by Polishchuk (Matveieva et al., 2019; P. G. Polishchuk et al., 2013; P. Polishchuk et al., 2016). Polishchuk et al. developed an approach to assess the contributions of specific fragments to a property of interest based on an existing QSAR model. A fragment contribution is assessed by first applying the model to calculate a predicted value for a molecule containing the fragment. The fragment is then removed from the molecule and a new predicted value is calculated. The new predicted value is then subtracted from the whole molecule predicted value to give the fragment contribution. This value is considered as a local contribution of that fragment to the property of interest. A global measure for a fragment is calculated by repeating the process for all molecules containing the given fragment and averaging the local fragment contributions. If the global fragment contribution is positive then the fragment is favourable with respect to the property modelled by the QSAR, otherwise, if the contribution is negative then the chemist should consider replacing the fragment either by one that has not been seen before or by one that contributes a positive amount to the property.

Polishchuk et al. investigated the effectiveness of their approach using different machine learning methods including random forest (RF), gradient boost model (GBM), partial least squares (PLS) and support vector machines (SVM) and using two types of molecular descriptors: the two dimensional simplex representation of molecular structure (SiRMS) descriptors; and Dragon descriptors ("Dragon," n.d.; V. E. Kuźmin, Artemenko, & Muratov, 2008; Victor E. Kuźmin et al., 2005; Mauri, Consonni, Pavan, & Todeschini, 2006). For each descriptor, average fragment significance values were calculated using each machine learning method and a consensus value was taken (P. G. Polishchuk et al., 2013; P. Polishchuk et al., 2016).

In the follow up paper, Matveieva examined how the environment of a fragment can have an impact on its significance. The distribution of fragment contribution scores was examined for a specific fragment, and if the distribution contained several peaks or one broad peak then the variance of the score was high suggesting that the environment of the fragment should be considered. This was done by using a Gaussian mixture model to separate the peaks. These peaks were then analysed using SMARTSminer to generate a SMARTS pattern that represents the fragment and surrounding environment (Bietz, Schomburg, Hilbig, & Rarey, 2015; Matveieva et al., 2019).

An outline of the methodology developed here is shown in Figure 7-1. First a QSAR model is developed for the LO series. Then each RG core is considered in turn and a significance score is calculated for each substructure for each node as follows. Each substructure of each node is considered in turn. A molecule containing the substructure at that node position is retrieved and a local substructure significance score is calculated. The QSAR model is used to predict the score for the molecule then the substructure is removed from the molecule and the QSAR model is applied again. The local substructure significance score is calculated as the difference in predicted scores with and without the fragment. A global score is then calculated by combining the local scores for all molecules that contain the substructure at that node position.

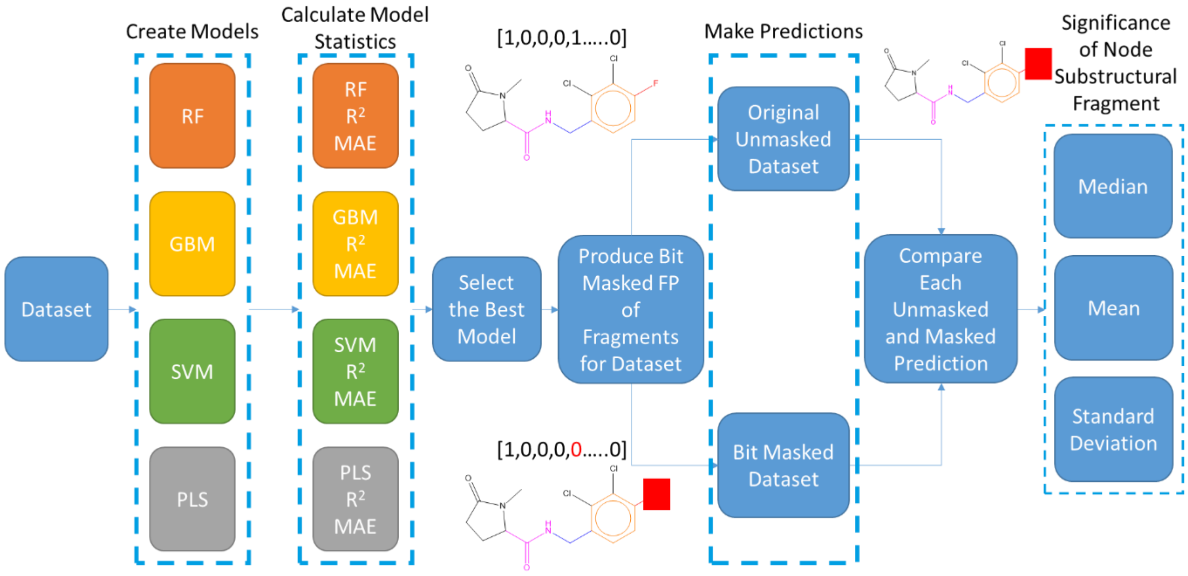


Figure 7-1: Workflow of RG core node significance

As indicated in the workflow in Figure 7-1, a number of alternatives were explored for each step. These include different QSAR methods and different ways of masking the fragments in order to derive predictions when the fragment of interest is removed from a molecule. These investigations are described in detail below with molecule CHEMBL2218289 from the P2x7 subset and P2x7 dataset used as an exemplar.

7.2.1 Creation of Machine Learning Models

Four machine learning methods were used to calculate the QSAR models. The biological activity, pIC50, is the target value of interest and the molecular descriptors were Morgan radius 2 (M2FP). The QSAR models were generated using the python module scikit-learn (Pedregosa et al., 2011). In each case, a ten-fold cross validation was undertaken to assess model performance. The cross validation was implemented using scikit-learn's model selection KFold function. The best model was that with the highest mean 10-fold cross validation R^2 score. The model was then recreated using the whole dataset, as the point of this study is to exploit the data within a dataset and not predict new molecules. Additionally, as no prediction of new molecules is occurring no further hyperparameter optimisation of the model occurs, such as changing the number of trees. The mean absolute error (MAE), R^2, root-mean squared error (RMSE) and mean squared error (MSE) are all calculated.

Both the RF and GBM models were created using the default settings of one-hundred estimators and a fixed random state (of 42) so that the results are reproducible. The SVM model was generated using default settings with the kernel set to radial basis function (rbf). The PLS model was generated using default settings with number of components equal to two.

7.2.2 Producing Bit Masked FP

The input to the QSAR method is the M2FP representation of a molecule. As above, and following the method of Polishchuk et al., the predicted activity is first calculated for the whole molecule and a predicted value is then calculated for the molecule without the substructure (P. G. Polishchuk et al., 2013). The latter requires that the M2FP is modified to represent the molecule with the substructure removed. Four different masking approaches were considered for modifying the M2FP fingerprint. The first two methods involved modifying the molecule and then recalculating the M2FP. The molecule was modified by: removing the fragment from the molecule; and altering the fragment atoms to wild atoms. The second two methods involved manipulating the fingerprint directly.

7.2.2.1 Removal of Fragments

The substructure, or fragment, of interest was removed from the molecule by deleting the atoms in the fragment along with any bonds incident on those atoms. The M2FP was then recalculated. An example of how this was achieved for a molecule is shown in Figure 7-2.

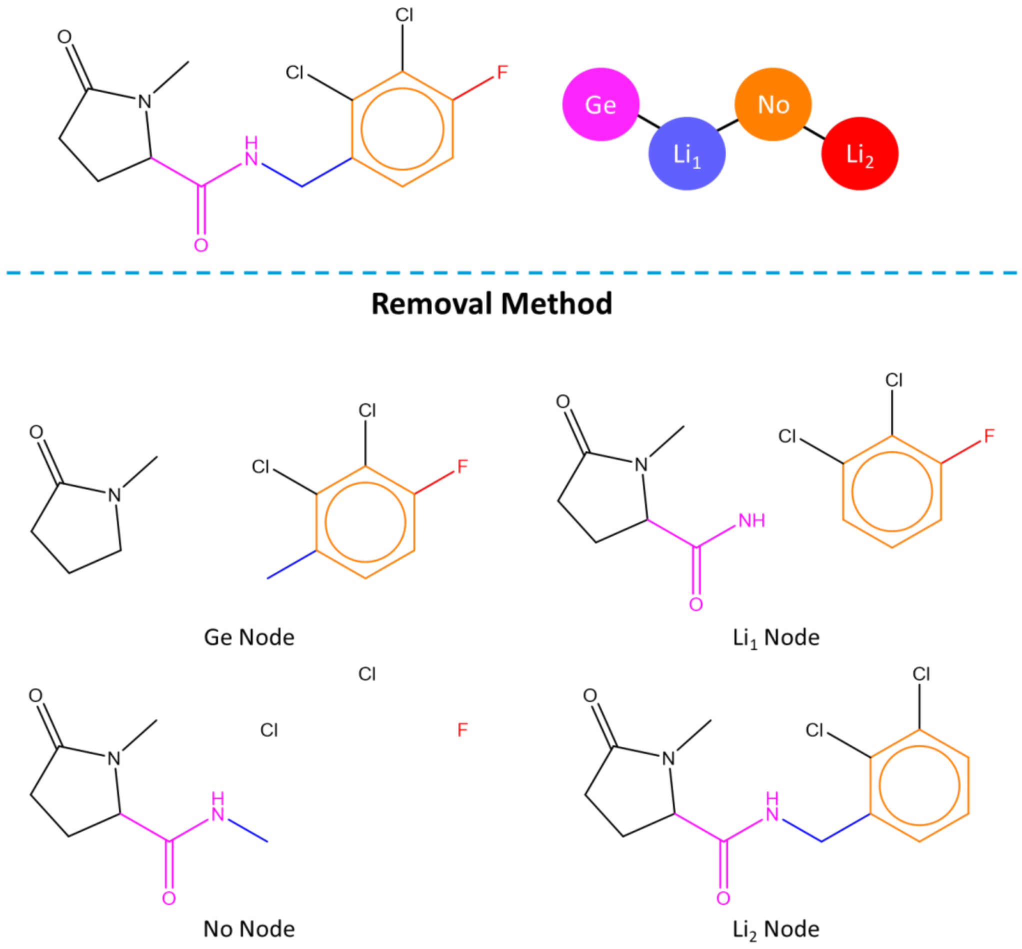


Figure 7-2: Example of how the removal method works for each RG core node within a molecule

7.2.2.2 Swapping to Wild Atoms

The second method replaced the substructural atoms of the fragment with wild atoms instead of removing the atoms and bonds. This helps to retain the molecular framework given the circular nature of the fingerprints while removing specific atom types. An example of the wild atom replacement is shown in Figure 7-3.

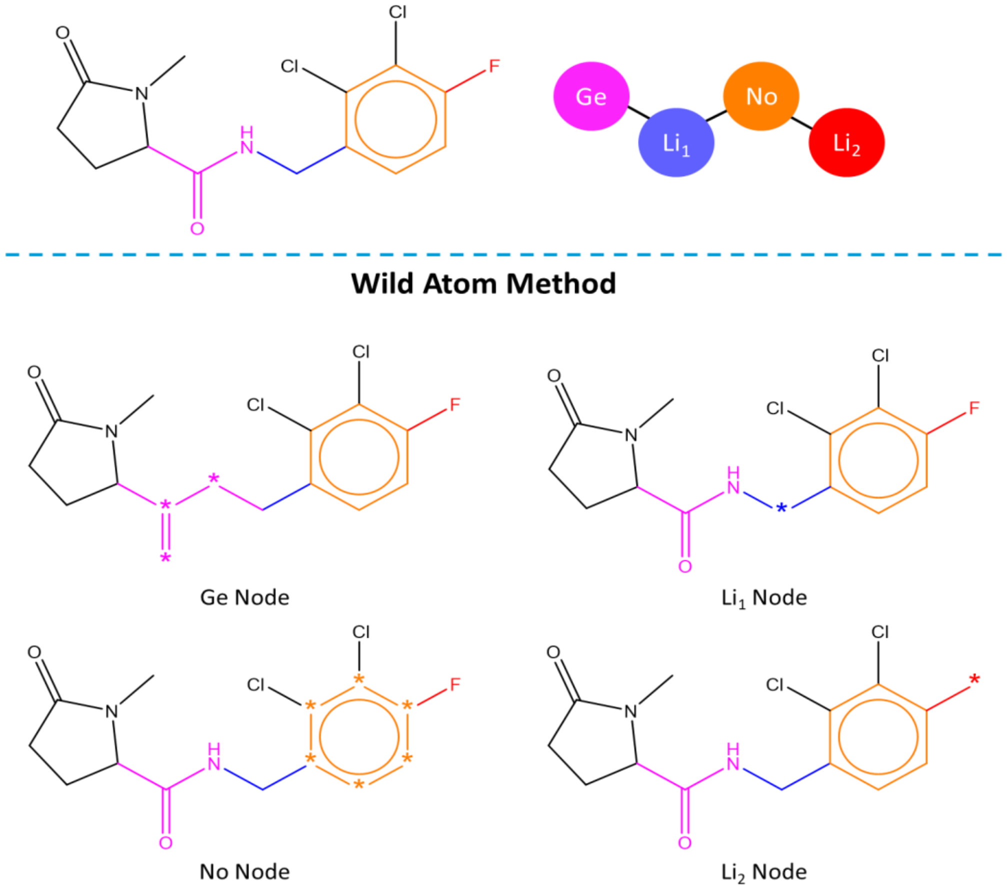
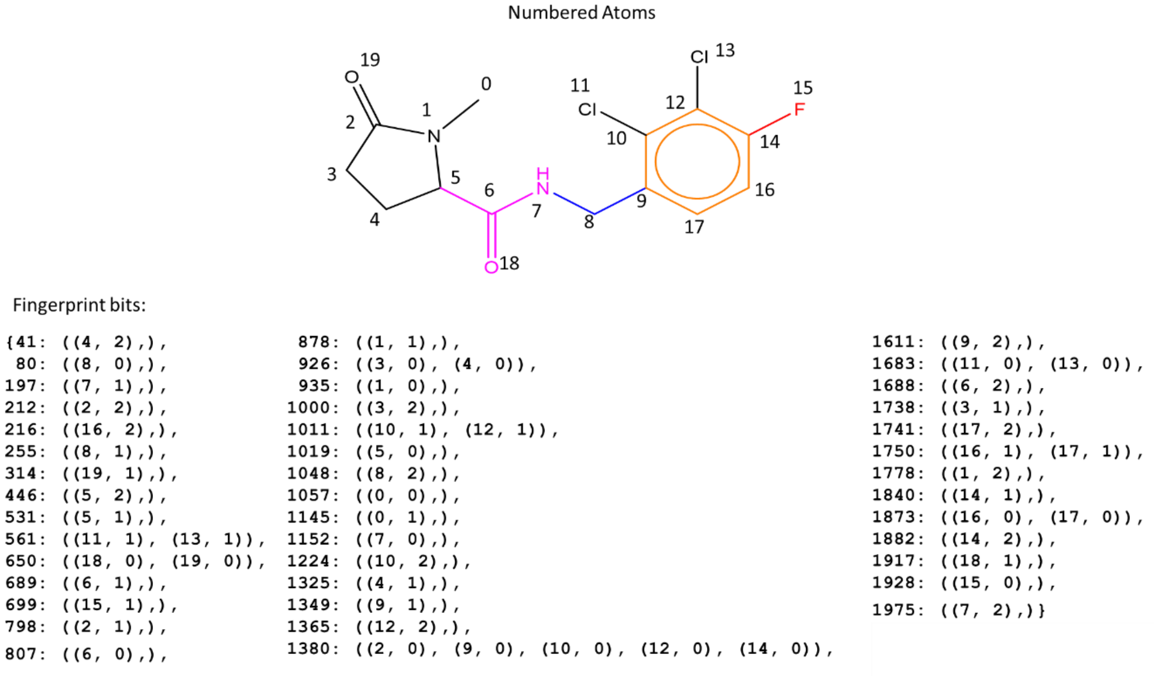


Figure 7-3: Example of how the wild atom method works for each RG core node within a molecule

7.2.2.3 Bit Masking

A dictionary can be generated when creating the M2FP fingerprint in RDKit that aligns each bit within the fingerprint to one or more tuples, where the tuple is of length 2 ("RDKit: Open-Source Chemoinformatics," 2018). The first number within a tuple is the atom index that the bit is centred upon and the second number is the radius that is represented, e.g. 0, 1 or 2 for M2FP. If the substructure is repeated with a molecule then the bit is associated with multiple tuples, one for each occurrence of the substructure.



```
{41: ((4, 2),),           878: ((1, 1),),                                   1611: ((9, 2),),
 80: ((8, 0),),           926: ((3, 0), (4, 0)),                            1683: ((11, 0), (13, 0)),
197: ((7, 1),),           935: ((1, 0),),                                   1688: ((6, 2),),
212: ((2, 2),),          1000: ((3, 2),),                                   1738: ((3, 1),),
216: ((16, 2),),         1011: ((10, 1), (12, 1)),                          1741: ((17, 2),),
255: ((8, 1),),          1019: ((5, 0),),                                   1750: ((16, 1), (17, 1)),
314: ((19, 1),),         1048: ((8, 2),),                                   1778: ((1, 2),),
446: ((5, 2),),          1057: ((0, 0),),                                   1840: ((14, 1),),
531: ((5, 1),),          1145: ((0, 1),),                                   1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)), 1152: ((7, 0),),                                   1882: ((14, 2),),
650: ((18, 0), (19, 0)), 1224: ((10, 2),),                                  1917: ((18, 1),),
689: ((6, 1),),          1325: ((4, 1),),                                   1928: ((15, 0),),
699: ((15, 1),),         1349: ((9, 1),),                                   1975: ((7, 2),)}
798: ((2, 1),),          1365: ((12, 2),),
807: ((6, 0),),          1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Figure 7-4: Molecule CHEMBL2218289 showing the fingerprint bits present within M2FP

Figure 7-4 shows the fingerprint bits present in the M2FP for molecule CHEMBL2218289. The M2FP is a binary vector fingerprint so that a bit is set to 1 if there is at least one tuple present. Bits 561, 650, 926, 1011, 1380, 1683, 1750 and 1873 are all examples where multiple substructures generate the same bit. Figure 7-5 indicates four examples of bits that are present within the M2FP of molecule CHEMBL2218289. Bit 80 has a radius of zero and represents an aliphatic carbon atom with two substitution sites and two hydrogens. Bit 197 has a radius of one, and represents a secondary nitrogen with two connecting carbon atoms, where one of the carbons has another substitution site and the other has two substitution sites one of which is a single bond and the other is a double bond. Bit 216 has a radius of two, and is a carbon atom with a fluorine atom connected at radius 2. Bit 561 represents a chlorine atom connected to a carbon and there are two occurrences in the molecule.

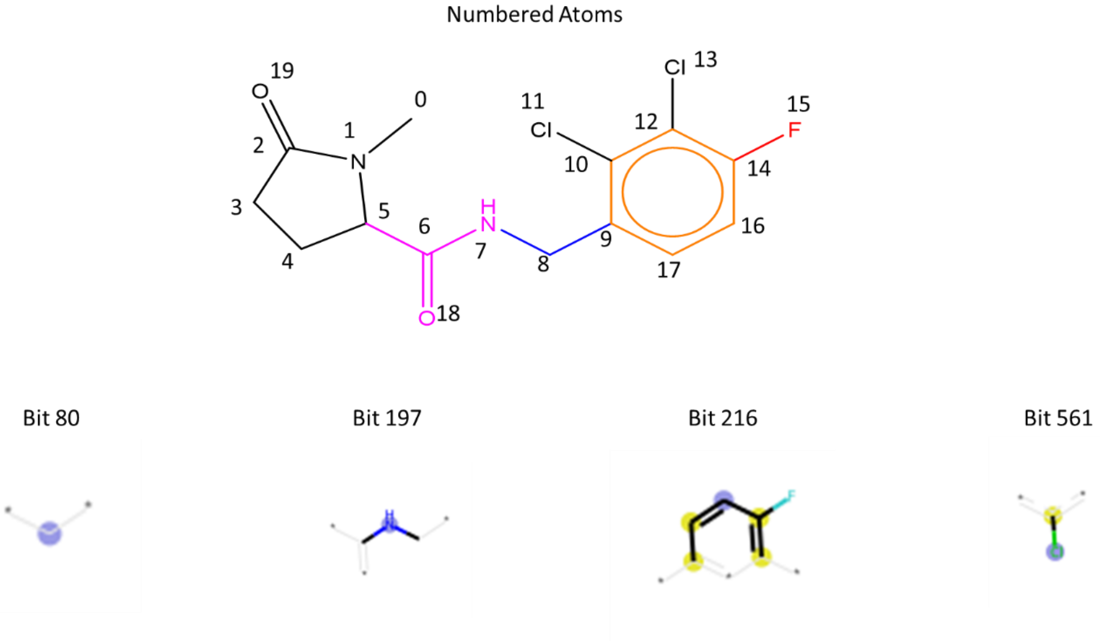
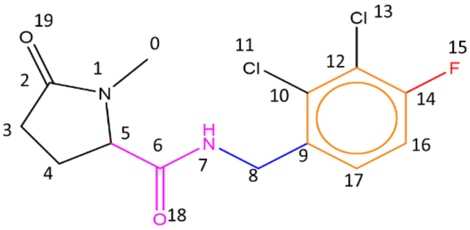


Figure 7-5: Bits example

There are several ways in which the bits could potentially be masked, for example, a bit can be masked based upon just the central atom or on all the atoms incorporated in the radius. In the first method, referred to as the central atom method, all of the tuples that represent atoms in the substructure of interest are removed. When this results in all of the tuples for a given bit being removed then the bit is removed (masked) from the fingerprint. An example is shown in Figure 7-6 where the different colours correspond to the tuples that would be removed for each of the correspondingly coloured substructures. When this process results in bits being removed/masked these are also highlighted. Two instances where bits remain despite tuples being removed are Ge bit 650 and No bit 1380.

Numbered Atoms



Central Atom Method

Fingerprint bits:

Ge

```
{41: ((4, 2),),            878: ((1, 1),),                                     1611: ((9, 2),),
 80: ((8, 0),),            926: ((3, 0), (4, 0)),                              1683: ((11, 0), (13, 0)),
197: ((7, 1),),            935: ((1, 0),),                                     1688: ((6, 2),),
212: ((2, 2),),           1000: ((3, 2),),                                     1738: ((3, 1),),
216: ((16, 2),),          1011: ((10, 1), (12, 1)),                            1741: ((17, 2),),
255: ((8, 1),),           1019: ((5, 0),),                                     1750: ((16, 1), (17, 1)),
314: ((19, 1),),          1048: ((8, 2),),                                     1778: ((1, 2),),
446: ((5, 2),),           1057: ((0, 0),),                                     1840: ((14, 1),),
531: ((5, 1),),           1145: ((0, 1),),                                     1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),  1152: ((7, 0),),                                     1882: ((14, 2),),
650: ((18, 0), (19, 0)),  1224: ((10, 2),),                                    1917: ((18, 1),),
689: ((6, 1),),           1325: ((4, 1),),                                     1928: ((15, 0),),
699: ((15, 1),),          1349: ((9, 1),),                                     1975: ((7, 2),)}
798: ((2, 1),),           1365: ((12, 2),),
807: ((6, 0),),           1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Li_1

```
{41: ((4, 2),),            878: ((1, 1),),                                     1611: ((9, 2),),
 80: ((8, 0),),            926: ((3, 0), (4, 0)),                              1683: ((11, 0), (13, 0)),
197: ((7, 1),),            935: ((1, 0),),                                     1688: ((6, 2),),
212: ((2, 2),),           1000: ((3, 2),),                                     1738: ((3, 1),),
216: ((16, 2),),          1011: ((10, 1), (12, 1)),                            1741: ((17, 2),),
255: ((8, 1),),           1019: ((5, 0),),                                     1750: ((16, 1), (17, 1)),
314: ((19, 1),),          1048: ((8, 2),),                                     1778: ((1, 2),),
446: ((5, 2),),           1057: ((0, 0),),                                     1840: ((14, 1),),
531: ((5, 1),),           1145: ((0, 1),),                                     1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),  1152: ((7, 0),),                                     1882: ((14, 2),),
650: ((18, 0), (19, 0)),  1224: ((10, 2),),                                    1917: ((18, 1),),
689: ((6, 1),),           1325: ((4, 1),),                                     1928: ((15, 0),),
699: ((15, 1),),          1349: ((9, 1),),                                     1975: ((7, 2),)}
798: ((2, 1),),           1365: ((12, 2),),
807: ((6, 0),),           1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

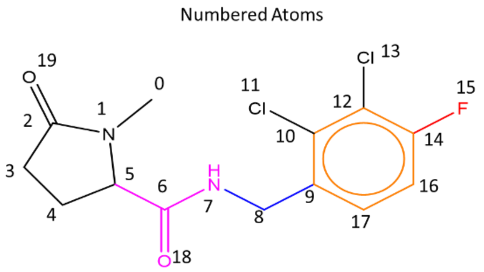
No

```
{41: ((4, 2),),            878: ((1, 1),),                                     1611: ((9, 2),),
 80: ((8, 0),),            926: ((3, 0), (4, 0)),                              1683: ((11, 0), (13, 0)),
197: ((7, 1),),            935: ((1, 0),),                                     1688: ((6, 2),),
212: ((2, 2),),           1000: ((3, 2),),                                     1738: ((3, 1),),
216: ((16, 2),),          1011: ((10, 1), (12, 1)),                            1741: ((17, 2),),
255: ((8, 1),),           1019: ((5, 0),),                                     1750: ((16, 1), (17, 1)),
314: ((19, 1),),          1048: ((8, 2),),                                     1778: ((1, 2),),
446: ((5, 2),),           1057: ((0, 0),),                                     1840: ((14, 1),),
531: ((5, 1),),           1145: ((0, 1),),                                     1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),  1152: ((7, 0),),                                     1882: ((14, 2),),
650: ((18, 0), (19, 0)),  1224: ((10, 2),),                                    1917: ((18, 1),),
689: ((6, 1),),           1325: ((4, 1),),                                     1928: ((15, 0),),
699: ((15, 1),),          1349: ((9, 1),),                                     1975: ((7, 2),)}
798: ((2, 1),),           1365: ((12, 2),),
807: ((6, 0),),           1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Li_2

```
{41: ((4, 2),),            878: ((1, 1),),                                     1611: ((9, 2),),
 80: ((8, 0),),            926: ((3, 0), (4, 0)),                              1683: ((11, 0), (13, 0)),
197: ((7, 1),),            935: ((1, 0),),                                     1688: ((6, 2),),
212: ((2, 2),),           1000: ((3, 2),),                                     1738: ((3, 1),),
216: ((16, 2),),          1011: ((10, 1), (12, 1)),                            1741: ((17, 2),),
255: ((8, 1),),           1019: ((5, 0),),                                     1750: ((16, 1), (17, 1)),
314: ((19, 1),),          1048: ((8, 2),),                                     1778: ((1, 2),),
446: ((5, 2),),           1057: ((0, 0),),                                     1840: ((14, 1),),
531: ((5, 1),),           1145: ((0, 1),),                                     1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),  1152: ((7, 0),),                                     1882: ((14, 2),),
650: ((18, 0), (19, 0)),  1224: ((10, 2),),                                    1917: ((18, 1),),
689: ((6, 1),),           1325: ((4, 1),),                                     1928: ((15, 0),),
699: ((15, 1),),          1349: ((9, 1),),                                     1975: ((7, 2),)}
798: ((2, 1),),           1365: ((12, 2),),
807: ((6, 0),),           1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Figure 7-6: Central atom masking method bits that have been highlighted are the bits that are masked for each node



All Atom Method

Fingerprint bits:

Ge

```
{41: ((4, 2),),              878: ((1, 1),),                                                  1611: ((9, 2),),
 80: ((8, 0),),              926: ((3, 0), (4, 0)),                                           1683: ((11, 0), (13, 0)),
197: ((7, 1),),              935: ((1, 0),),                                                  1688: ((6, 2),),
212: ((2, 2),),             1000: ((3, 2),),                                                  1738: ((3, 1),),
216: ((16, 2),),            1011: ((10, 1), (12, 1)),                                         1741: ((17, 2),),
255: ((8, 1),),             1019: ((5, 0),),                                                  1750: ((16, 1), (17, 1)),
314: ((19, 1),),            1048: ((8, 2),),                                                  1778: ((1, 2),),
446: ((5, 2),),             1057: ((0, 0),),                                                  1840: ((14, 1),),
531: ((5, 1),),             1145: ((0, 1),),                                                  1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),    1152: ((7, 0),),                                                  1882: ((14, 2),),
650: ((18, 0), (19, 0)),    1224: ((10, 2),),                                                 1917: ((18, 1),),
689: ((6, 1),),             1325: ((4, 1),),                                                  1928: ((15, 0),),
699: ((15, 1),),            1349: ((9, 1),),                                                  1975: ((7, 2),)}
798: ((2, 1),),             1365: ((12, 2),),
807: ((6, 0),),             1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Li$_1$

```
{41: ((4, 2),),              878: ((1, 1),),                                                  1611: ((9, 2),),
 80: ((8, 0),),              926: ((3, 0), (4, 0)),                                           1683: ((11, 0), (13, 0)),
197: ((7, 1),),              935: ((1, 0),),                                                  1688: ((6, 2),),
212: ((2, 2),),             1000: ((3, 2),),                                                  1738: ((3, 1),),
216: ((16, 2),),            1011: ((10, 1), (12, 1)),                                         1741: ((17, 2),),
255: ((8, 1),),             1019: ((5, 0),),                                                  1750: ((16, 1), (17, 1)),
314: ((19, 1),),            1048: ((8, 2),),                                                  1778: ((1, 2),),
446: ((5, 2),),             1057: ((0, 0),),                                                  1840: ((14, 1),),
531: ((5, 1),),             1145: ((0, 1),),                                                  1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),    1152: ((7, 0),),                                                  1882: ((14, 2),),
650: ((18, 0), (19, 0)),    1224: ((10, 2),),                                                 1917: ((18, 1),),
689: ((6, 1),),             1325: ((4, 1),),                                                  1928: ((15, 0),),
699: ((15, 1),),            1349: ((9, 1),),                                                  1975: ((7, 2),)}
798: ((2, 1),),             1365: ((12, 2),),
807: ((6, 0),),             1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

No

```
{41: ((4, 2),),              878: ((1, 1),),                                                  1611: ((9, 2),),
 80: ((8, 0),),              926: ((3, 0), (4, 0)),                                           1683: ((11, 0), (13, 0)),
197: ((7, 1),),              935: ((1, 0),),                                                  1688: ((6, 2),),
212: ((2, 2),),             1000: ((3, 2),),                                                  1738: ((3, 1),),
216: ((16, 2),),            1011: ((10, 1), (12, 1)),                                         1741: ((17, 2),),
255: ((8, 1),),             1019: ((5, 0),),                                                  1750: ((16, 1), (17, 1)),
314: ((19, 1),),            1048: ((8, 2),),                                                  1778: ((1, 2),),
446: ((5, 2),),             1057: ((0, 0),),                                                  1840: ((14, 1),),
531: ((5, 1),),             1145: ((0, 1),),                                                  1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),    1152: ((7, 0),),                                                  1882: ((14, 2),),
650: ((18, 0), (19, 0)),    1224: ((10, 2),),                                                 1917: ((18, 1),),
689: ((6, 1),),             1325: ((4, 1),),                                                  1928: ((15, 0),),
699: ((15, 1),),            1349: ((9, 1),),                                                  1975: ((7, 2),)}
798: ((2, 1),),             1365: ((12, 2),),
807: ((6, 0),),             1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Li$_2$

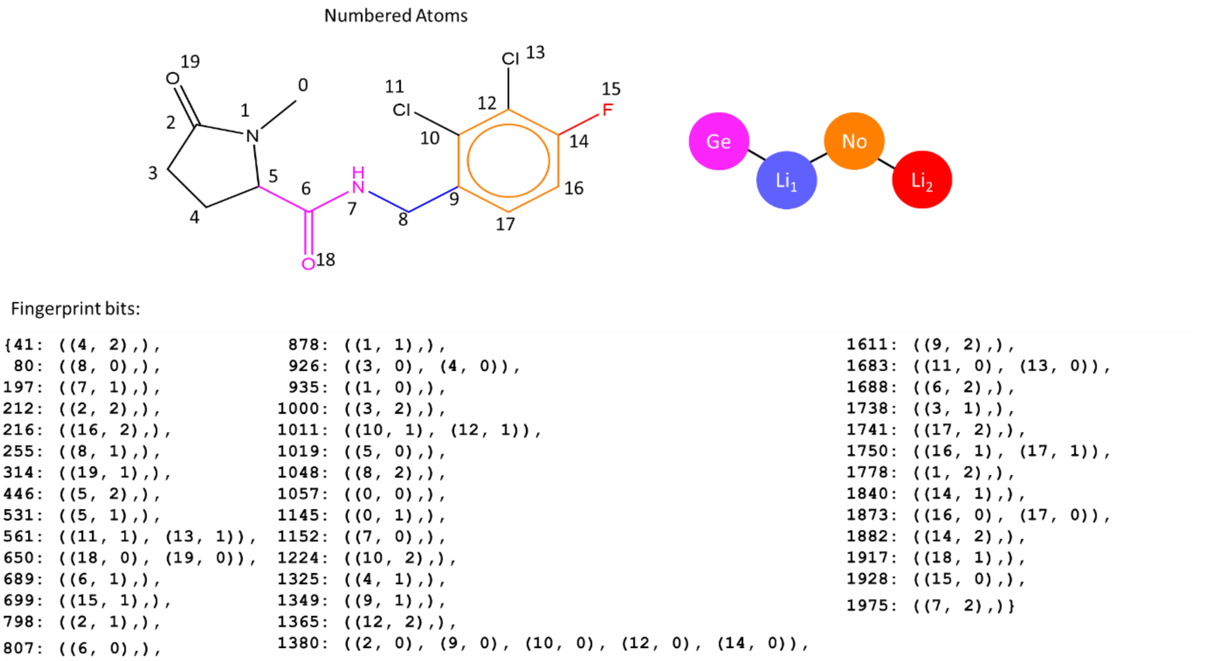
```
{41: ((4, 2),),              878: ((1, 1),),                                                  1611: ((9, 2),),
 80: ((8, 0),),              926: ((3, 0), (4, 0)),                                           1683: ((11, 0), (13, 0)),
197: ((7, 1),),              935: ((1, 0),),                                                  1688: ((6, 2),),
212: ((2, 2),),             1000: ((3, 2),),                                                  1738: ((3, 1),),
216: ((16, 2),),            1011: ((10, 1), (12, 1)),                                         1741: ((17, 2),),
255: ((8, 1),),             1019: ((5, 0),),                                                  1750: ((16, 1), (17, 1)),
314: ((19, 1),),            1048: ((8, 2),),                                                  1778: ((1, 2),),
446: ((5, 2),),             1057: ((0, 0),),                                                  1840: ((14, 1),),
531: ((5, 1),),             1145: ((0, 1),),                                                  1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),    1152: ((7, 0),),                                                  1882: ((14, 2),),
650: ((18, 0), (19, 0)),    1224: ((10, 2),),                                                 1917: ((18, 1),),
689: ((6, 1),),             1325: ((4, 1),),                                                  1928: ((15, 0),),
699: ((15, 1),),            1349: ((9, 1),),                                                  1975: ((7, 2),)}
798: ((2, 1),),             1365: ((12, 2),),
807: ((6, 0),),             1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Figure 7-7: All atom masking method bits that have been highlighted are the bits that are masked for each node

The second method, referred to as the all atom method, is to remove all of the tuples, and then bits, that contain any of the atoms involved in the substructure, as shown in Figure 7-7 where the tuples and bits that are removed are highlighted.

7.2.3 Comparing the fingerprint masking methods

The effects of each of the different masking techniques were analysed using the substructure represented by the terminal Li node of the molecule CHEMBL2218289. This substructure is a fluorine atom. The fingerprint bits of the original unaltered molecule are shown in Figure 7-8, with each atom labelled with its corresponding atom index and the 2048 Morgan 2 fingerprint bits.



Fingerprint bits:

```
{41: ((4, 2),),               878: ((1, 1),),                                          1611: ((9, 2),),
 80: ((8, 0),),               926: ((3, 0), (4, 0)),                                   1683: ((11, 0), (13, 0)),
197: ((7, 1),),               935: ((1, 0),),                                          1688: ((6, 2),),
212: ((2, 2),),              1000: ((3, 2),),                                          1738: ((3, 1),),
216: ((16, 2),),             1011: ((10, 1), (12, 1)),                                 1741: ((17, 2),),
255: ((8, 1),),              1019: ((5, 0),),                                          1750: ((16, 1), (17, 1)),
314: ((19, 1),),             1048: ((8, 2),),                                          1778: ((1, 2),),
446: ((5, 2),),              1057: ((0, 0),),                                          1840: ((14, 1),),
531: ((5, 1),),              1145: ((0, 1),),                                          1873: ((16, 0), (17, 0)),
561: ((11, 1), (13, 1)),     1152: ((7, 0),),                                          1882: ((14, 2),),
650: ((18, 0), (19, 0)),     1224: ((10, 2),),                                         1917: ((18, 1),),
689: ((6, 1),),              1325: ((4, 1),),                                          1928: ((15, 0),),
699: ((15, 1),),             1349: ((9, 1),),                                          1975: ((7, 2),)}
798: ((2, 1),),              1365: ((12, 2),),
807: ((6, 0),),              1380: ((2, 0), (9, 0), (10, 0), (12, 0), (14, 0)),
```

Figure 7-8: Unaltered molecule and corresponding M2FP bits

Figure 7-9 indicates the impact of each method of masking the fluorine atom. For both the removal of fragment method and the wild atom method, while bits are removed from the fingerprint, new bits are also gained due to the nature of the spherical fingerprint and the environment of the atoms that remain in the molecule changing. These methods were therefore not investigated further. For the two bit masking methods, the central atom method does not remove all of the bits associated with the F atom and was also not considered further. The method chosen was the all atom bit masking method which results in all bits associated with the F atom being removed.

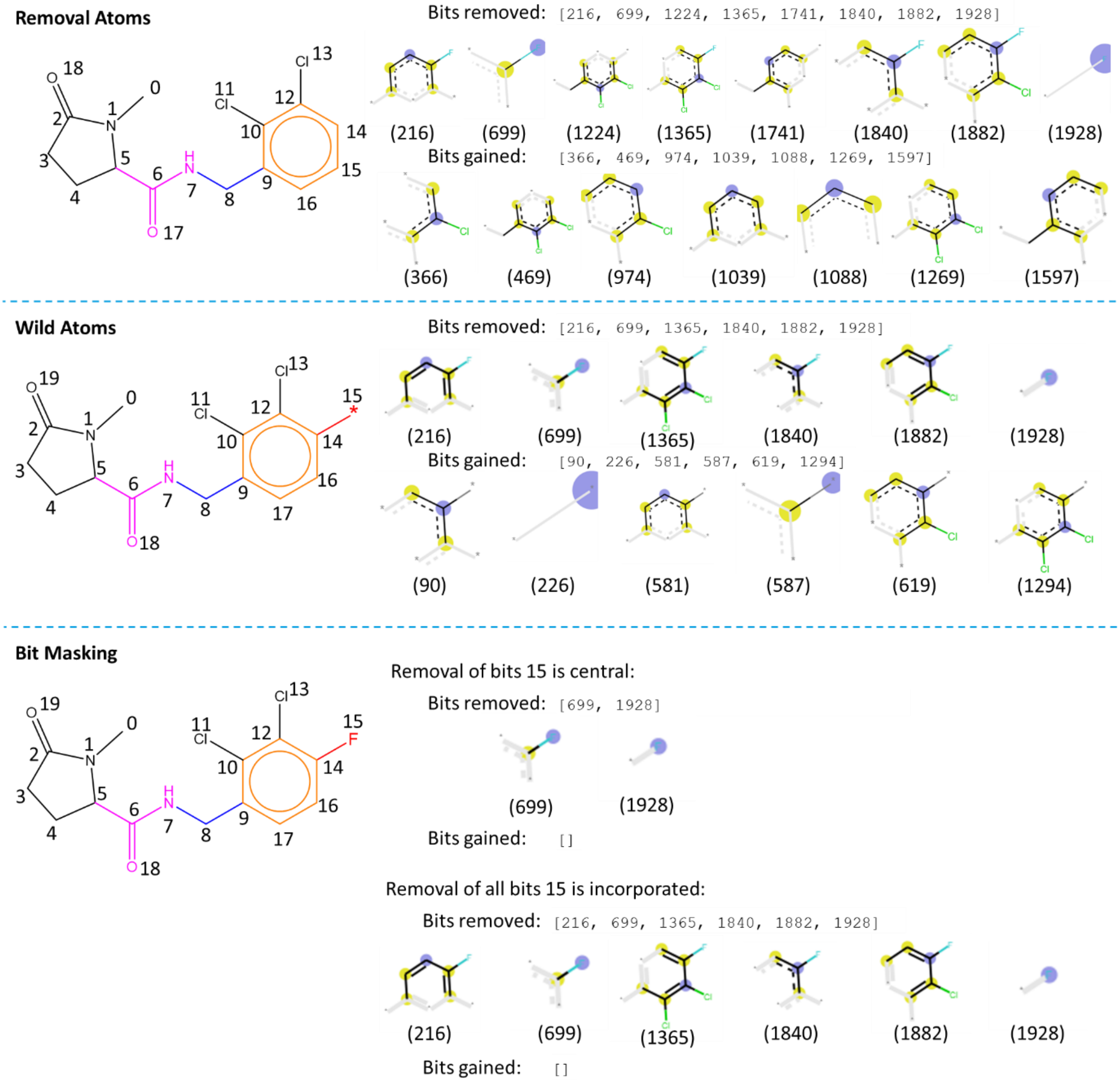

Removal Atoms
Bits removed: [216, 699, 1224, 1365, 1741, 1840, 1882, 1928]
(216) (699) (1224) (1365) (1741) (1840) (1882) (1928)
Bits gained: [366, 469, 974, 1039, 1088, 1269, 1597]
(366) (469) (974) (1039) (1088) (1269) (1597)
Wild Atoms
Bits removed: [216, 699, 1365, 1840, 1882, 1928]
(216) (699) (1365) (1840) (1882) (1928)
Bits gained: [90, 226, 581, 587, 619, 1294]
(90) (226) (581) (587) (619) (1294)
Bit Masking
Removal of bits 15 is central:
Bits removed: [699, 1928]
(699) (1928)
Bits gained: []
Removal of all bits 15 is incorporated:
Bits removed: [216, 699, 1365, 1840, 1882, 1928]
(216) (699) (1365) (1840) (1882) (1928)
Bits gained: []

Figure 7-9: Altered molecule with bits that have been removed or added depending on the methodology

7.3 Results

For each of the datasets, the molecules without a pIC50 value were removed and the rest of the dataset was split into a 90:10% split. This is the same split as used in the previous chapter. The QSAR models and substructure significance values were calculated using the 90% set and exploitation scores were calculated for the molecules in the 10% holdout sets.

7.3.1 Creation of Machine Learning Models

The statistics for the cross validated and final models built using the four machine learning methods are shown in Table 7-1 for all datasets.

Table 7-1: Table of model statistics for all datasets

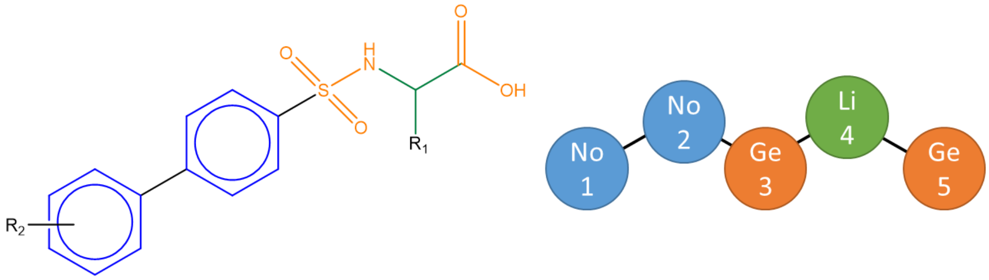
Dataset	Model	10-fold CV R^2		Final Model			
		Mean	Std	Mean Absolute Error	R^2	RMSE	MSE
Bajorath	RF	0.801	0.039	0.138	0.973	0.196	0.039
	GBM	0.736	0.050	0.387	0.825	0.498	0.248
	SVM	0.773	0.040	0.244	0.878	0.416	0.173
	PLS	0.677	0.038	0.456	0.727	0.621	0.386
CDK2	RF	0.734	0.064	0.199	0.960	0.301	0.091
	GBM	0.710	0.052	0.500	0.820	0.638	0.407
	SVM	0.764	0.051	0.278	0.904	0.466	0.393
	PLS	0.720	0.046	0.467	0.826	0.627	0.393
Chk1	RF	0.067	0.270	0.301	0.890	0.365	0.133
	GBM	-0.015	0.344	0.244	0.934	0.281	0.079
	SVM	0.096	0.267	0.445	0.684	0.618	0.382
	PLS	-0.016	0.289	0.367	0.812	0.476	0.226
Cyto	RF	0.306	0.064	0.118	0.892	0.177	0.031
	GBM	0.256	0.039	0.325	0.384	0.422	0.178
	SVM	0.348	0.059	0.155	0.716	0.287	0.082
	PLS	0.241	0.048	0.306	0.376	0.424	0.180
FactorXa	RF	0.667	0.052	0.216	0.946	0.304	0.093
	GBM	0.573	0.046	0.589	0.693	0.724	0.524
	SVM	0.669	0.050	0.354	0.829	0.540	0.292
	PLS	0.513	0.071	0.623	0.631	0.794	0.630
MMP12	RF	0.838	0.021	0.113	0.978	0.151	0.023
	GBM	0.836	0.021	0.280	0.881	0.352	0.124
	SVM	0.876	0.020	0.178	0.934	0.263	0.069
	PLS	0.827	0.028	0.306	0.848	0.399	0.159
Neurokinin	RF	0.643	0.039	0.240	0.931	0.347	0.120
	GBM	0.585	0.032	0.564	0.719	0.699	0.488
	SVM	0.660	0.031	0.355	0.831	0.542	0.294
	PLS	0.573	0.037	0.575	0.695	0.728	0.530
P2x7	RF	0.494	0.044	0.172	0.932	0.230	0.053
	GBM	0.390	0.043	0.464	0.578	0.573	0.328
	SVM	0.518	0.026	0.260	0.792	0.402	0.162
	PLS	0.376	0.056	0.460	0.552	0.590	0.348
P2x7 subset	RF	0.560	0.072	0.144	0.941	0.181	0.033
	GBM	0.511	0.074	0.318	0.733	0.385	0.148
	SVM	0.611	0.073	0.197	0.859	0.280	0.078
	PLS	0.411	0.138	0.350	0.656	0.437	0.191
p38α	RF	0.702	0.027	0.195	0.957	0.270	0.073
	GBM	0.553	0.027	0.629	0.642	0.781	0.609
	SVM	0.728	0.023	0.293	0.876	0.459	0.211
	PLS	0.584	0.040	0.586	0.669	0.751	0.563

For the majority of datasets, the SVM models had the highest 10-fold CV R^2 scores. For all datasets and models there is a slight increase in R^2 for the final models as would be expected as more data is being incorporated into the model. This increase is larger for the RF and GB models which is likely due to these methods being prone to overfitting. The 10-fold CV models for the Chk1 and Cyto datasets are poor which can be rationalised by the small number of molecules, Chk1, or the molecules being too dissimilar, Cyto, which has the lowest pairwise similarity, see Chapter 2.

The best machine learning method was selected based on the mean 10-fold cross validation R^2 score and the final model was used to calculate the fragment significance scores, i.e., the model built using all molecules within the dataset.

7.3.2 Fragment Significance

This section presents results for the MMP12 dataset as this consists of one RG core representing all 1534 molecules, Figure 7-10. Three of the nodes (2No, 3Ge and 5Ge) represent one substructural fragment, respectively, i.e., there is no variation in the molecules at these nodes. Nodes 1No and 4Li each represent multiple substructural fragments, nine and 11, respectively, and the number of molecules that have those substructural fragments at those node positions is given in the table.



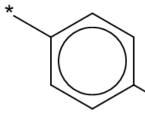
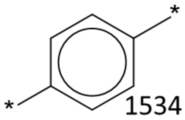
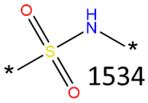
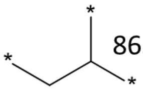
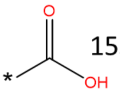
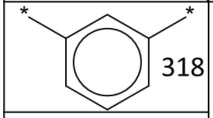
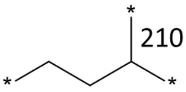
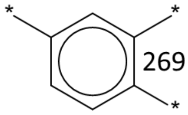
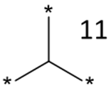
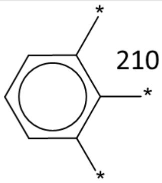
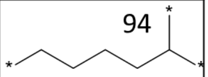
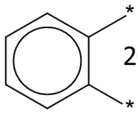
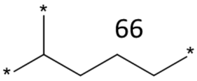
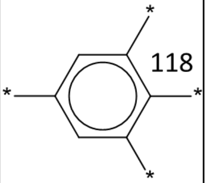
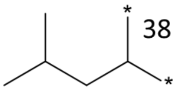
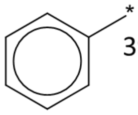
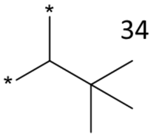
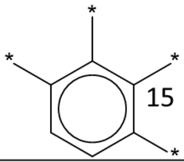
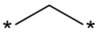
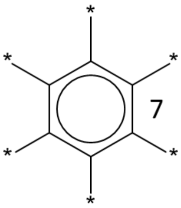
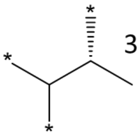
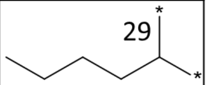
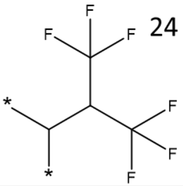
No 1	No 2	Ge 3	Li 4	Ge 5
 349	 1534	 1534	 865	 1534
 318			 210	
 269			 110	
 210			 94	
 209			 66	
 118			 38	
 39			 34	
 15			 33	
 7			 31	
			 29	
			 24	

Figure 7-10: RG Core extracted from the MMP12 dataset and substructural fragments present for each node with the number of occurrences

First, fragment significance values were calculated for a given fragment regardless of where that fragment occurred within the molecules, i.e., the mapping of the molecules to the RG core was not considered. This was then compared with the fragment significance values of the same fragment within a given position based on the mapping of the molecules onto the RG core. The aim was to examine whether the environment of a fragment is important when considering the contribution it makes to activity.

Figure 7-11 shows the distribution of local fragment significance values calculated for the para substituted benzene fragment regardless of its position in the molecules (orange distribution) compared to the fragment significance values for the same fragment at the position corresponding to node 1No in the RG core (blue and darker orange in the overlapping region). The fragment significance values were calculated using the SVM model and the all atom bit masking method.

The median and mean scores are 0.472 and 0.327 (standard deviation 0.594) when the environment is ignored and 1.015 and 0.988 (standard deviation of 0.295) when the environment is taken into account. This pattern is similar for most substructural fragments, unless there are very few examples that occur outside of the node. Therefore, it was concluded that the environment is important when assessing fragment significance with respect to activity and this is the approach used hereon.

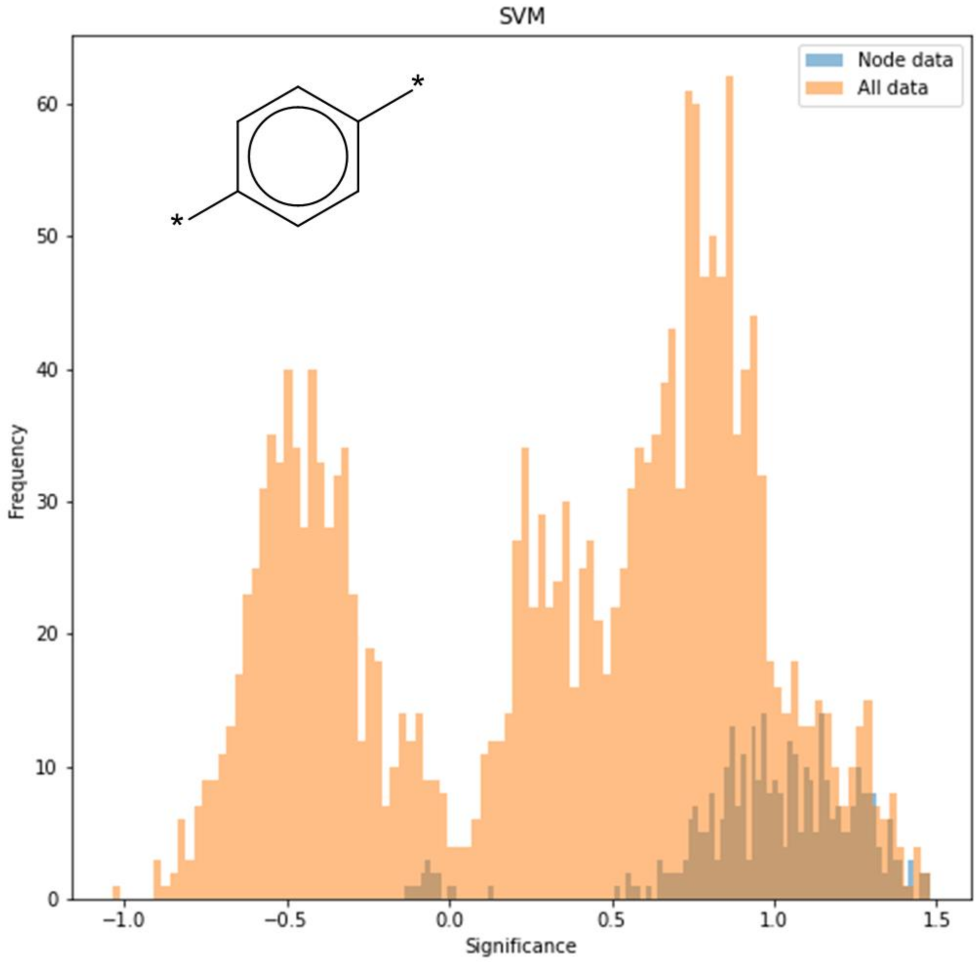


Figure 7-11: Fragment comparison of substructural fragment *c1ccc(*)cc1 from whole dataset (orange) and the specific environment node No1 (blue) in MMP12 dataset

7.3.2.1 Fragment Significance From The Different Methods of Bit Masking the Fingerprints

The different methods for masking the fingerprints are compared within this section.

7.3.2.1.1 Bit Masking All Occurrences

Fragment significance scores were calculated as median and mean (standard deviation) values considering each substructural fragment for each node in turn for the RG core [No][No][Ge][Li][Ge] representing the MMP12 dataset. The scores were calculated for all four machine learning methods.

Table 7-2 presents the results for the three nodes that each represent a single substructural fragment that is, therefore, common to all molecules in the dataset. For each fragment, the number of bits that was altered by removing the fragment and masking the bitstring is quoted. For all three examples, more than 60 bits are altered. The fragment significance scores are positive for all four machine learning methods (with the exception of the mean score for the first fragment using PLS) indicating that each fragment makes a positive contribution to the bioactivity. However, the values for a given fragment differ for the different machine learning methods. It was therefore decided to base the significance scores on the best modelling method rather than a consensus score, as used by Polishchuk et al. It is also notable that the standard deviations are large relative to the mean values indicating that the fragment significance values should be considered as a general trend. The fragment significance values differ across the fragments with the para substituted benzene fragment having the largest significance values.

Table 7-3 demonstrates the results for node 1No for which there are nine different substructural fragments. The fragments with the largest and smallest significance values are bolded, and the rank order for each model is included in the corner of each cell. There is more disagreement in the significance values for the fragments represented by this node when considering the different modelling methods, and for some fragments the contribution is positive or negative depending on the method. However, when examining the ranks, the models tend to be in agreement based on both median and mean values as shown by the overall average pearson rank coefficient of the pairwise values of 0.973. There are some slight differences in rank order for a given model when considering the median and the mean values which is due to the mean values being influenced by outliers. The fragment that is at rank one and is therefore predicted to contribute the most to the biological activity is the para-substituted benzene ring which is also the most frequent with 349 examples. According to the best model for the MMP12 dataset, SVM, the para-substituted benzene in this position has a median value of 1.015 indicating a positive effect on bioactivity. This is a reasonable increase as this is a log unit indicating a 1.015 increase in pIC50. In contrast, the fragment with the most negative effect on activity is the benzene ring with four adjacent substitution sites which has the median value of -1.178.

Table 7-4 shows the results for the fragments represented by the node 4Li. For a given fragment, there is strong agreement across the models on the contribution to bioactivity being either positive or negative. There are some slight variations in the rank orders of the fragments but these are not large variations as shown by the overall average pearson rank coefficient of the pairwise values of 0.953. The substructural fragment at rank one for all methods is [1,1,3-butyl], which has 210 examples and the median fragment significance of 0.595. In contrast, the substructural fragment with the most negative median fragment significance is [1,1,2-propyl], with 31 examples, with median decrease in biological activity of -0.342.

It is also important to note that the standard deviation of the fragment significance can be quite large, and in some instances larger than the significance value itself. This is the case for most of the substructural fragments for each node. A large standard deviation indicates that the local significances that are calculated have a large range.

Table 7-2: Node significance for nodes 2No, 3Ge and 5Ge in [No][No][Ge][Li][Ge] which contain just one substructural fragment

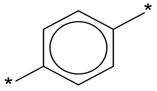
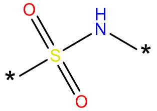
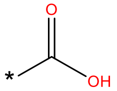
			Number of Bits That Are Altered	RF			GBM			SVM			PLS		
				Median	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median	Mean	Std
2No		1534	79	0.774	1.041	0.990	0.729	0.845	0.750	0.353	0.203	0.553	0.133	-0.043	0.357
3Ge		1534	69	0.064	0.097	0.179	0.134	0.126	0.134	0.175	0.202	0.231	0.149	0.130	0.097
5Ge		1534	63	0.063	0.094	0.178	0.133	0.124	0.132	0.106	0.116	0.166	0.102	0.080	0.100

Table 7-3: Node significance for node 1No substructural fragments in [No][No][Ge][Li][Ge]

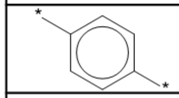
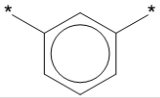
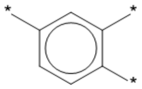
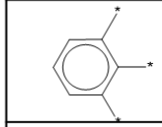
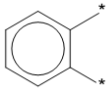
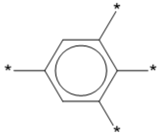
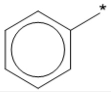
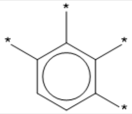
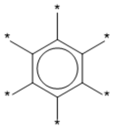
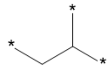
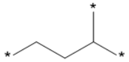
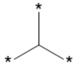
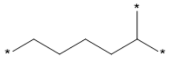
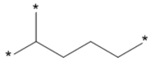
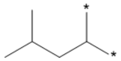
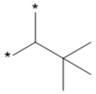
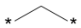
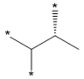
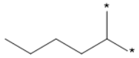
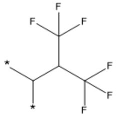
1No		Number of Bits That Are Altered	RF			GBM			SVM			PLS		
			Median	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median	Mean	Std
	349	50	**2.396** [1]	**2.314** [1]	0.580	**2.035** [1]	**1.998** [1]	0.380	**1.015** [1]	**0.988** [1]	0.295	**0.880** [1]	**0.787** [1]	0.273
	318	64	0.822 [5]	0.969 [4]	0.670	0.638 [5]	0.777 [4]	0.459	-0.387 [5]	-0.221 [4]	0.556	-0.541 [5]	-0.346 [4]	0.472
	269	70	0.992 [3]	1.160 [3]	0.905	0.992 [3]	0.900 [3]	0.796	-0.171 [3]	-0.134 [3]	0.808	-0.327 [3]	-0.277 [3]	0.780
	210	63	0.160 [6]	0.176 [7]	0.224	0.018 [7]	0.000 [7]	0.111	-0.883 [6]	-0.901 [7]	0.337	-0.946 [6]	-1.020 [7]	0.361
	209	43	0.156 [7]	0.347 [6]	0.519	0.032 [6]	0.169 [6]	0.353	-0.903 [7]	-0.833 [6]	0.432	-1.029 [7]	-0.937 [6]	0.368
	118	28	0.937 [4]	0.880 [5]	0.645	0.849 [4]	0.724 [5]	0.591	-0.245 [4]	-0.376 [5]	0.559	-0.451 [4]	-0.470 [5]	0.654
	39	6	1.704 [2]	1.757 [2]	0.385	1.768 [2]	1.744 [2]	0.227	0.464 [2]	0.465 [2]	0.076	0.045 [2]	0.082 [2]	0.052
	15	29	0.053 [8]	0.038 [8]	0.204	-0.103 [8]	-0.110 [8]	0.075	**-1.178** [9]	**-1.177** [9]	0.222	**-1.435** [9]	-1.404 [8]	0.050
	7	9	**-0.165** [9]	**-0.166** [9]	0.028	**-0.158** [9]	**-0.148** [9]	0.024	-1.151 [8]	-1.155 [8]	0.048	-1.405 [8]	**-1.407** [9]	0.004

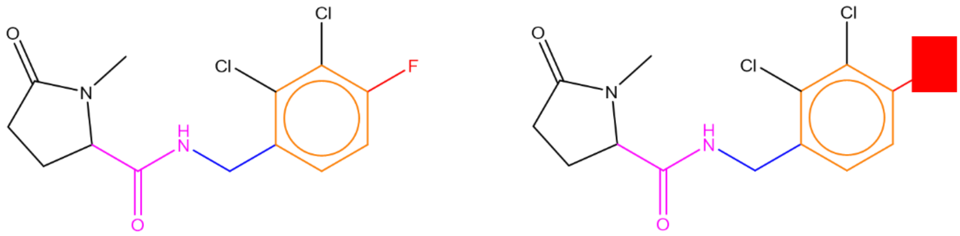
Table 7-4: Node significance for node 3Li substructural fragments in [No][No][Ge][Li][Ge]

4Li		Number of Bits That Are Altered	RF			GBM			SVM			PLS		
			Median	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median	Mean	Std
	865	106	0.113 [7]	0.116 [7]	0.220	0.156 [7]	0.123 [7]	0.105	0.160 [7]	0.169 [7]	0.177	0.259 [5]	0.235 [6]	0.141
	210	35	0.366 [4]	**0.429** [1]	0.377	**0.443** [1]	**0.447** [1]	0.220	**0.595** [1]	**0.552** [1]	0.306	**0.631** [1]	**0.527** [1]	0.193
	110	22	-0.042 [9]	-0.067 [9]	0.186	-0.070 [9]	-0.085 [9]	0.106	-0.040 [9]	-0.056 [9]	0.129	-0.019 [9]	0.031 [8]	0.113
	94	23	0.203 [6]	0.226 [6]	0.282	0.223 [6]	0.212 [6]	0.103	0.339 [6]	0.322 [6]	0.172	0.222 [7]	0.207 [7]	0.041
	66	20	0.319 [5]	0.340 [5]	0.269	0.285 [4=]	0.244 [5]	0.102	0.352 [5]	0.371 [5]	0.210	0.246 [6]	0.262 [5]	0.030
	38	15	0.454 [2]	0.427 [2]	0.319	0.406 [2]	0.334 [2]	0.123	0.510 [2]	0.501 [2]	0.144	0.400 [4]	0.400 [4]	0.018
	34	13	0.398 [3]	0.353 [4]	0.259	0.312 [3]	0.281 [3]	0.048	0.484 [3]	0.495 [3]	0.089	0.508 [2]	0.508 [2]	0.009
	33	8	**-0.127** [11]	-0.133 [10]	0.241	-0.014 [10]	**-0.134** [11]	0.148	-0.170 [10]	-0.219 [10]	0.128	-0.074 [10]	-0.080 [10]	0.027
	31	14	-0.120 [10]	**-0.165** [11]	0.117	**-0.117** [11]	-0.123 [10]	0.058	**-0.342** [11]	**-0.325** [11]	0.083	**-0.198** [11]	**-0.198** [11]	0.002
	29	17	**0.461** [1]	0.400 [3]	0.303	0.285 [4=]	0.271 [4]	0.117	0.409 [4]	0.386 [4]	0.183	0.489 [3]	0.471 [3]	0.043
	24	15	0.013 [8]	0.027 [8]	0.284	-0.023 [8]	-0.006 [8]	0.059	0.031 [8]	-0.008 [8]	0.253	0.002 [8]	-0.009 [9]	0.019

7.3.3 Uncertainty

The RF and GBM machine learning methods have in-built ways of estimating the uncertainty of predictions, therefore, these two methods are further examined.

Both RF and GBM are ensemble approaches consisting of multiple trees each of which makes a prediction based on input data. The predictions made by individual trees are then combined to generate the overall prediction. The standard deviation of the predictions provides a measure of uncertainty that is associated with a given prediction. When calculating local fragment significance using either RF and GBM, the uncertainties in the predictions for both the original molecule and the masked molecule were found. As the significance score is the original molecule's prediction minus the masked molecule's prediction, the uncertainty of the fragment significance score is the sum of these two uncertainties. An example is given in Figure 7-12 for one molecule and one masked substructural fragment.



Original Molecule

Masked Molecule

RF – 100 Trees

Original Molecule

	Tree 1	Tree 2	…	Tree 100
Prediction	7.4	7.6	…	7.7

Overall Prediction = 7.408

Standard Deviation = 0.186

Masked Molecule

	Tree 1	Tree 2	…	Tree 100
Prediction	6.7	6.6	…	6.9

Overall Prediction = 6.777

Standard Deviation = 0.158

Substructural Fragment Significance = Original Molecule – Masked Molecule

$$0.631 = 7.408 - 6.777$$

Uncertainty = Original Molecule Std + Masked Molecule Std

$$0.344 = 0.186 + 0.158$$

Figure 7-12: An example of how the uncertainty of a significance score is generated

The median, mean and standard deviation of the uncertainty values for each substructural fragment were calculated. The results for all substructural fragments and all nodes for the MMP12 dataset are shown in Table 7-5, with the uncertainty data shown in the shaded columns. The distribution of fragment significance and the uncertainty values are shown for the C(=O)O fragment at node 5 Ge , Figure 7-13. Together, these demonstrate that the uncertainties of the significance scores are large and in some instances are bigger than the significance score itself. These large uncertainties may be due to changes in multiple bits at once and due to the large sizes of the fragments being removed. The high uncertainties indicate that the significance scores may not be as accurate as originally hoped, however, they give some indications of trends, which may be better than a random selection.

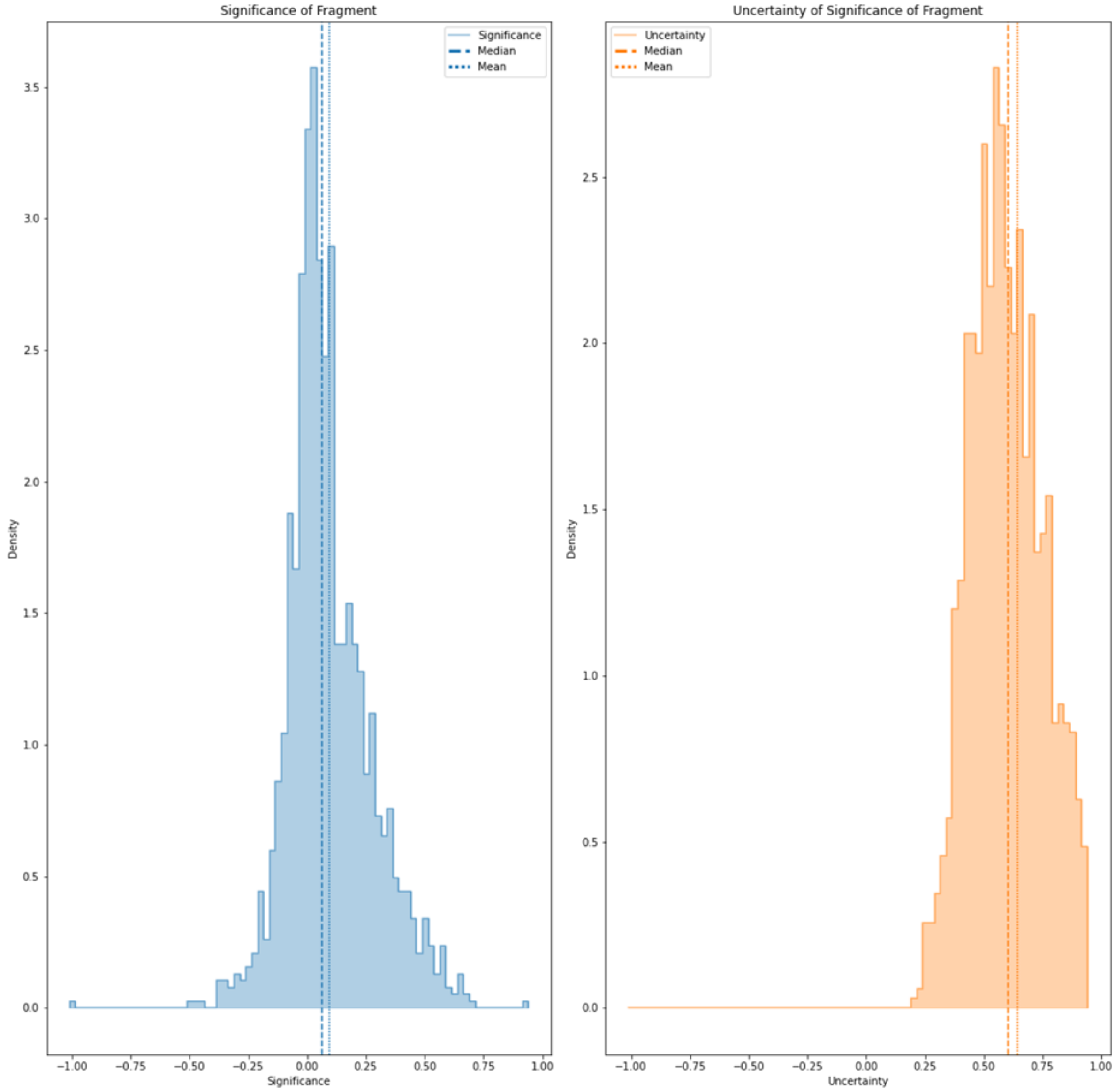
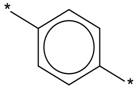
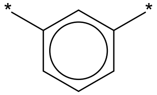
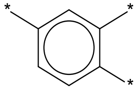
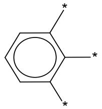
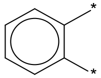
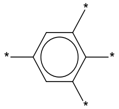
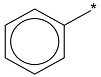
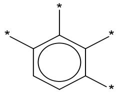
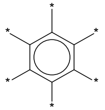
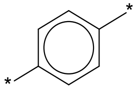
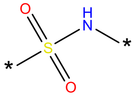
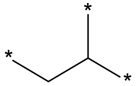
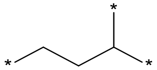
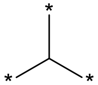
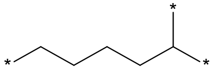
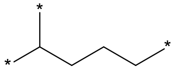
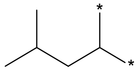
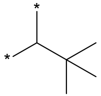
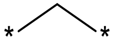
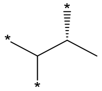
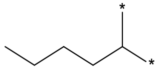
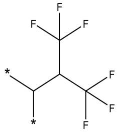
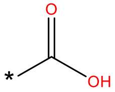


Figure 7-13: Node 5Ge substructural fragment *C(=O)O distribution of significance and uncertainties

Table 7-5: All nodes and substructural fragments significance and uncertainty for RF and GBM for core [No][No][Ge][Li][Ge] in MMP12 dataset, of the bit masked approach

Node	Substructural Fragment	RF						GBM					
		Significance			Uncertainty			Significance			Uncertainty		
		Median	Mean	Std	Median	Mean	Std	Median	Mean	Std	Median	Mean	Std
1No		2.396	2.314	0.580	0.569	0.601	0.163	2.035	1.998	0.380	0.469	0.466	0.041
		0.822	0.969	0.670	0.630	0.643	0.129	0.638	0.777	0.459	0.353	0.355	0.047
		0.992	1.160	0.905	0.605	0.627	0.134	0.992	0.900	0.796	0.417	0.415	0.030
		0.160	0.176	0.224	0.677	0.670	0.149	0.018	0.000	0.111	0.402	0.378	0.050
		0.156	0.347	0.519	0.608	0.650	0.198	0.032	0.169	0.353	0.410	0.389	0.056
		0.937	0.880	0.645	0.590	0.599	0.121	0.849	0.724	0.591	0.376	0.375	0.045
		1.704	1.757	0.385	0.585	0.617	0.122	1.768	1.744	0.227	0.455	0.453	0.035
		0.053	0.038	0.204	0.594	0.602	0.085	-0.103	-0.110	0.075	0.419	0.413	0.035
		-0.165	-0.166	0.028	0.530	0.548	0.107	-0.158	-0.148	0.024	0.430	0.431	0.009
2No		0.774	1.041	0.990	0.626	0.652	0.182	0.729	0.845	0.750	0.415	0.410	0.072

3Ge		0.064	0.097	0.179	0.609	0.646	0.221	0.134	0.126	0.134	0.407	0.392	0.103
4Li		0.113	0.116	0.220	0.610	0.649	0.216	0.156	0.123	0.105	0.403	0.382	0.106
		0.366	0.429	0.377	0.637	0.665	0.177	0.443	0.447	0.220	0.416	0.410	0.100
		-0.042	-0.067	0.186	0.574	0.626	0.227	-0.070	-0.085	0.106	0.401	0.384	0.099
		0.203	0.226	0.282	0.605	0.628	0.189	0.223	0.212	0.103	0.405	0.392	0.102
		0.319	0.340	0.269	0.670	0.666	0.152	0.285	0.244	0.102	0.428	0.411	0.087
		0.454	0.427	0.319	0.625	0.623	0.140	0.406	0.334	0.123	0.407	0.388	0.098
		0.398	0.353	0.259	0.615	0.648	0.154	0.312	0.281	0.048	0.419	0.396	0.095
		-0.127	-0.133	0.241	0.647	0.701	0.188	-0.014	-0.134	0.148	0.401	0.380	0.090
		-0.120	-0.165	0.117	0.677	0.670	0.160	-0.117	-0.123	0.058	0.418	0.412	0.107
		0.461	0.400	0.303	0.536	0.536	0.109	0.285	0.271	0.117	0.403	0.383	0.110
		0.013	0.027	0.284	0.681	0.625	0.153	-0.023	-0.006	0.059	0.401	0.371	0.109
5Ge		0.063	0.094	0.178	0.601	0.643	0.222	0.133	0.124	0.132	0.406	0.390	0.105

7.3.4 Comparing Significance of Fragments To Other Methods

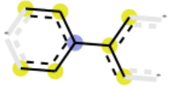
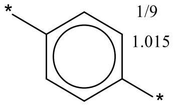
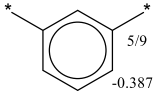
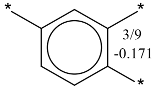
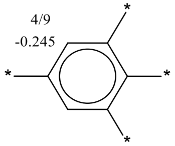
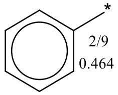
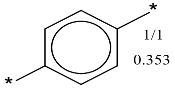
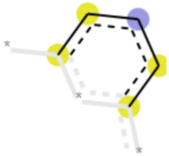
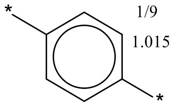
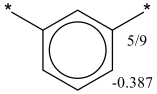
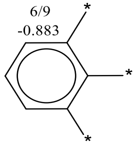
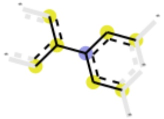
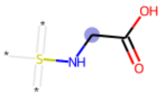
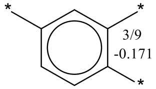
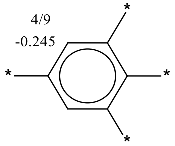
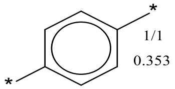
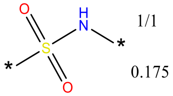
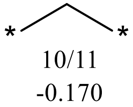
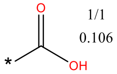
The significance of each of the fragments is compared to other existing methods that attempt to find significant fragments. Both the significance score and the substructural fragments that the score is linked to are compared.

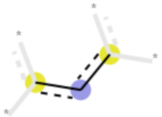
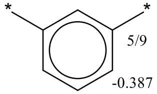
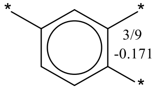
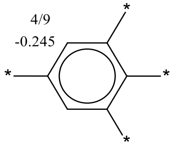
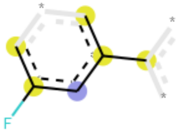
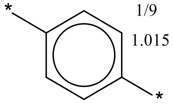
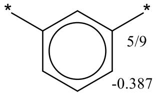
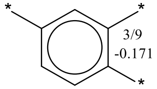
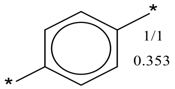
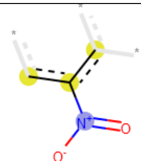
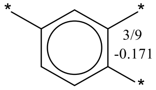
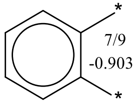
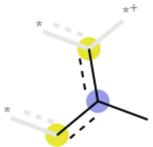
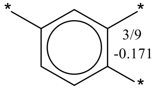
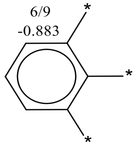
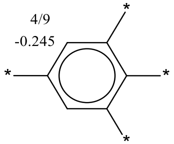
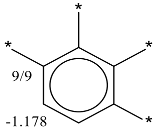
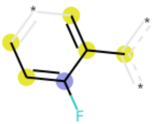
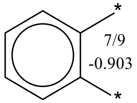
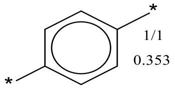
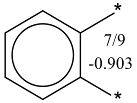
7.3.4.1 Comparison to Random Forest significance bits

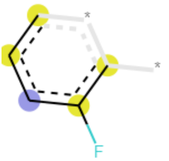
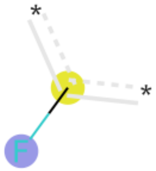
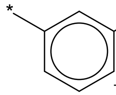
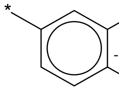
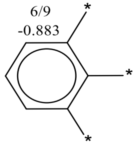
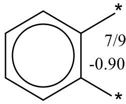
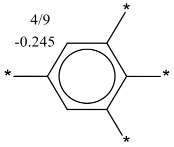
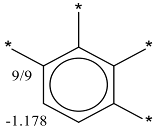
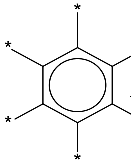
The random forest (RF) method in sci-kit learn returns the feature importance of each bit from the input data that is included in the model. Therefore, the feature importance values are compared to the information extracted from the fragment significances. To extract the feature importance from a RF the methodology observes the performance of the RF when features are absent. The default feature importance in sci-kit learn is based on the mean decrease in impurities. This method assesses how each feature, bit, decreases the impurity of the tree node split within the tree and these are averaged over all trees within the forest to provide a feature importance.

Table 7-6 shows the top ten bits according to feature importance. An image of what each bit represents is shown, along with the number of times that bit is seen within the dataset. For each node, within the RG core, the substructural fragment is displayed if this bit is disguised in the masking process. For each of these substructural fragments the significance score and rank number is provided.

Table 7-6: Top ten bits with feature importance for MMP12 dataset

Bit	Feature Importance	Number of Molecules With Bit	Present within substructural fragment…				
			1No	2No	3Ge	4Li	5Ge
 1964	0.409	942	    		-	-	-
 1039	0.140	564	  	-	-	-	-
 945 	0.028	115	 				

 875	0.027	762	  	-	-	-	-
 1034	0.016	89	  		-	-	-
 354	0.012	89	 	-	-	-	-
 1391	0.011	185	   	-	-	-	-
 451	0.009	42			-	-	-
	0.009	42		-	-	-	-

 1032							
 699	0.007	404	 5/9 -0.387  3/9 -0.171  6/9 -0.883  7/9 -0.903  4/9 -0.245  9/9 -1.178  8/9 -1.151	-	-	-	-

All of the top ten bits have atoms that are contained within the chemical graph mapping of the RG core. It is interesting to note that bit 945 demonstrates a bit clash as two different substructures are represented by this bit. This could have an impact on the bit masking for the significance score, if both fragments are present and only one is being masked then for the bit masking it would not be removed. However, if the count FP was to be used it would be reduced.

The feature importance results focus on specific bits, whereas the method outlined in this chapter examines the effect of a fragment and therefore multiple bits. When looking at the node substructural fragments, the top five fragments from 1No with the most positive fragment significance scores all contain the bit that is deemed most important by the RF. However, their significance scores are not all within the top five scores of the overall substructural fragments observed for all nodes within this RG core. But when using the count masked FP they are the top five scores. It is also interesting that one bit can be seen in multiple different substructures across multiple node, such as bit 945.

When comparing the feature importance to the associated substructural fragments significance it is interesting to note that seven out of nine of the substructural fragments within node 1No have negative significance scores. However, these all seem to contain a bit or bits that are in the top ten feature importance. Yet when ranking all the substructural fragments observed within the RG core only two are in the top ranked fragments based on their significance score. This difference could be due to the nature of some of the R-groups but also one bit does not provide much information, however, when combined with multiple bits can provide more of an overall picture of the impact of that substructural fragment. Although, more of these fragments within the count masked FP are considered to be within the top ranked substructural fragments based on their significance scores.

A drawback of the RF impurity based feature importance method is that the results can be biased towards high cardinality features, meaning where a feature has lots of potential values. This is not a problem for this investigation as none of the features have high cardinality as the feature is either a bit turned on 1 or turned off 0 (Pedregosa et al., 2011).

7.3.4.2 Comparison to Matched Molecular Pairs

Matched molecular pairs (MMPs) are a powerful tool in identifying fragments that are significant. Therefore, the MMPs are extracted to analyse the differences between the MMPs and the significances of the substructural fragments. A MMP links a molecular transformation with a molecular property. An example is a hydrogen atom to a fluorine atom that could be linked to a specific increase in biological activity.

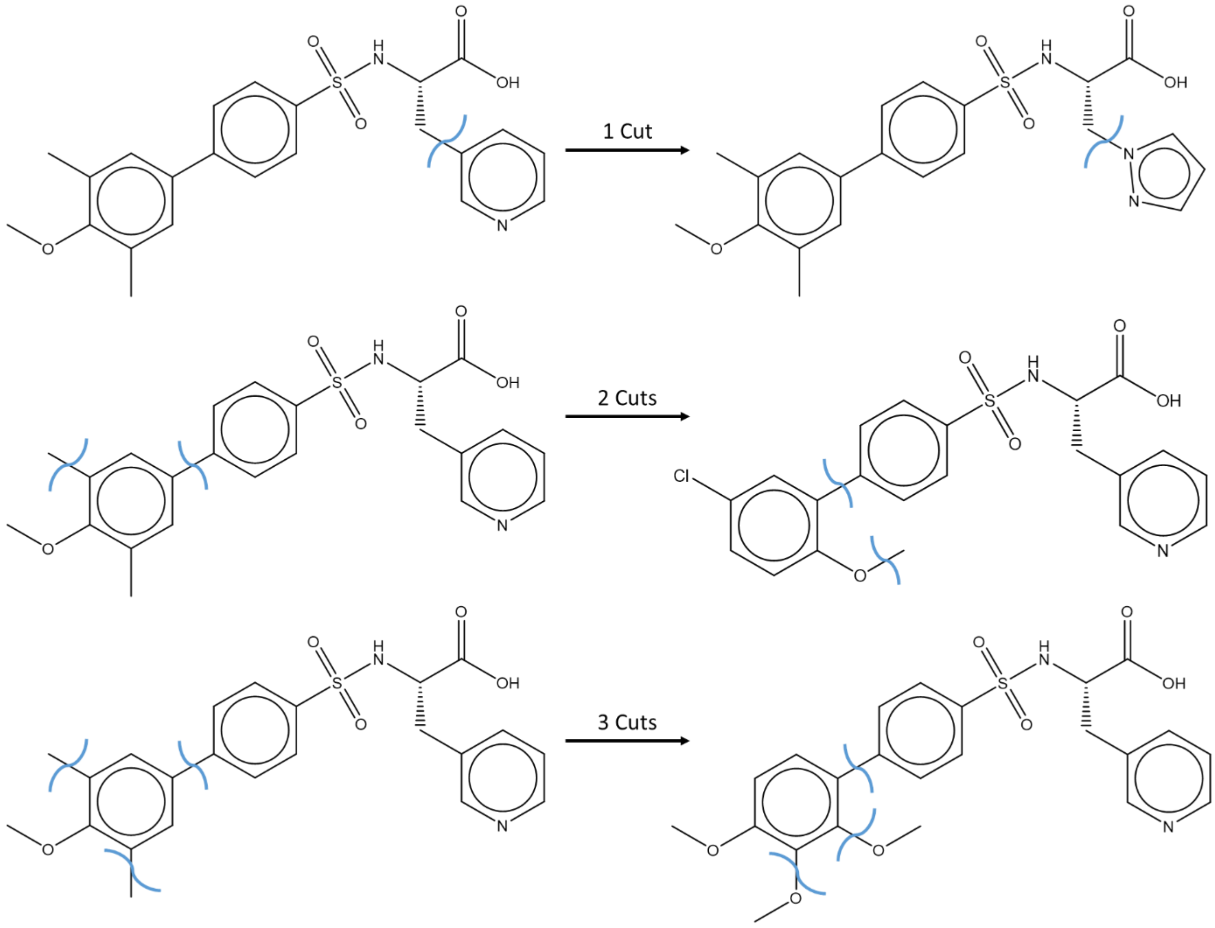


Figure 7-14: MMP example for one cute, two cuts and three cuts for molecule A35B31 in MMP12 dataset

Unfortunately, MMPs are not as useful when handling datasets that are sparse and small as they can be more difficult to extract these transformations as there is less likely to be three examples of the transformation to become a MMP. The Hussain and Rea algorithm with one, two and three cuts with at least three examples were extracted from the datasets to see if MMPs could be used to identify the significance of substructural fragments (Hussain & Rea, 2010). An example can be seen in Figure 7-14, where the cuts are shown with blue curved lines. Table 7-7 shows the number of MMP extracted for each dataset with one, two or three cuts. The number in the square brackets indicates the maximum number of examples for a single MMP.

Table 7-7: Number of MMP for each dataset at 1, 2 and 3 cuts [max number is the maximum number of example for a single MMP]

Dataset	Number of Molecules in 90%	Number of MMPs with 1 Cut	Number of MMPs with 2 Cuts	Number of MMPs with 3 Cuts
Bajorath	1876	245 [max 16]	283 [max 21]	37 [max 11]
CDK2	1230	24 [max 5]	79 [max 6]	65 [max 5]
Chk1	95	0	0	0
Cyto	5676	365 [max 41]	155 [max 88]	75 [max 8]
FactorXa	1760	86 [max 16]	85 [max 11]	66 [max 7]
MMP12	1534	1065 [max 83]	747 [max 54]	221 [max 38]
Neurokinin	1321	104 [max 10]	100 [max 26]	93 [max 20]
P2x7	1607	409 [max 46]	725 [max 53]	214 [max 37]
P2x7 Subset	622	106 [max 10]	302 [max 30]	73 [max 30]
P38α	3280	274 [max 19]	410 [max 26]	280 [max 12]

It is clear to see that some datasets are more amenable to extracting MMPs than others, for example MMP12 compared to Chk1 for which no MMPs were found. Also, when looking at the maximum number of examples of a MMP some datasets also produce MMP that have more examples than others. A good example of this is the CDK2 dataset compared to the MMP12 dataset. Both datasets contain more than 1000 molecules, yet the maximum number of examples for a MMP is six within the CDK2 dataset, but 83 within the MMP12 dataset. MMPs rely on lots of data that is not sparse, whereas, the exploitation score can still provide some information about small sparse datasets.

When the MMPs were examined, some were bigger than the node substructural fragments identified by reduced graphs. Some of the MMPs do include features of the RG core. An example is demonstrated in Figure 7-15. This MMP was seen 14 times in the 90% linked to an increase the pIC50 value of 0.429 with a standard deviation of 0.383. The standard deviation of these increases were also large. In terms of RG nodes, this MMP goes from being two nodes to three nodes so would not be identified in the RG nodes significance score.

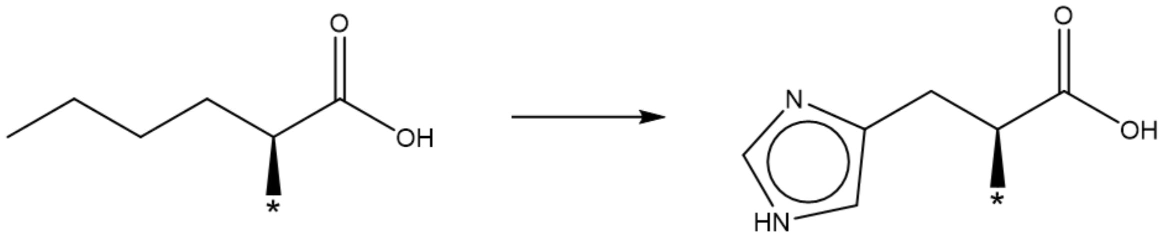
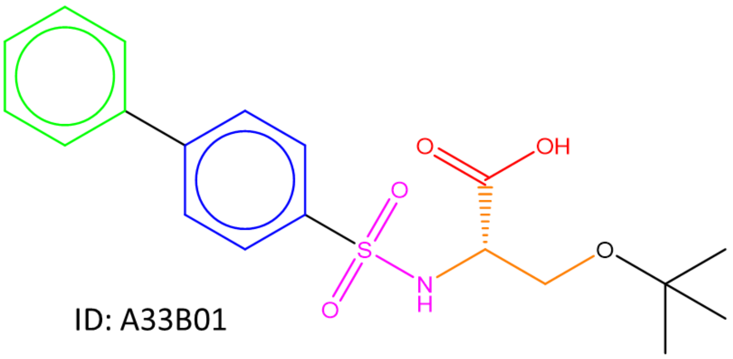
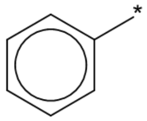
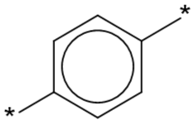
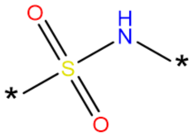
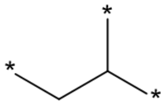
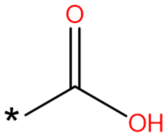


Figure 7-15: MMP with 1 cut seen in MMP12 dataset

7.3.5 Creating an Exploitation Score

Even though the confidence level of the scores is rather low, they have been used to generate an exploitation score to act as a guide for a chemist when considering which molecule should be made next. A new molecule is mapped onto a RG core and for each node the median significance score of the substructure at that node is retrieved based on the best performing model. When a substructural fragment has not been seen before, it is assigned a significance score of zero as there is no knowledge for that substructural fragment. The exploitation score for the molecule is then the mean of the median value of the significance scores for its constituent fragments. The mean value is used rather than the sum to allow the exploitation scores to be compared across cores. The higher the exploitation score the more a new molecule is considered to investigate an area of chemical space likely to be active. Figure 7-16 demonstrates how the exploitation score for a molecule would be generated, using one of the holdout molecules from the MMP12 dataset.



1No	2No	3Ge	4Li	5Ge
				

Significance Scores (Median) From Top Model (SVM):

0.464	0.353	0.175	0.160	0.106

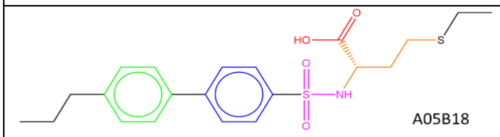
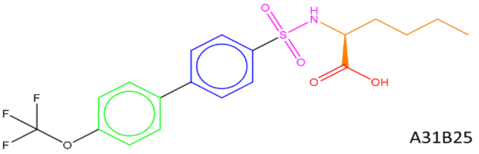
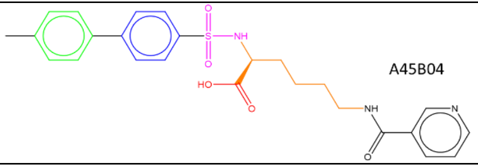
Exploitation Score = 0.252

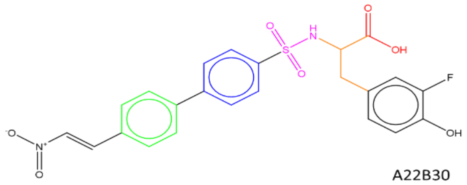
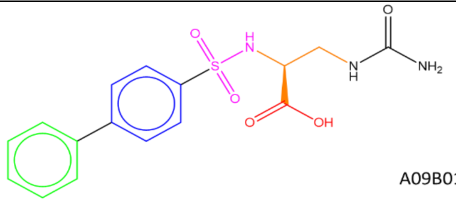
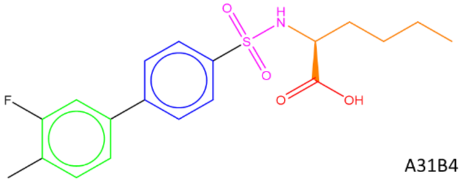
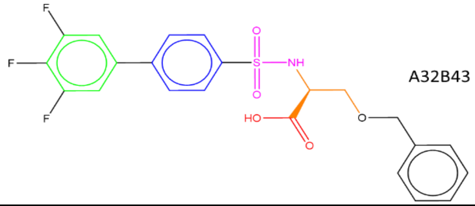
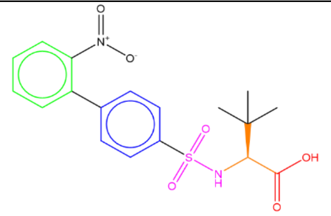
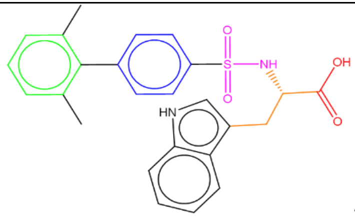
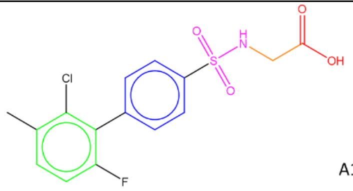
Figure 7-16: Example of generating exploitation score for molecule A33B01 within 10% of MMP12 dataset, MMP12 RG Core significance scores can be found in the Appendix

7.3.5.1 Generating Scores for the Ten Percent Hold Out Set

All 170 molecules in the MMP12 holdout set were scored. Table 7-8 shows a selection of molecules with different exploitation scores from this holdout percent where the scores range from 0.449 to -0.143. Nine molecules have the highest score substructure at each node position and therefore have the highest exploitation score of 0.449, with one shown in the top row of the table.

Table 7-8: Ten molecule that have had been assigned an exploitation score according to significance of core [No][No]Ge][Li][Ge] from MMP12 dataset

Molecule	1No	2No	3Ge	4Li	5Ge	Exploitation Score	Original pIC50
	1.015	0.353	0.175	0.595	0.106	0.449	7.5
	1.015	0.353	0.175	0.409	0.106	0.412	8.0
	1.015	0.353	0.175	0.339	0.106	0.398	6.6

 A22B30	1.015	0.353	0.175	0.160	0.106	0.362	6.4
 A09B01	0.464	0.353	0.175	0.160	0.106	0.252	5.3
 A31B48	-0.171	0.353	0.175	0.409	0.106	0.174	7.1
 A32B43	-0.245	0.353	0.175	0.160	0.106	0.120	6.1
 A25B09	-0.903	0.353	0.175	0.484	0.106	0.043	3.9
 A28B22	-0.883	0.353	0.175	0.160	0.106	-0.018	4.6
 A11B50	-1.178	0.353	0.175	-0.170	0.106	-0.143	3.8

As the holdout molecules have already been made and tested, the pIC50 values were examined. The pIC50 values range from 3.8 to 8. There are 170 molecules contained within the holdout set. These generate 48 unique exploitation scores. The molecule with the highest pIC50 value was molecule A31B25, which has an exploitation score of 0.412. This molecule was ranked 12 out of the 170 molecules using the exploitation score, however, there are only two higher exploitation scores. These molecules with the two highest exploitation scores have pIC50 values ranging from 5.3 to 7.7. The two molecules with the lowest pIC50 values, molecules A11B47 and A11B50 have exploitation scores of 0.015 and -0.143, respectively. A11B50 is the molecule with the lowest exploitation score and A11B47 is ranked 130 out 170 molecules, which is the 36 highest exploitation score.

Four of the nine molecules with the highest exploitation score, also had high pIC50 values, with them being ranked second, third, seventh and tenth according to their pIC50 values. This indicates that the exploitation score can identify new molecules with high activity values, although, the remaining three molecules were ranked 33, 53 and 81 on their pIC50 values. The range of pIC50 values, 5.3 to 7.7, for the top exploitation score may indicate that the R-groups also play a key role in the biological activity, whereas, the exploitation score is based on the substructures that map to an RG core only.

7.4 Conclusion

An exploitation score is generated for a molecule that matches to an existing RG core. The exploitation score is based upon the substructural fragments that have already been investigated for each node. The exploitation score is found by calculating the mean of the median significance score that has been generated for each substructural fragment at that point in the existing molecules. This method is based upon work that has already been done by Polishchuk. The input for the molecule, however, has been altered.

A Morgan 2 fingerprint was used with several different ways of masking the substructural fragment of interest. The best approach was masking the bits that incorporated any atom that was associated to the fragment of interest. Only the substructural fragment that is present at that specific node are combined together to generate a significance score as the environment in which the fragment is was also shown to be important.

The significance scores generated within this investigation had high standard deviations and tend to have a high uncertainty associated with them. This uncertainty was found from the RF and GBM models, due to the ensemble nature of the models.

These significance scores were combined together by finding the mean to generate an exploitation score. Even with these high uncertainties, when examining the hold out dataset it was found that the molecules that had a higher exploitation score generally had a higher pIC50 value. Therefore, these large uncertainties and standard deviations may be caused by noise within the model. The exploitation score produced is not perfect but provides a good estimation from the existing information as to what molecule to potentially investigate next.

The next chapter shall use all the work done in previous chapters to generate new molecules to suggest to the chemist and present them with an exploration and an exploitation score.

8 Molecular Generation Using Reduced Graphs

8.1 Introduction

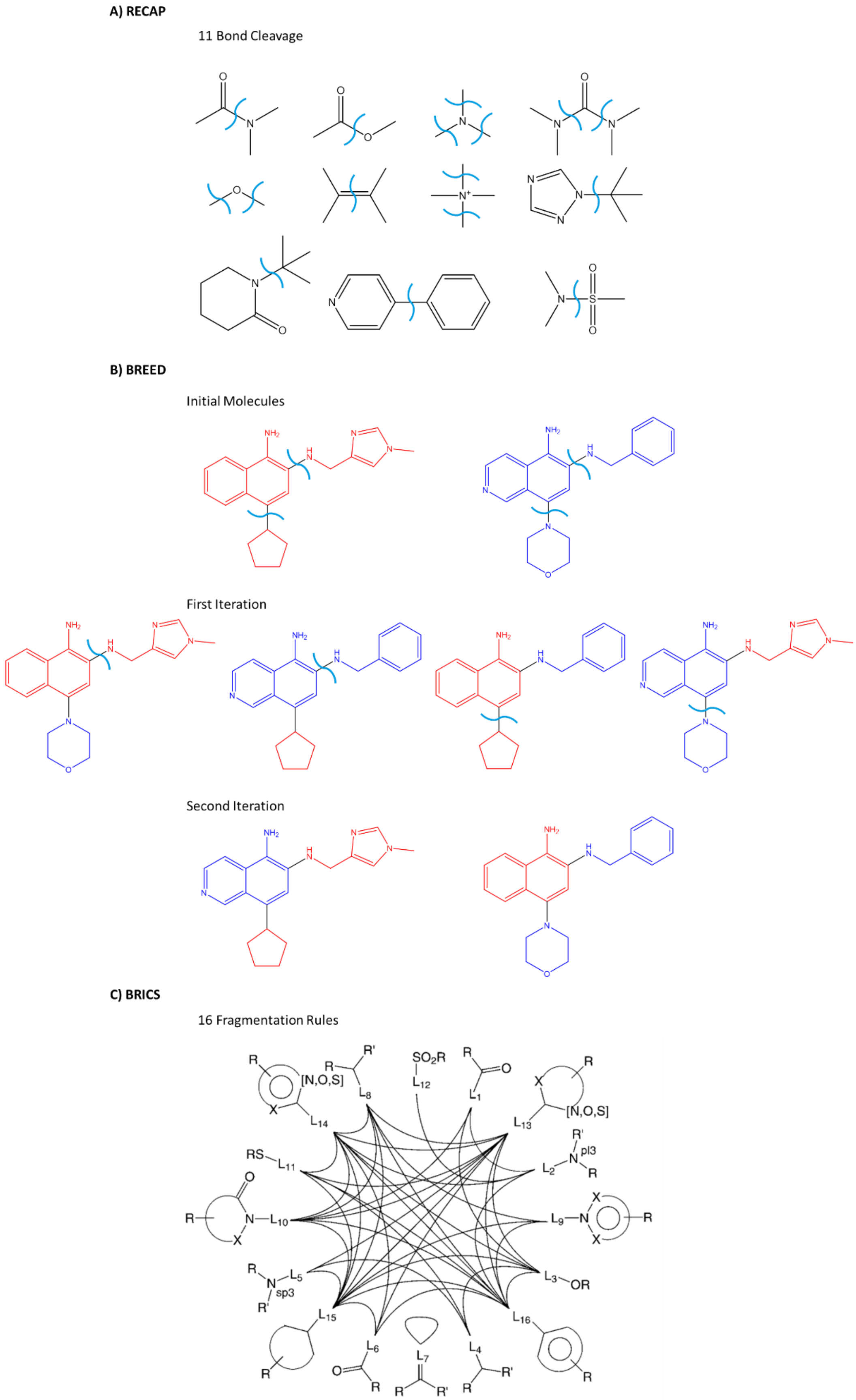
An important branch of chemoinformatics is molecular generation. This is commonly known as de novo design. De novo design techniques generate new molecule ideas that commonly go through a filtering and assessment process to determine which should be taken forward for synthesis. There are two ways in which molecular generation is typically achieved, by either an atom-based (Bohacek & McMartin, 1994; Nishibata & Itai, 1991) or a fragment-based construction approach (Dossetter et al., 2013; P. Polishchuk, 2020). More recently, deep learning approaches have been developed which are typically atom-based methods (Jin et al., 2018; Lim et al., 2020).

The only current approach that takes advantage of reduced graph (RGs) in their approach to molecular generation is that of Pogány et. al (Pogány et al., 2019). RG2SMI is a seq-to-seq deep learning approach where an input RG is used to generate an arbitrary number of molecules that are represented by the RG. Initially, a neural network has to learn the relationship between a SMILES structure and a RG SMILES structure. Therefore, the resulting molecule does not always reflect the RG.

This chapter describes a new fragment-based approach to de novo design where the fragments are defined according to the RG generation rules and correspond to RG nodes. The method involves a pre-processing step in which RGs are generated for a set of molecules. The substructures associated with each node are then aggregated by node type to form a node-substructure dictionary where there is a one-to-many relationship between nodes and substructures. New molecules are then generated based on an input molecule that is first converted to a RG. A node of interest is selected and alternative substructures are extracted from the dictionary. This process can be repeated for any number of nodes in the input molecule. The method is inspired by a number of existing approaches that apply fragmentation rules and then replace part of a molecule with different fragments, for example, RECAP (Lewell et al., 1998), BREED (Pierce et al., 2004) and BRICS (Degen et al., 2008). This chapter first describes the molecular generation methods and then presents several applications of the methods to generate new molecules.

8.2 Methodology

RECAP, BREED and BRICS are existing methodologies that demonstrate how structural information can be taken advantage of to suggest new molecules. All methods are based on fragmenting existing molecules and then recombining the fragments to generate new molecules. RECAP generates fragments by cleaving molecules at 11 chemical bond types which have been derived according to common chemical reactions, shown in Figure 8-1a. The number of attachment points for each fragment is found. Building blocks with the same number of attachment points can then be exchanged (Lewell et al., 1998). BREED is an iterative process that aligns two molecules and finds bonds that match, which fit certain requirements. The bonds are then broken iteratively and the fragments are recombined by swapping them with the opposite molecule to generate new molecules, Figure 8-1b (Pierce et al., 2004). BRICS further develops the principles in RECAP and breaks molecules according to retrosynthetic rules, shown in Figure 8-1c (Degen et al., 2008).



A) RECAP

11 Bond Cleavage

B) BREED

Initial Molecules

First Iteration

Second Iteration

C) BRICS

16 Fragmentation Rules

Figure 8-1: A) RECAP bond cleaving rules, adapted from (Lewell et al., 1998)*. B) Illustration of the BREED process. The overlapping bonds are shown in black and the cuts to make the fragments are shown with the blue line, adapted from* (Pierce et al., 2004)*. C) BRICS fragmentation prototypes taken from* (Degen et al., 2008)

The approach developed within this chapter takes advantage of RGs and uses the RG generation rules and the resulting nodes to define fragmentation points in a molecule. The substructural fragment represented by a node can then be replaced by a different fragment as long as it is the same node type. The fragmentation process using RGs is illustrated in Figure 8-2, using one of the molecules from the BREED explanation. Each node is highlighted in a solid colour and connecting or shared bonds and atoms are highlighted in a checked colour formed from the individual colour of each node. The checked bonds are broken and each connection point to an adjacent node is retained using a wild atom.

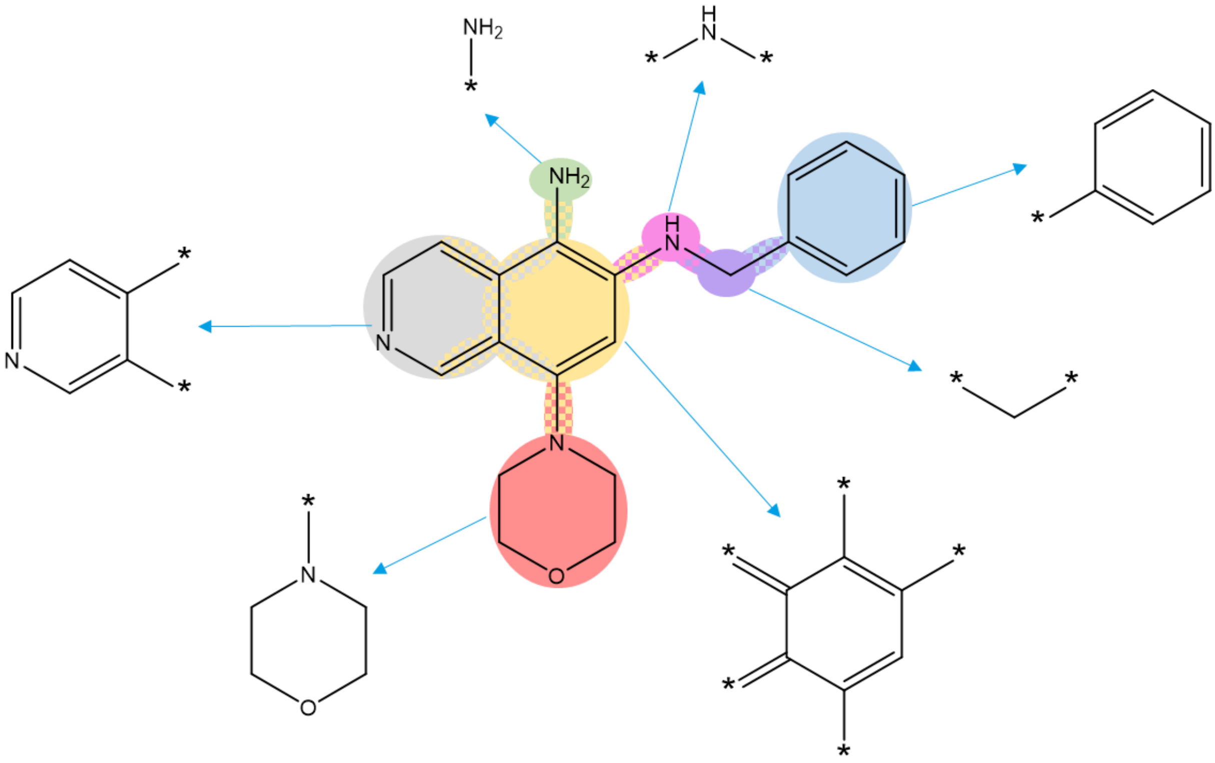


Figure 8-2: Molecule fragmentation process using RG nodes

The replacement fragments must be of the same node type, have the same number of connection points and have compatible bond types. The new fragment must have the same number of connection points so that the molecule can be fully connected once the substructure is replaced. Also, fragments that have multiple connection points could potentially be connected in multiple different ways. This applies to all nodes other than terminal nodes, which only have one connection point. A simplistic example of how a replacement substructural fragment can generate several different molecules due to there being more than one connection point is shown in Figure 8-3. The cyclohexane fragment has three different connection points, leading to the generation of six unique combinations. R groups represent parts of the input molecule that do not change and which the replacement fragment should connect to.

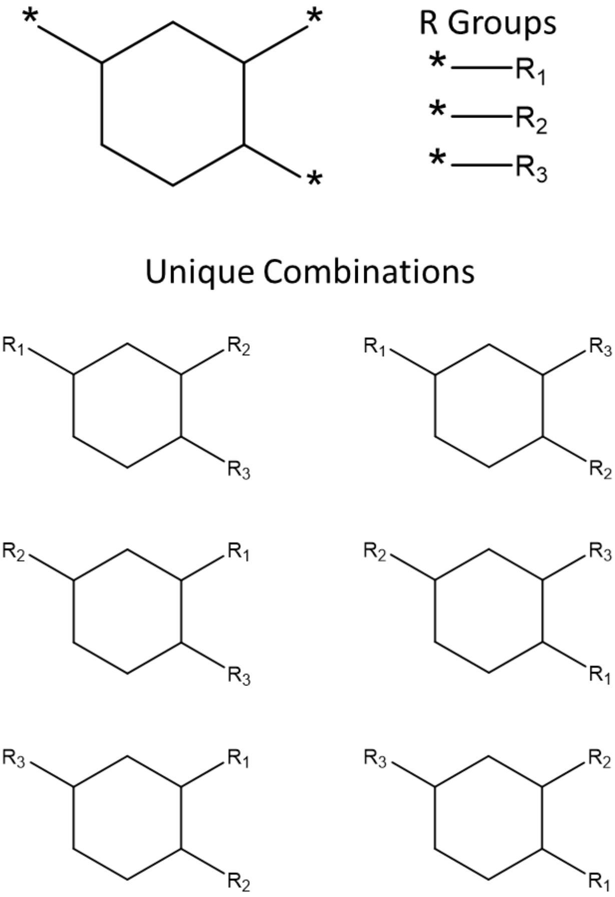


Figure 8-3: Different combinations for one substructural fragment

All six combinations are valid for this example as all connection points are compatible bonds, i.e., they are all single bonds. Figure 8-4 shows a different example where one of the connection points of the cyclohexane ring is a double bond, and assuming there is one connecting double bond in the input molecule, then there would only be two possible combinations to generate.

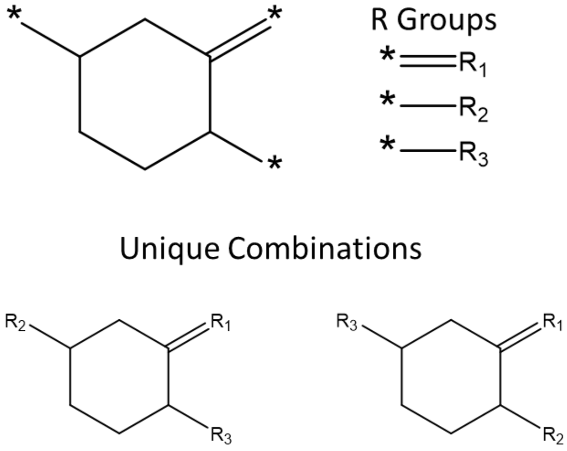
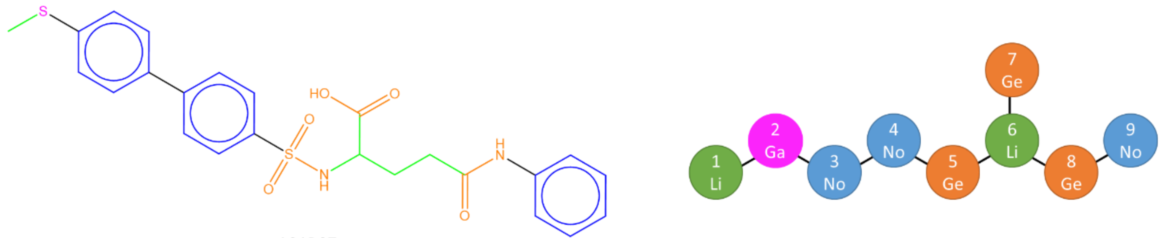


Figure 8-4: An example demonstrating that the bond type is important

In order to identify appropriate substructural fragments, a node-substructure dictionary has to be created. This is a pre-processing step with the dictionary used as a lookup during the generation phase of the molecular generation process. For the experiments presented later in this chapter, all ten previously studied datasets have been used to construct node substructural fragment dictionaries so that there is one tailored to each LO series. (Generic dictionaries have also been created using the ChEMBL and Zinc databases.) A dictionary is generated by first representing a set of molecules by RGs and then extracting the substructures for each node and each molecule. The substructures for a given node type are aggregated and each substructure is annotated with the number of connection points. An example of the substructural fragments for each node along with the number of neighbours is shown in Figure 8-5. Molecule A21B07 is from the MMP12 dataset and is the most potent molecule in the dataset with a pIC50 value of eight.



A21B07

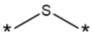
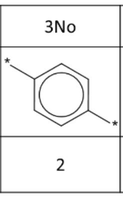
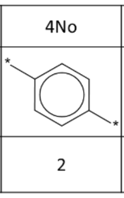
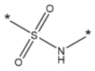
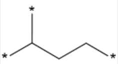
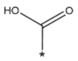
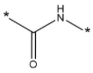
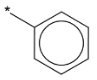
	1Li	2Ga	3No	4No	5Ge	6Li	7Ge	8Ge	9No
Substructural Fragment									
Number of Neighbours	1	2	2	2	2	3	1	2	1

Figure 8-5: Molecule A21B07 from MMP12 dataset with RG (parameters linker and linker and complex) and node substructural fragments

Three molecular generation approaches were developed. All three are based on a given input molecule for which an RG is generated. The first method is the exchange of substructures corresponding to a single RG node of the input molecule. The second method is the exchange of substructures for multiple RG nodes of the input molecule. The final method is a full enumeration of substructures for all nodes of the RG. The first two methods represent alternatives to current scaffold hopping methods. The latter provides a method for molecular suggestions to expand the chemical space already explored.

8.2.1 Node-substructure Dictionaries

To be able to generate molecules that can explore new regions of chemical space then, in addition to the tailored dictionaries, a wider range of substructural fragments needs to be used than those already explored in a LO series. Therefore, node-substructure dictionaries were generated from ChEMBL23 and Zinc 20 drug-like and in-stock datasets (Gaulton et al., 2012; Irwin & Shoichet, 2005;

Irwin et al., 2020). As these datasets contain drug molecules then they should provide fragments that could potentially be useful. Both of these datasets were cleaned and desalted before the substructural fragments dictionaries were generated from the RGs of the cleaned molecules. The number of molecules within each of these two datasets is shown in Table 8-1. For each of the LO datasets, a list of unique substructural fragments was also collected for each node type, with the number of substitution sites for each substructural fragment recorded.

Table 8-1: Number of molecules and RGs for ChEMBL and Zinc. Three different RG types were used as described in Chapter 2. These are Default, Linker and Linker and complex

Dataset	Number of Molecules	Number of unique RGs		
		Default	Linker	Linker and complex
ChEMBL	1,636,835	537,933	654,366	753,693
Zinc	6,836,016	760,754	1,121,126	1,389,928

8.2.2 Substructure Replacement based on One Node

The RG of the molecule of interest is generated and one node is selected. The node type and the number of neighbours are then searched within a node-substructure dictionary. The choice of which node-substructure dictionary to use for fragment replacements is left to the user. The retrieved substructural fragments are then considered in turn and used to replace the substructure of the input molecule corresponding to the selected node. The broken bonds in the input molecule must be compatible with the connection bonds of the replacement substructures. If the selected node is one ring in a fused ring system, then the atoms that are within both fused rings must match. All possible combinations of the connection points for each substructural fragment are considered. Figure 8-6 indicates how one fragment can have several different ways of being connected to the rest of the molecule. The green linker node with three different connection points is being replaced. The corresponding substructural fragment has three connection points, labelled 1), 2) and 3). A replacement fragment is also shown which also has three connection points, labelled a), b) and c). There are six different ways in which the replacement substructural fragments could be connected to the connection points. However, only three unique structures can be generated. One combination of connection points is 1-a, 2-b, 3-c. The other two combinations are 1-a, 2-c, 3-b and 1-c, 2-a, 3-b; these will be generated within the molecular generation algorithm. Molecules generated from combinations 1-b, 2-a, 3c; 1-b, 2-c, 3-a and 1-c, 2-b, 3-a are duplicates.

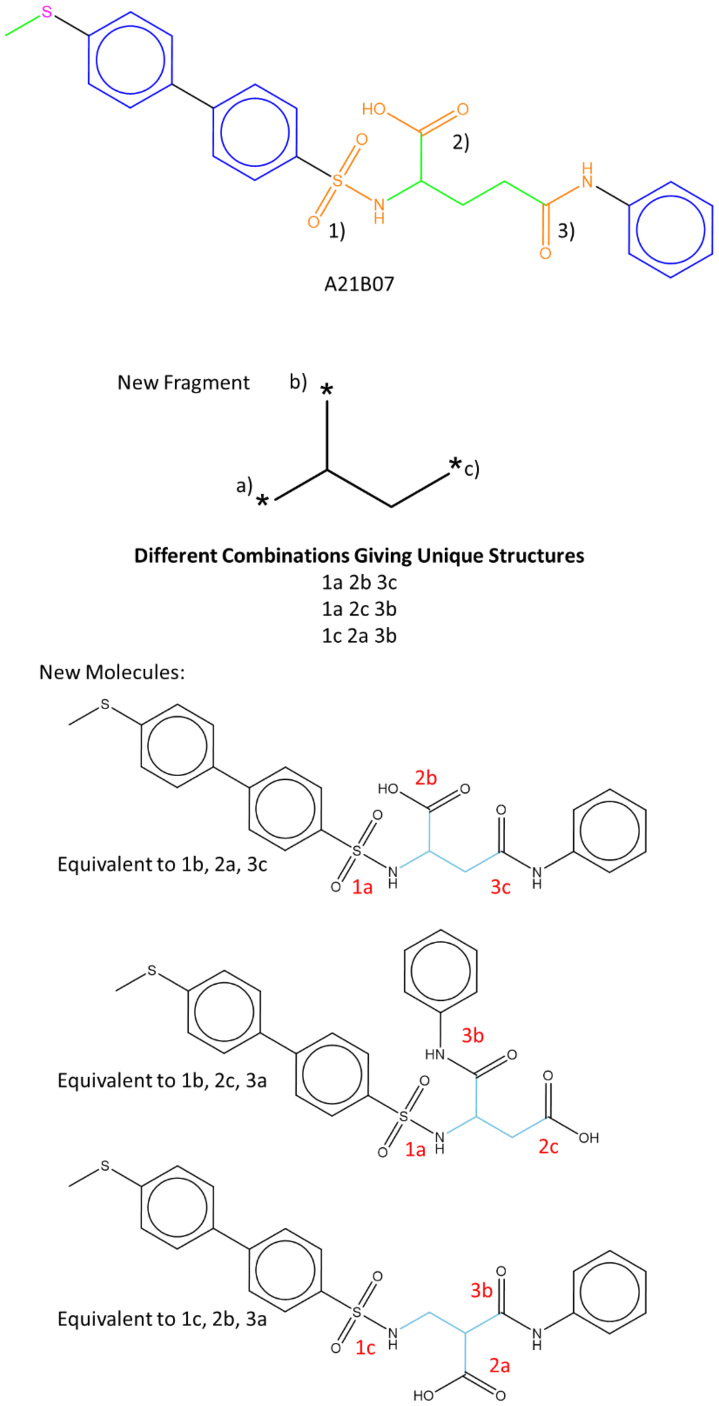


Figure 8-6: An example that one substructural fragment can be connected in different ways

An example of a single node change can be seen in Figure 8-7 based on a small node fragment dictionary. The linker node, Li, represents the substructure to be replaced. Only three substructural fragments are selected as these are the only substructural fragments with three connection points, like the existing fragment. The fragment matching the input fragment is selected as the different orientations of this fragment are also considered. Nine molecules are generated, three for each fragment and then duplicates are deleted as well as generated molecules that are identical to the input molecule. The new molecules are also compared to the molecules within the LO series and if any already exist in the series, they are also removed. The final molecules are validated using RDKit's rdMolStandardize to ensure that they are viable molecules ("RDKit: Open-Source Chemoinformatics," 2018). For this example, six unique molecules are retained: two for substructure a; one for substructure b since the fragment is symmetrical; and three for substructure c.

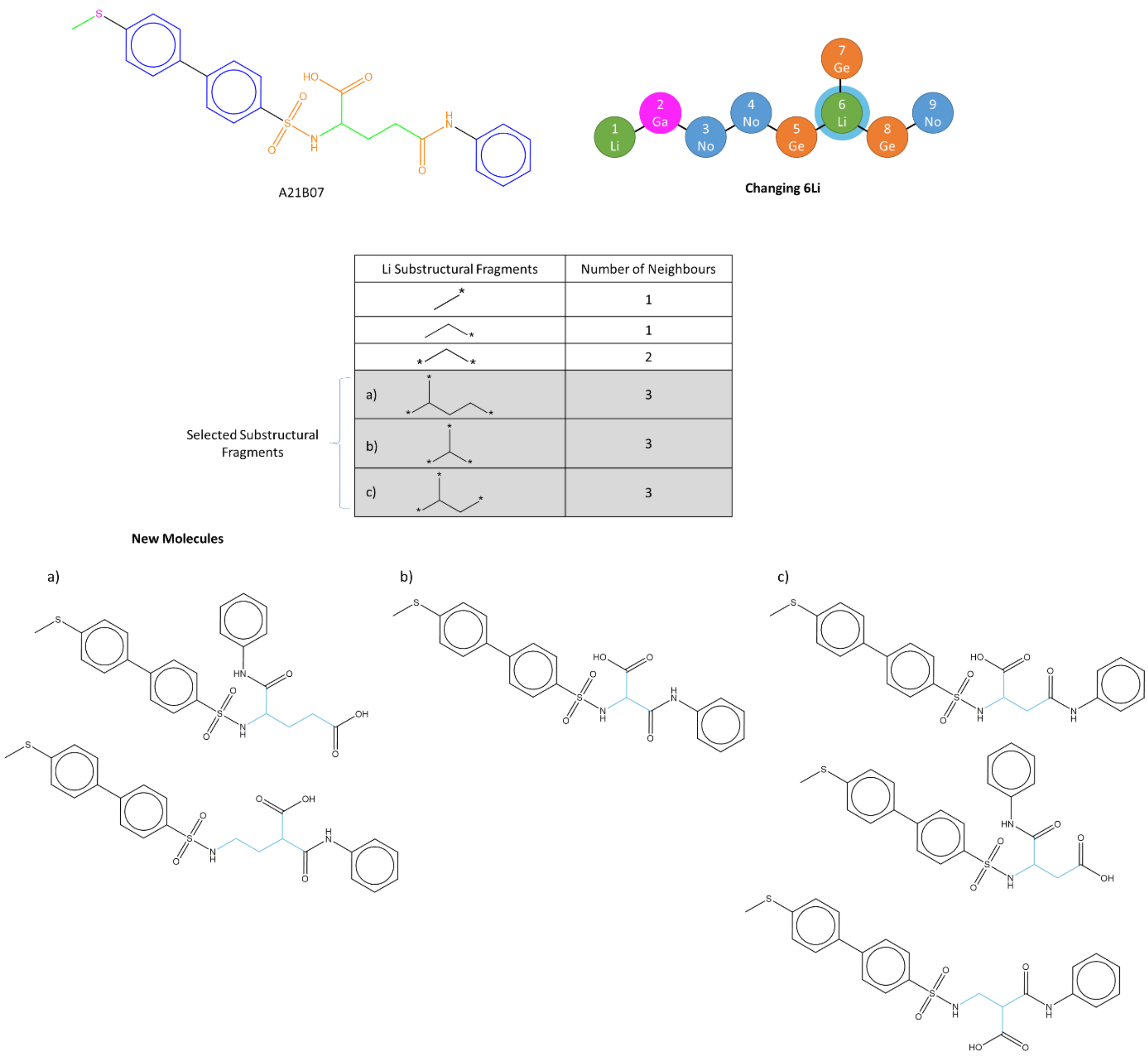


Figure 8-7: An example of a single node, 6Li, replacement for molecule A21B07

Another rule to note when altering nodes is when one of the nodes is within a fused ring, that the length of the fusion path and the atoms must be the same. An example of how the fused ring rules are applied is shown within Figure 8-8. Figure 8-8a is an example of where both the equal path length and the same atoms are met, generating a combined molecule. Figure 8-8b is an example of where the path lengths are not equal, therefore, a new molecule is not generated. Figure 8-8c is an example where the path length is equal, however, the atoms that would be fused are different as the new fragment contains a carbon and an oxygen atom, whereas the remaining molecule it contains two carbons, so cannot be combined.

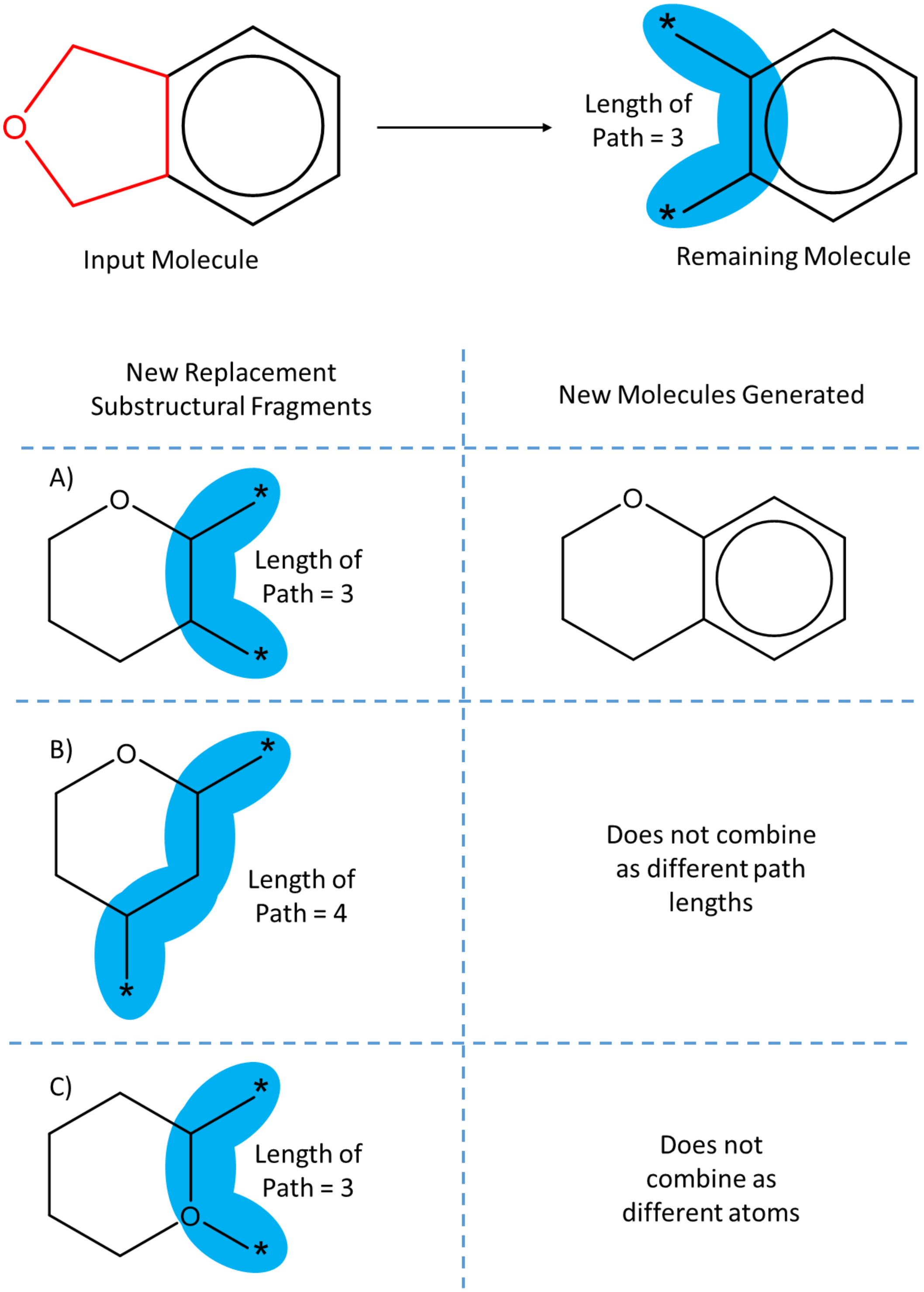


Figure 8-8: Demonstration of a fused ring combination where the length of the paths must be the same. The red highlighted atom in the input molecule is the node to be replaced. A) Is an example of when both rules are met. B) Is an example of when the same path length is not met. C) Is an example of when the same atoms is not met.

8.2.3 Substructure Replacements based on Multiple Nodes

For each node that has been selected for replacement, all substructural fragments that could potentially replace the substructure represented by the node are found. The replacements are then carried out combinatorially. Similar to the single node replacement, duplicates are removed and molecules identical to the input molecule or molecules currently present within the LO series are removed, and the molecules are validated.

An example of a multiple node molecular generation is demonstrated in Figure 8-9. Two nodes are replaced within molecule A21B07; nodes 2Ga and 6Li. There are two replacement substructures for node 2Ga and three for node 6Li giving a total of six substructural fragment combinations. These six combinations generate 12 new and unique molecules through the different ways of connecting the substructural fragments.

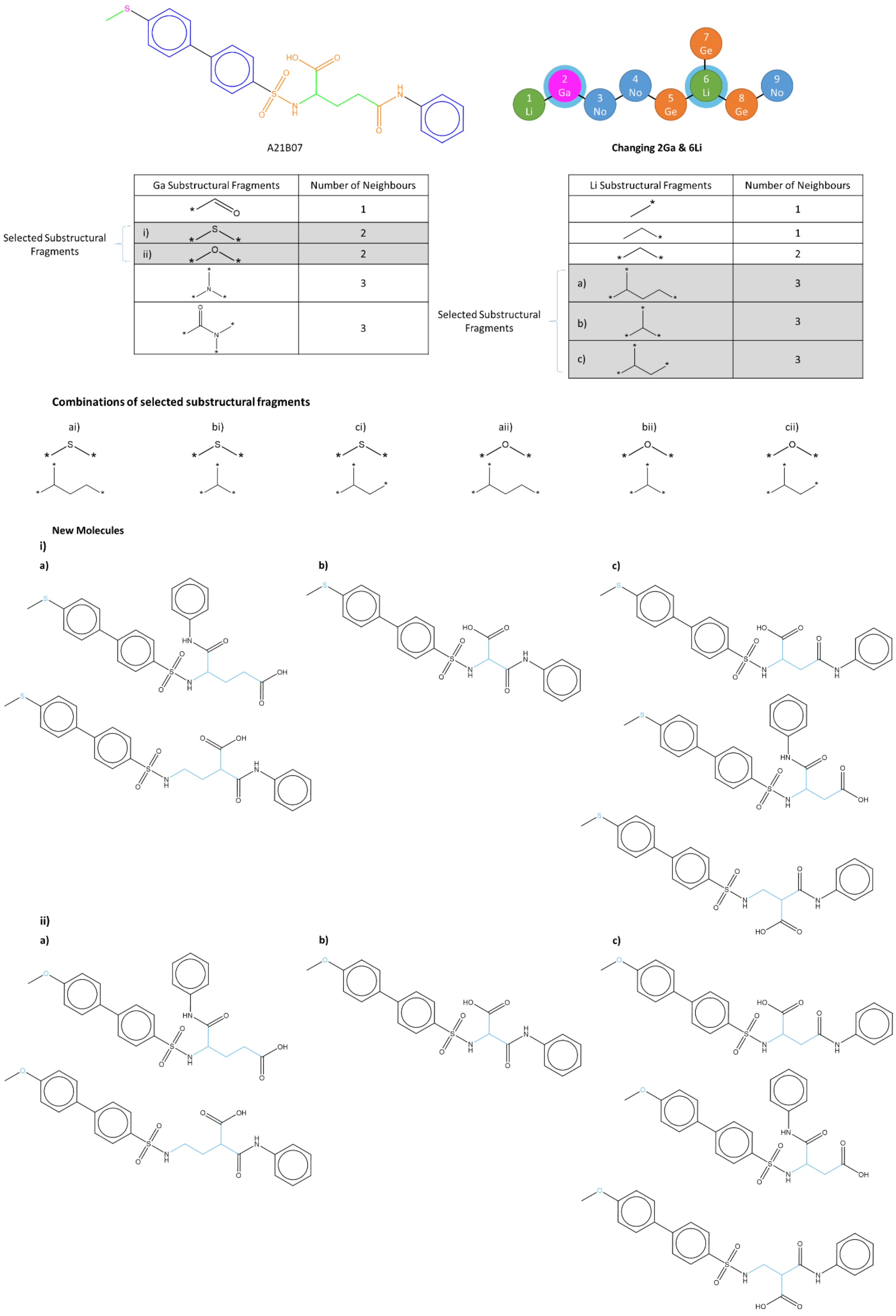
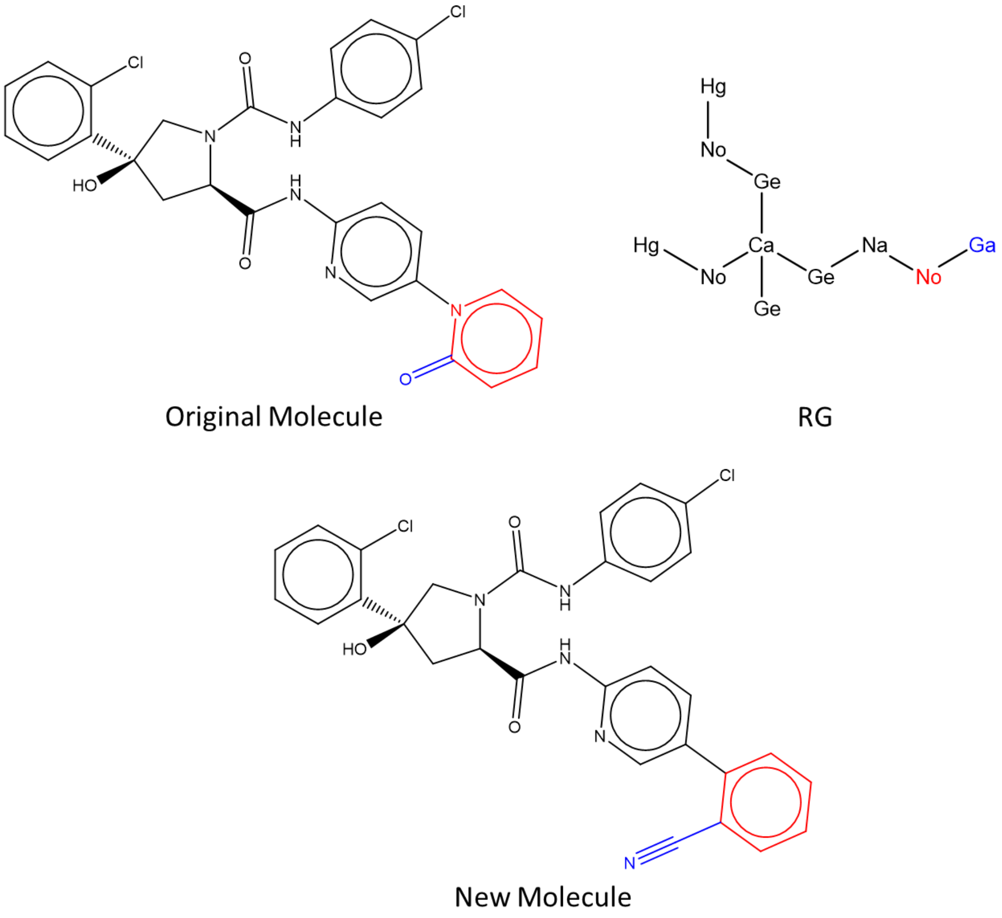


Figure 8-9: An example of a multi node, 2Ga and 6Li, replacement for molecule A21B07

If the two nodes that are being replaced are connecting, then the bond that joins these two nodes does not have to be the same as in the original molecule. Figure 8-10 illustrates how two adjoining nodes that are being altered do not have to retain the original joining bond. This is demonstrated as the carbonyl double bond in the new molecule becomes a single bond. The only requirement is that the connecting bond matches between the two node substructural fragments.



Original Molecule

RG

New Molecule

Figure 8-10: Example of how the bonds between adjoining altering nodes do not have to be retained

8.2.4 Full Enumeration

The final method allows a RG to be input and a full enumeration of all possible and applicable substructural fragments combinations for all nodes is completed. This can be done using the substructural fragments from either ChEMBL, Zinc, or the dataset itself. The user can specify specific substructural fragments for individual or all nodes within the RG being enumerated. A full enumeration provides the chemist with a large exploration of the chemical space. All molecules can then be scored using the exploration and exploitation score generated in the previous two chapters to allow the chemist to select a molecule based on their interests.

8.2.5 Filtering Molecules

In some instances, the number of molecules generated is very large. The molecules need to be filtered to be useful as suggestions to allow an easier and more manageable list of proposed molecules. Four possible ways in which the generated molecules can be filtered have been examined.

The first filtering method is to use the exploration and exploitation scores that were produced in the previous two chapters and set a cut off for each depending on the area of chemical space the user is interested in. If the chemist is more interested in exploration, then a large exploration score cut off would be set and a low exploitation score; and vice versa if they were interested in exploiting the chemical space.

The second filtering method is to set a minimum and maximum change in heavy atom count (HAC). This then allows the generated molecules to be a similar size to the input molecule. The change in HAC filter can only be applied for the single and multiple node experiments as the full enumeration does not start with a molecule.

The third filtering method is to apply a QSAR model and only retain molecules that are above a specified biological activity (pIC50) value. For this chapter, the molecules are filtered at a pIC50 value of 6.5. 6.5 is considered as the active/ inactive threshold as this is in accordance with Bosc et al. (Bosc et al., 2019). Table 8-2 summarises the activity values for the datasets and indicates how they are classified as active and inactive based on this threshold. However, the biological activity threshold can be altered by the user.

Table 8-2: Activity data within all datasets, provided to 2 decimal places

Dataset	Range of pIC50 Values		Mean pIC50	Std	Median pIC50	Classification	
	Minimum	Maximum				Active	Inactive
Bajorath	4.01	11.92	7.48	1.18	7.54	1647	437
CDK2	2.91	9.52	5.92	1.50	5.68	500	867
Chk1	4.76	9.52	7.03	1.10	7.19	72	34
Cyto	4.30	8.89	5.19	0.54	5.10	165	6145
FactorXa	2.97	10.70	7.09	1.31	7.22	1317	639
MMP12	3.70	8.00	5.46	1.03	5.30	380	1324
Neurokinin	4.00	11.00	7.91	1.32	7.94	1248	220
P2x7	4.02	11.00	7.12	0.88	7.10	1368	418

P2x7 Subset	5.90	9.10	7.30	0.74	7.30	593	98
P38α	2.89	10.40	6.76	1.30	6.82	2159	1485

The final filtering method is to use Medchem filters. There are two parts to this filtering. The first is that the molecule should have specific drug-like properties. These properties and associated values can be found in Table 8-3. The second stage is to filter out molecules if they contain certain structural alerts that are considered to be undesirable functional groups. Several different structural alerts can be used, but the ones used within this chapter are the BMS filters (Pearce, Sofia, Good, Drexler, & Stock, 2006). There are 180 alerts in the BMS filters within the Walters implementation (Walters, 2020).

Table 8-3: Medchem property filters

Property	Minimum Value	Maximum Value
Number of hydrogen bond acceptors (HBA)	0	10
Number of hydrogen bond donors (HBD)	0	5
Lipophilicity (LogP)	-5	5
Molecular Weight (MW)	0	500
Topological polar surface area (TPSA)	0	200

8.3 Results

8.3.1 Node-substructure Dictionaries

Node-substructure dictionaries were built using ChEMBL, Zinc and the ten datasets researched within the thesis so far, for RGs generated using the parameter Linker and Complex. This parameter was selected as it incorporates all atoms within the molecule into nodes and it allows a greater differentiation between the terminal nodes than using just the Linker parameter.

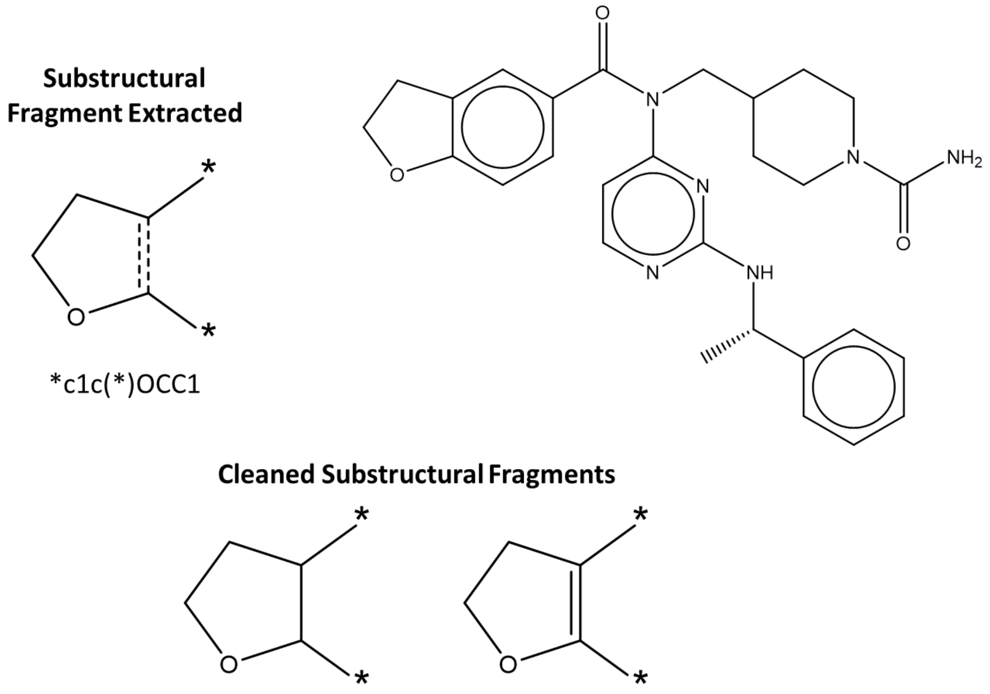

Substructural Fragment Extracted
*c1c(*)OCC1
Cleaned Substructural Fragments

Figure 8-11: Example of how to clean substructural fragments

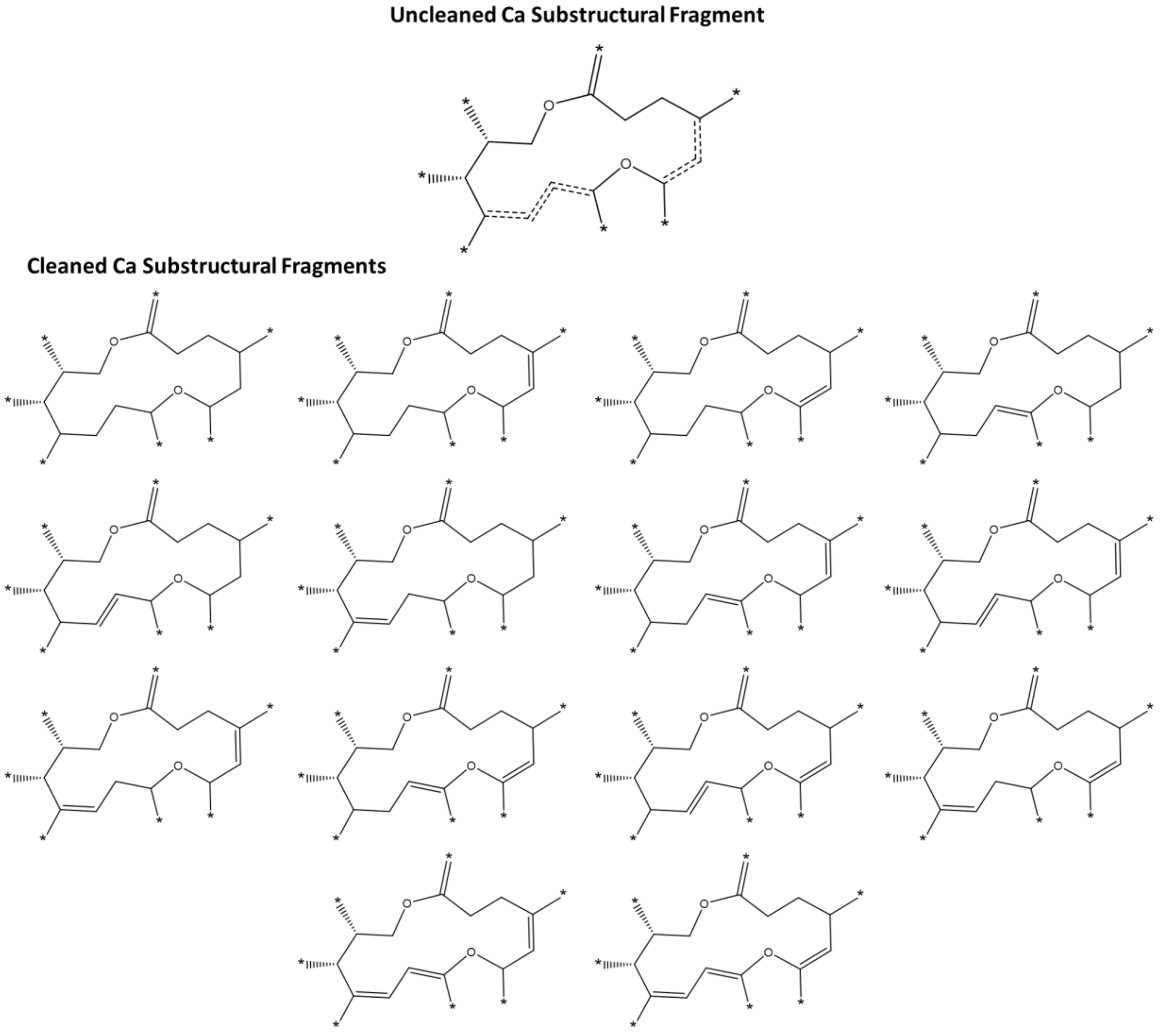
Fragments generated from fused rings where an aromatic ring is fused to an aliphatic ring need to be cleaned. This is because the atoms in the extracted aliphatic ring are a mixture of aromatic and aliphatic and the bond is considered to be aromatic. An example is shown in Figure 8-11, where the dashed bond is considered to be an aromatic bond. The atoms are changed to aliphatic atoms and two separate substructural fragments are generated, one with a single bond and the other with a double bond, if the atoms' valence allows. Within the generation phase, the new bond type formed is not considered for either a fused or non-fused alteration.

Table 8-4: Table showing all of the number of substructural fragments extracted for each node type from ChEMBL and Zinc for RG made with linker and complex parameter

Nodes	ChEMBL	Zinc	Number of substructures that are the same
Acyclic inert - Li	7938	4905	1693
Acyclic HBA - Ga	848	981	372
Acyclic HBD - Gd	20	12	5
Aromatic inert - No	557	546	419
Aromatic HBA- Na	580	549	425
Aromatic HBD - Nd	234	247	171
Aliphatic HBD - Cd	30	9	0
Aliphatic HBA - Ca	2103161	16862	6672
Aliphatic inert - Co	4565603	11879	4565
Acyclic HBA HBD - Ge	2556	2663	990
Aromatic HBA HBD - Ne	200	204	137
Aliphatic HBA HBD - Ce	2282292	11249	2680
Hydrophobic - Hg	2240	1393	525

The number of each type of node generated for each dataset can be seen in Table 8-4 and Table 8-5. Table 8-4 demonstrates the number of unique substructural fragments for each node for ChEMBL and Zinc. It is interesting to note that even though Zinc has a larger number of molecules, there are generally fewer substructures for each node. Additionally, the overlap generally lies between 30% and 80% other than aliphatic HBD, Cd, where no substructural fragments are the same. However, there are very few examples for this node.

Nodes aliphatic HBA, Ca, aliphatic inert, Co, and aliphatic HBA HBD, Ce, generate a vast number of substructural fragments due to the cleaning process of the fragments. Many of the uncleaned fragments contain one or more bonds that need to be cleaned. Additionally, within the ChEMBL dataset, there are several large heterocyclic rings containing several bonds to be cleaned and as the cleaning process generates all combinations of single and double bonds for aromatic fused bonds, one unclean fragment can generate a large number of cleaned fragments. An example is shown in Figure 8-12, where one substructure leads to 14 substructural fragments.


Uncleaned Ca Substructural Fragment

Cleaned Ca Substructural Fragments

Figure 8-12: An example of how one substructural fragment is cleaned

Table 8-5: Table showing all of the number of substructural fragments extracted for each node type for each dataset for RG made with linker and complex parameter

Nodes	Bajorath	CDK2	Chk1	Cyto	FactorXa	MMP12	Neurokinin	P2x7	P2x7 Subset	P38α
Acyclic inert - Li	44	56	15	229	64	17	33	26	8	45
Acyclic HBA - Ga	18	27	10	76	19	8	20	15	6	22
Acyclic HBD - Gd	0	0	0	0	0	0	1	1	0	0
Aromatic inert - No	29	51	12	93	43	12	26	22	12	70
Aromatic HBA- Na	81	77	8	101	57	9	40	62	42	107
Aromatic HBD - Nd	4	22	3	32	9	1	5	4	2	17
Aliphatic HBD - Cd	0	0	0	1	0	0	0	0	0	0
Aliphatic HBA - Ca	91	107	12	1369	125	5	128	83	34	125

Aliphatic inert - Co	27	53	8	793	35	2	57	47	6	63
Acyclic HBA HBD - Ge	20	34	12	101	62	8	28	17	5	27
Aromatic HBA HBD - Ne	12	16	6	19	11	1	15	15	1	23
Aliphatic HBA HBD - Ce	29	502	9	204	38	0	58	26	3	58
Hydrophobic - Hg	28	22	7	76	23	6	16	22	13	32

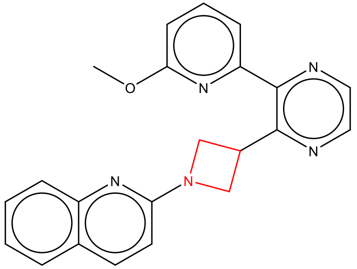
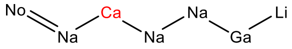
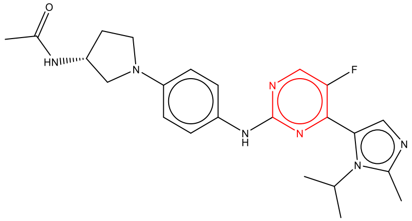
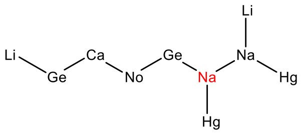
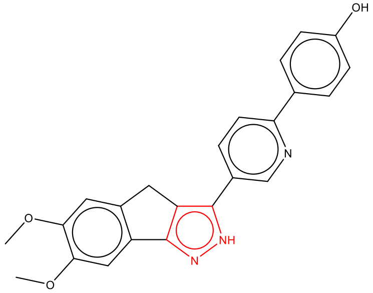
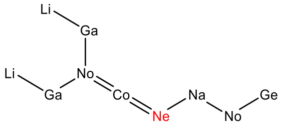
Table 8-5 demonstrates the number of unique substructural fragments for each node for the ten datasets that have been examined in this thesis. There are a varying number of substructural fragments for each of the nodes across the datasets. For aromatic HBD, Gd, there is only one example across all of the datasets which is seen in the Neurokinin and P2x7 datasets, and only one example of the aliphatic inert, Cd, node, which is for the Cyto dataset.

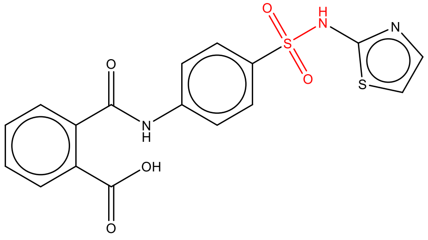
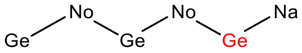
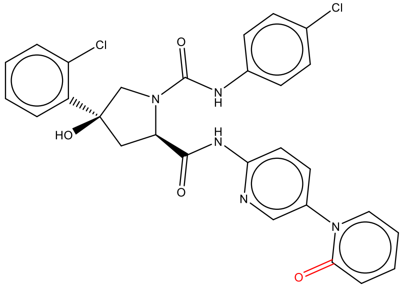
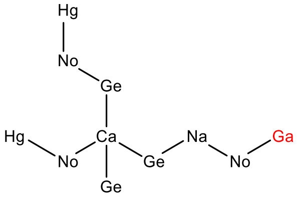
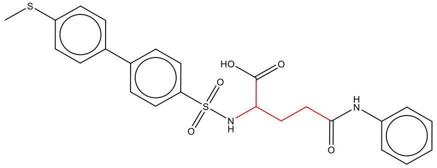
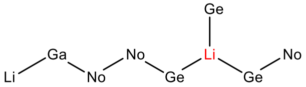
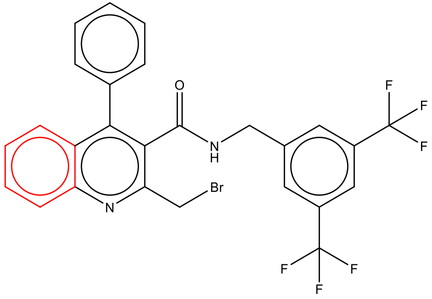
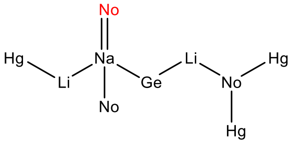
8.3.2 Substructure Replacement based on One Node

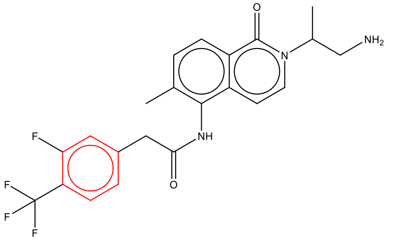
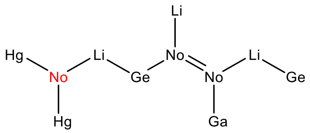
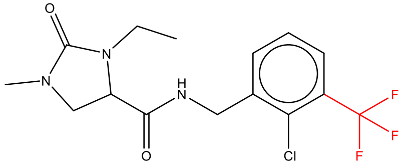
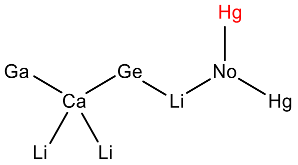
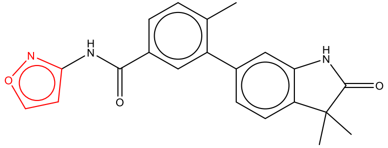
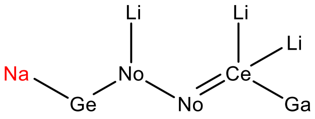
For each of the datasets, the molecule with the highest pIC50 value was selected. One random node was selected to be replaced and the results from the three different substructural fragments dictionaries, ChEMBL, Zinc and the dataset itself are reported in Table 8-6. For each dataset, the node and corresponding substructure that has been replaced is highlighted in red. In all cases, 100% of the generated molecules were valid.

For each dataset, two of the generated molecules were selected to display when using the node-substructure dictionary generated from the LO series, other than for Neurokinin and FactorXa, which instead are from the ChEMBL node-substructure dictionary. These molecules can be seen in Figure 8-13. Some observations based on these follow. For Chk1, there are no new substructures in the node-substructure dictionary, but two new molecules are generated by changing the connection points of the highlighted substructure. A similar effect is seen in one of the Cyto examples where the existing sulfonamide has been flipped. The FactorXa examples indicate that the bond types between the nodes are retained as only substructural fragments with a connecting double bond are used. Finally, no new molecules are generated for the Neurokinin dataset when the substructural fragments are used from just the dataset. This is because none of the substructural fragments have the same shortest path as the fused atoms in the remaining part of the input molecule.

Table 8-6: Results of the most active molecule from each dataset undergoing single node transformation

Dataset	**Molecule**	**RG**	**Number of Molecules Generated Using…**					
			Dataset		**ChEMBL**		**Zinc**	
			Replacement Fragments	**Number of Molecules Generated**	**Replacement Fragments**	**Number of Molecules Generated**	**Replacement Fragments**	**Number of Molecules Generated**
Bajorath	 CHEMBL3691541 (pIC50 = 11.92)		29	38	1054	1350	1004	1197
CDK2	 50422965 (pIC50 = 9.52)		25	104	180	429	185	375
Chk1	 Chk1N144 (pIC50 = 9.52)		1	3	55	11	67	17

Cyto	 4806 (pIC50 = 8.89)		42	59	1055	1626	1160	1666
FactorXa	 50266775 (pIC50 = 10.70)		6	1	77	8	70	13
MMP12	 A21B07 (pIC50 = 8.00)		6	15	2166	9700	1425	6218
Neurokinin	 CHEMBL281797 (pIC50 = 11.00)		6	0	68	8	58	9

P2x7	 CHEMBL2324343 (pIC50 = 11.00)	 Li Hg Li No Li No Ge No Ge Hg Ga	3	9	120	204	126	229
P2x7 Subset	 CHEMBL2218174 (pIC50 = 9.51)		13	12	2087	1937	1178	1081
P38α	 255896 (pIC50 = 10.40)		18	17	54	53	46	45

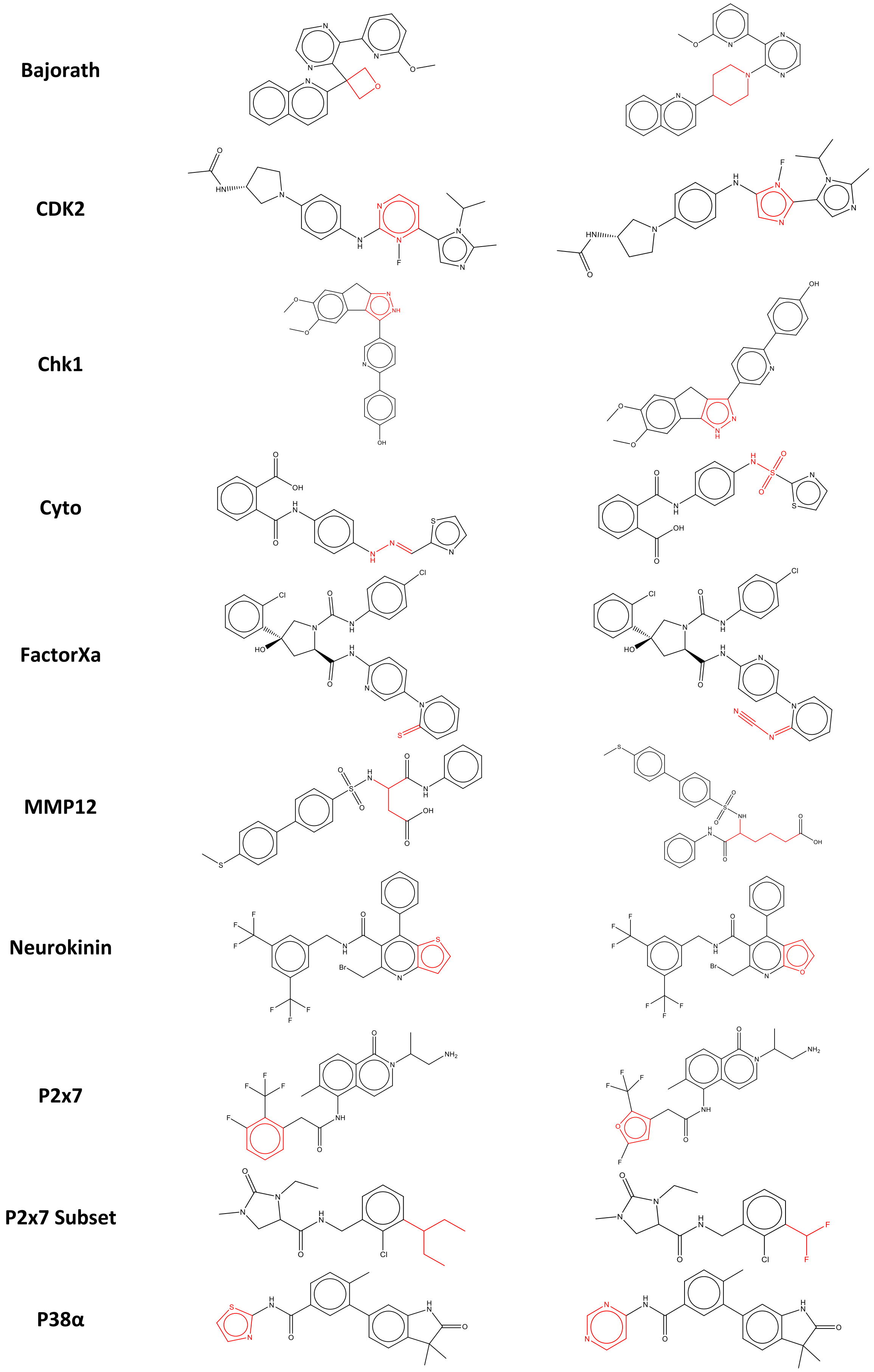


Figure 8-13: Two examples of molecules generated from the single node generation

A comparison was drawn between the molecules generated using the three different node-substructure dictionaries. The number of molecules that are the same is shown in Table 8-7. For all datasets, other than Cyto, the molecules generated from the substructural fragments from the dataset are also all generated when using the substructural fragments the ChEMBL. This is not surprising as all the datasets, except Cyto, are subsets of ChEMBL. Similarly, the molecules generated using substructural fragments from the dataset are also generated using the substructures from Zinc, except for the Bajorath and Cyto datasets. When comparing the molecules generated from ChEMBL and Zinc the overlap is not 100% and this reflects the differences in the node-substructure dictionaries shown in Table 8-4.

The molecules generated from the dataset substructural fragments are within the domain applicability, whereas, the molecules generated from the ChEMBL and Zinc substructural fragments are exploring new areas of chemical space and introducing unseen fragments. To get the largest chemical space coverage, a combination of all three substructural fragments would be preferred. However, a greater number of molecules would be generated, so a filtering process would be needed to narrow down the molecules.

Table 8-7: Comparison of the molecules between each of the methods to establish the overlap

Dataset	**Molecules That Are The Same in…**		
	Dataset		**ChEMBL**
	ChEMBL	**Zinc**	**Zinc**
Bajorath	38	31	616
CDK2	104	104	300
Chk1	3	3	11
Cyto	56	52	708
FactorXa	1	1	4
MMP12	15	15	1813
Neurokinin	-	-	6
P2x7	9	9	171
P2x7 Subset	12	12	440
P38α	17	17	44

Additionally, on closer inspection of the molecules generated there are some peculiar molecules created. Two are identified in Figure 8-14. Molecule a) is an example of how the size of the fragment can significantly change. A small ring of size four has been replaced by a macrocycle of size 19. Additionally, some of the molecules are peculiar, one is identified in Figure 8-14b. The bonds that are dashed indicate that these are aromatic bonds, therefore, this is an aromatic ring with a triple bond also included within this ring. This is considered valid as it can be read into a molecule object in RDKit. Numerous molecules contain this fragment, with an example from Zinc shown in Figure 8-15. Due to these factors and the vast number of molecules generated in some instances the molecules go through a filtering process. Of the filtering processes examined within this chapter, the only method that removes molecule Figure 8-14b is the medchem filters as it would be filtered out due to *primary_halide_sulfate > 0* filter*,* [CH2][Cl,Br,I,$(O(S(=O)(=O)[!$(N);!$([O&D1])]))] which matches to the -CH_2Br*,* and the current MW filter.

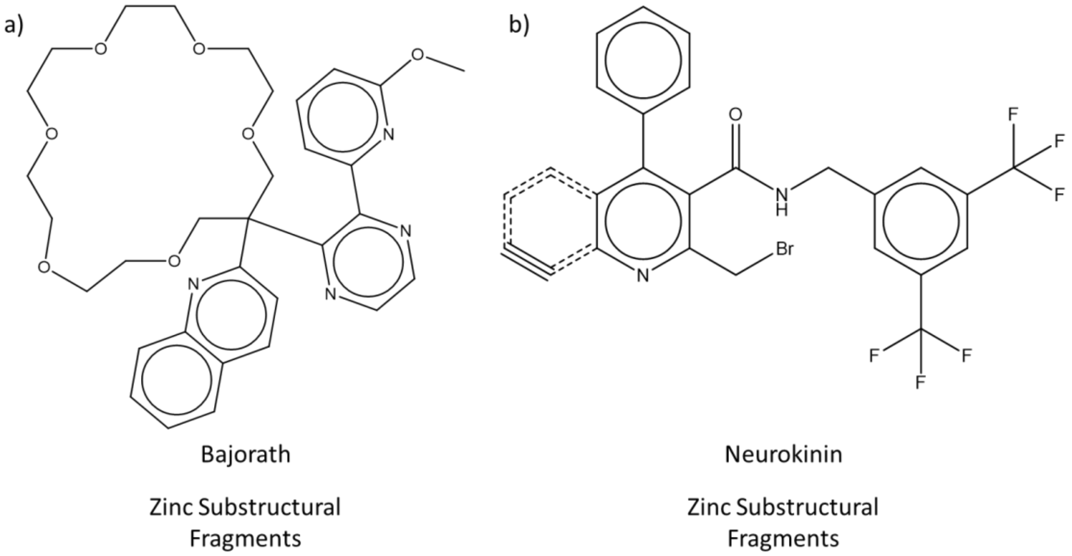


Figure 8-14: Two peculiar molecules identified from the molecular generation process. a) from Bajorath dataset and b) from Neurokinin both created with Zinc substructural fragments

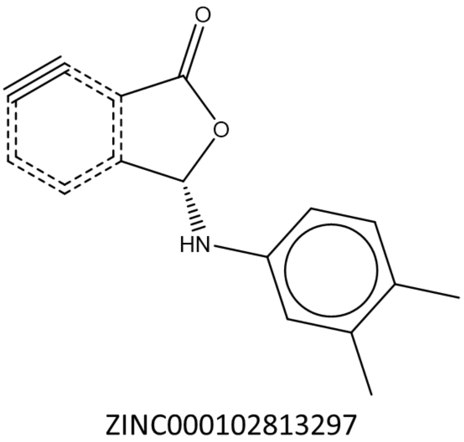


*Figure 8-15: Molecule contained with Zinc with the peculiar fragment, *c1c#cccc1**

8.3.2.1 Filtering Molecules

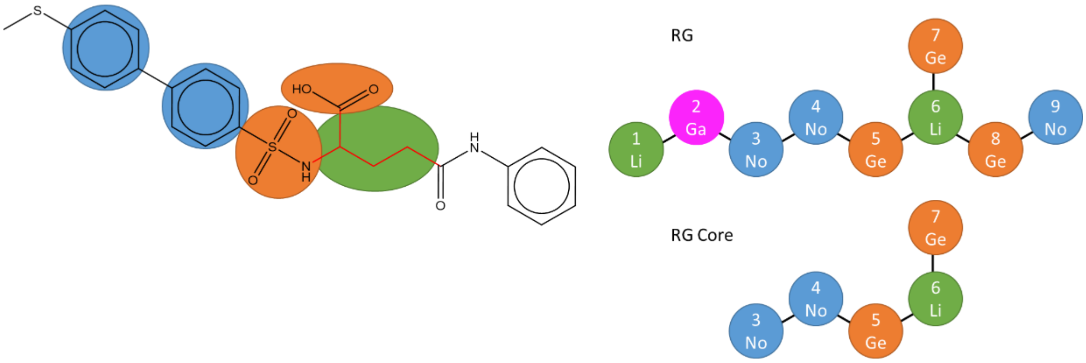
All the molecules generated in the above section then undergo the four filtering processes.

8.3.2.1.1 Exploration and Exploitation Score

The exploration and exploitation scores from the previous chapter can be used to filter out molecules that do not fit the potential exploration or exploitation goals that the chemist is after. Given that the two scores are based on the RG cores, these methods are applicable to the replacement of nodes in the RG core only.

The use of the exploration and exploitation scores is demonstrated using molecule A21B07 from the MMP12 dataset. The MMP12 dataset is represented by a single RG core as described in Chapter 3 and the exploration and exploitation scores for each substructural fragment in the RG core are displayed in Figure 8-16. These were derived using the methods described in Chapter 6 and 7. Table 8-8 illustrates the exploration and exploitation scores for the 15 molecules generated by replacing the linker node referred to earlier and using the MMP12 node-substructure dictionary. In each case, the overall molecule score is shown as well as each of the individual node scores. None of these new molecules generated has an unseen linker substructure within this node. Given that the scores are agnostic of how the substructures are connected, then the node scores are assigned according to substructures in the LO series. Therefore, multiple molecules can achieve the same score. The first new molecule generated receives the highest exploration score as the RG core substructures are relatively rare in the LO series. The contributions that are made by the other nodes/ substructures that are unchanged are the same across all 15 new molecules. The exploitation scores do not produce the reverse ordering of the exploration score because this score is dependent on how important the substructure at that node position is to the biological activity and not the distribution of how many times it has been observed. Once again, the four unchanged node contributing scores remain the same throughout the new molecules.

However, this was not the case for the substructure replacements to molecule 50266775 from the FactorXa dataset since the node that has been replaced is not part of the RG core. As all the nodes within the RG core remain the same, the exploration and exploitation scores are the same and do not allow a way to distinguish between or filter molecules. Therefore, this filtering method could only be used for molecules that have alterations to the RG core nodes. Additionally, the filtration could be on either just the exploration score or exploitation score or could be of both.



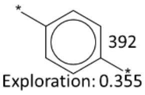
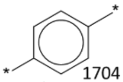
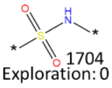
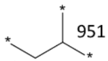
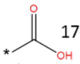
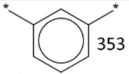
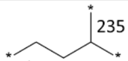
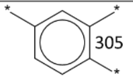
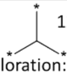
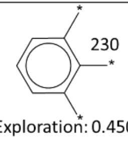
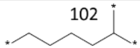
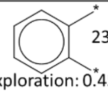
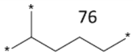
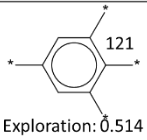
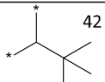
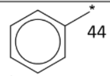
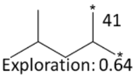
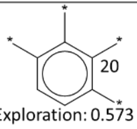
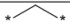
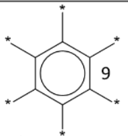
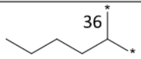
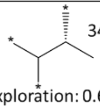
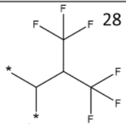
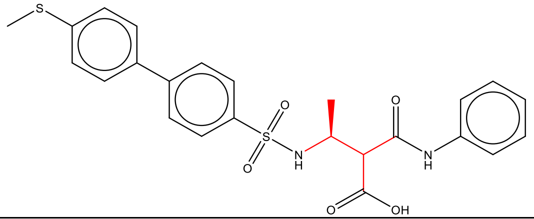
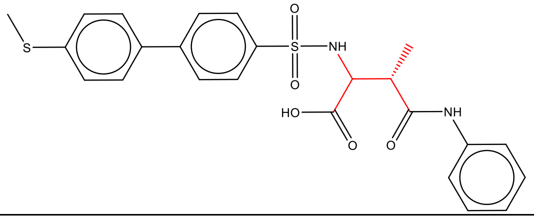
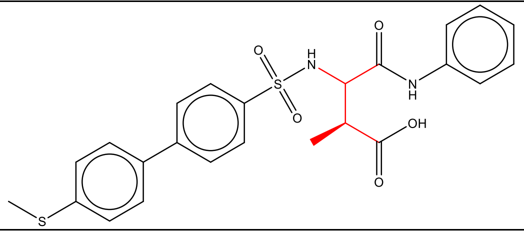
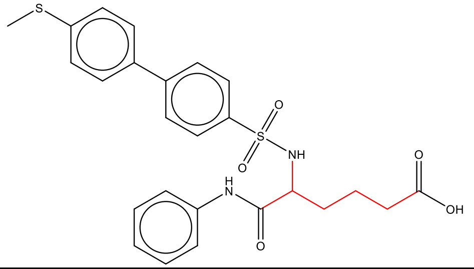
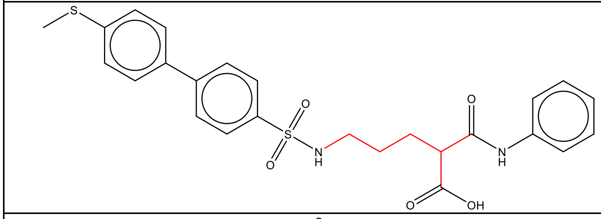
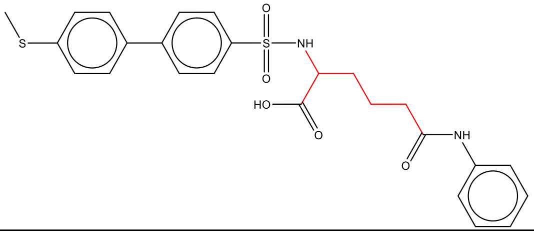
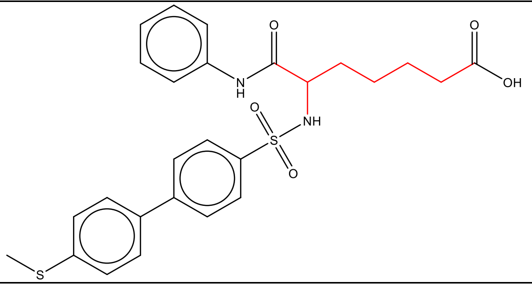
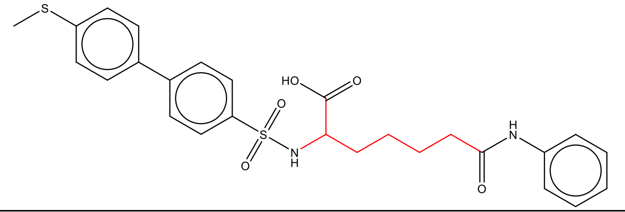
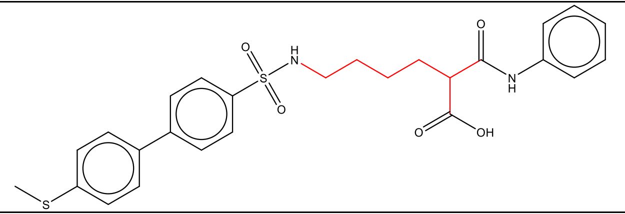
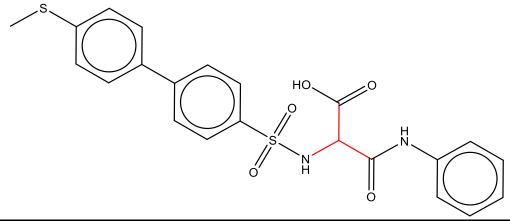
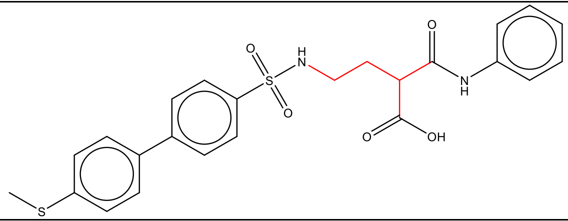
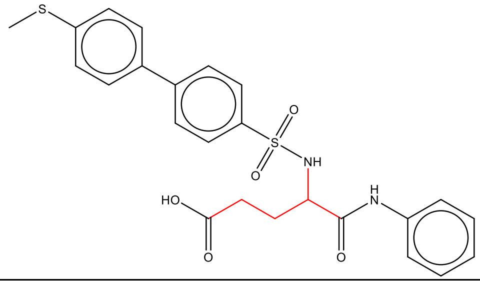
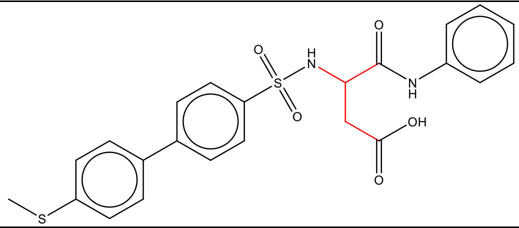
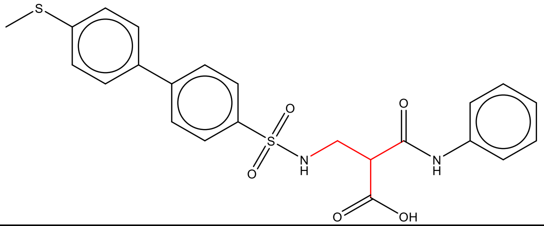
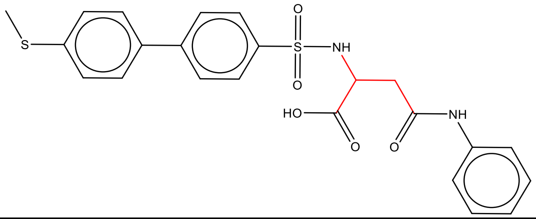
3No	4No	5Ge	6Li	7Ge
 392 Exploration: 0.355 Exploitation: 1.037	 1704 Exploration: 0 Exploitation: 0.358	 1704 Exploration: 0 Exploitation: 0.169	 951 Exploration: 0.114 Exploitation: 0.169	 1704 Exploration: 0 Exploitation: 0.103
 353 Exploration: 0.378 Exploitation: -0.370	New Substructural Fragment Exploration: 1 Exploitation: 0	New Substructural Fragment Exploration: 1 Exploitation: 0	 235 Exploration: 0.534 Exploitation: 0.563	New Substructural Fragment Exploration: 1 Exploitation: 0
 305 Exploration: 0.406 Exploitation: -0.150			 122 Exploration: 0.600 Exploitation: -0.032	
 230 Exploration: 0.450 Exploitation: -0.866			 102 Exploration: 0.612 Exploitation: 0.341	
 230 Exploration: 0.450 Exploitation: -0.910			 76 Exploration: 0.627 Exploitation: 0.338	
 121 Exploration: 0.514 Exploitation: -0.234			 42 Exploration: 0.647 Exploitation: 0.522	
 44 Exploration: 0.559 Exploitation: 0.474			 41 Exploration: 0.648 Exploitation: 0.521	
 20 Exploration: 0.573 Exploitation: -1.141			 37 Exploration: 0.650 Exploitation: -0.206	
 9 Exploration: 0.580 Exploitation: -1.254			 36 Exploration: 0.651 Exploitation: 0.428	
New Substructural Fragment Exploration: 0.585 Exploitation: 0			 34 Exploration: 0.652 Exploitation: -0.317	
			 28 Exploration: 0.656 Exploitation: 0.034	
			New Substructural Fragment Exploration: 0.672 Exploitation: 0	

Figure 8-16: MMP12 RG core substructural fragment

Table 8-8: Table of exploration and exploitation scores for the molecules generated from A21B07 and the MMP12 node-substructure dictionary

New Molecule	Exploration Score	Exploitation Score
	0.201 [0.355, 0, 0, 0.652, 0]	0.270 [1.037, 0.358, 0.169, -0.317, 0.103]
	0.201 [0.355, 0, 0, 0.652, 0]	0.270 [1.037, 0.358, 0.169, -0.317, 0.103]
	0.201 [0.355, 0, 0, 0.652, 0]	0.270 [1.037, 0.358, 0.169, -0.317, 0.103]
	0.196 [0.355, 0, 0, 0.627, 0]	0.401 [1.037, 0.358, 0.169, 0.338, 0.103]
	0.196 [0.355, 0, 0, 0.627, 0]	0.401 [1.037, 0.358, 0.169, 0.338, 0.103]
	0.196 [0.355, 0, 0, 0.627, 0]	0.401 [1.037, 0.358, 0.169, 0.338, 0.103]
	0.193 [0.355, 0, 0, 0.612, 0]	0.402 [1.037, 0.358, 0.169, 0.341, 0.103]

	0.193 [0.355, 0, 0, 0.612, 0]	0.402 [1.037, 0.358, 0.169, 0.341, 0.103]
	0.193 [0.355, 0, 0, 0.612, 0]	0.402 [1.037, 0.358, 0.169, 0.341, 0.103]
	0.191 [0.355, 0, 0, 0.600, 0]	0.327 [1.037, 0.358, 0.169, -0.032, 0.103]
	0.178 [0.355, 0, 0, 0.534, 0]	0.446 [1.037, 0.358, 0.169, 0.563, 0.103]
	0.178 [0.355, 0, 0, 0.534, 0]	0.446 [1.037, 0.358, 0.169, 0.563, 0.103]
	0.094 [0.355, 0, 0, 0.114, 0]	0.367 [1.037, 0.358, 0.169, 0.169, 0.103]
	0.094 [0.355, 0, 0, 0.114, 0]	0.367 [1.037, 0.358, 0.169, 0.169, 0.103]
	0.094 [0.355, 0, 0, 0.114, 0]	0.367 [1.037, 0.358, 0.169, 0.169, 0.103]

8.3.2.1.2 Change in HAC

The generated molecules were filtered using maximum and minimum change in HAC which were set to -3 and 3. The numbers of molecules filtered out are shown in Table 8-9. There is very little effect for the molecules derived from the substructures in the LO datasets, therefore, this shows that there is less fluctuation in the HAC when using those substructural fragments. For some of the datasets, no molecules are filtered when using the substructures derived from ChEMBL or Zinc. This suggests that this filter should be fine-tuned for each individual dataset.

Table 8-9: Number of molecules from the single node change that were filtered out for each dataset using the change in HAC filter

Dataset	**Number of Molecules Filtered Out…**		
	Dataset	**ChEMBL**	**Zinc**
Bajorath	0 [0%]	128 [9.5%]	126 [10.5%]
CDK2	0 [0%]	0 [0%]	0 [0%]
Chk1	0 [0%]	0 [0%]	0 [0%]
Cyto	3 [5.1%]	510 [31.4%]	550 [33.0%]
FactorXa	0 [0%]	0 [0%]	0 [0%]
MMP12	0 [0%]	7057 [72.8%]	4406 [70.9%]
Neurokinin	-	0 [0%]	0 [0%]
P2x7	0 [0%]	0 [0%]	0 [0%]
P2x7 Subset	0 [0%]	1476 [76.2%]	689 [63.7%]
P38α	0 [0%]	0 [0%]	0 [0%]

8.3.2.1.3 QSAR Filter

The QSAR models that were trained for each of the datasets from the previous chapter were used to produce predictions for all generated molecules. The SVM models were used for all datasets, as this method typically produces the best model. Molecules that have a predicted pIC50 value of less than 6.5 were filtered out. The filtered results can be seen in Table 8-10. There are only three datasets where any of the generated molecules are filtered. These three datasets are the only datasets where the molecules used to generate the models have a mean and median pIC50 value below 6.5, and they all have a high inactive to active ratio, which is the reverse for all of the other seven datasets. For the Cyto dataset, nearly 100% of the molecules are filtered. This is due to the Cyto model having the lowest mean and median pIC50 activity values and the lowest active to inactive ratio. Therefore, if the QSAR filter technique were to be implemented, it would have to be either based on a user-defined threshold or the threshold could be determined using all the predictions that are calculated for the new molecules.

Table 8-10: Number of molecules from the single node change that were filtered out for each dataset using the QSAR model

Dataset	**Number of Molecules Filtered Out…**		
	Dataset	**ChEMBL**	**Zinc**
Bajorath	0 [0%]	0 [0%]	0 [0%]
CDK2	0 [0%]	25 [5.8%]	32 [8.5%]
Chk1	0 [0%]	0 [0%]	0 [0%]
Cyto	59 [100%]	1626 [100%]	1666 [100%]
FactorXa	0 [0%]	0 [0%]	0 [0%]
MMP12	0 [0%]	6515 [67.2%]	4426 [71.2%]
Neurokinin	-	0 [0%]	0 [0%]
P2x7	0 [0%]	0 [0%]	0 [0%]
P2x7 Subset	0 [0%]	0 [0%]	0 [0%]
P38α	0 [0%]	0 [0%]	0 [0%]

8.3.2.1.4 Medchem Filters

The final filtering method that was explored was using the medchem filters. Both the BMS filters and drug-like properties were applied simultaneously. The number of filtered molecules can be found in Table 8-11. The medchem filters have a varied performance across the datasets and the different variations of substructural fragments available to build new molecules. For FactorXa and Neurokinin, all of the new molecules are filtered out. These filters may therefore be only applicable for some datasets and therefore would need to be changed accordingly. The input molecule for the FactorXa molecule is an important example as the molecule being altered also would not pass these filters as the MW is above 500.

Table 8-11: Number of molecules from the single node change that were filtered out for each dataset using the BMS and medicinal chemistry filters

Dataset	**Number of Molecules Filtered Out…**		
	Dataset	**ChEMBL**	**Zinc**
Bajorath	6 [15.8%]	368 [27.3%]	298 [24.9%]
CDK2	15 [14.4%]	214 [49.9%]	189 [50.4%]
Chk1	0 [0%]	0 -0%]	0 [0%]
Cyto	8 [13.6%]	576 [35.4%]	525 [31.5%]
FactorXa	1 [100%]	8 [100%]	13 [100%]
MMP12	6 [40%]	9404 [96.9%]	5985 [96.3%]
Neurokinin	-	8 [100%]	9 [100%]
P2x7	0 [0%]	77 [37.7%]	101 [44.1%]
P2x7 Subset	0 [0%]	1165 [60.1%]	529 [48.9%]
P38α	0 [0%]	5 [9.4%]	2 [4.4%]

The number of molecules that were filtered according to each filter was also investigated. Table 8-12 demonstrates the split in the number of molecules that are filtered out by the BMS filters and the property filters and the number in brackets, [], indicates how many BMS filters were used. Figure 8-17 shows the BMS filters used with their frequencies when applying them to the molecules generated from the datasets' substructural fragments. The breakdown of how many filters were used and the corresponding number of molecules removed by each can be found within the Appendix. There were only six BMS filters used within the molecules generated from the datasets' substructural fragments. 66 and 63 BMS filters were used for the molecules generated from the ChEMBL and Zinc substructural fragments, respectively.

For the molecules generated using the substructures from within the LO series, the BMS filter that filters the most molecules is the *halogen_heteroatom* with 15 examples across all ten datasets. This alert identifies a halogen is connected to anything other than a carbon or hydrogen atom. For both the ChEMBL and Zinc substructures, the BMS filter that filters the most molecules is *gte_10_carbon_sb_chain* with 2249 and 1065 examples across all ten datasets, respectively. This is a ten carbon chain link where each carbon is not connected to a ring atom.

Table 8-12: Table showing the split of molecule that are filtered by either the BMS filters or the property filter for the single node alteration. The number in the square brackets equals the number of BMS filters

Dataset	Filter	Number of Molecules Filtered Out…		
		Dataset	ChEMBL	Zinc
Bajorath	BMS	2 [1]	212 [14]	138 [11]
	Properties	4	156	160
CDK2	BMS	15 [1]	214 [6]	183 [3]
	Properties	0	0	6
Chk1	BMS	0	0	0
	Properties	0	0	0
Cyto	BMS	8 [4]	486 [39]	394 [34]
	Properties	0	90	131
FactorXa	BMS	0	2 [2]	3 [3]
	Properties	1	6	10
MMP12	BMS	0	2682 [17]	1675
	Properties	3	6700	4288
Neurokinin	BMS	-	8 [2]	9 [1]
	Properties	-	0	0
P2x7	BMS	0	77 [8]	101 [8]
	Properties	0	0	0
P2x7 Subset	BMS	0	569 [7]	213 [7]
	Properties	0	570	302
P38α	BMS	0	5 [2]	2 [1]

	Properties	0	0	0

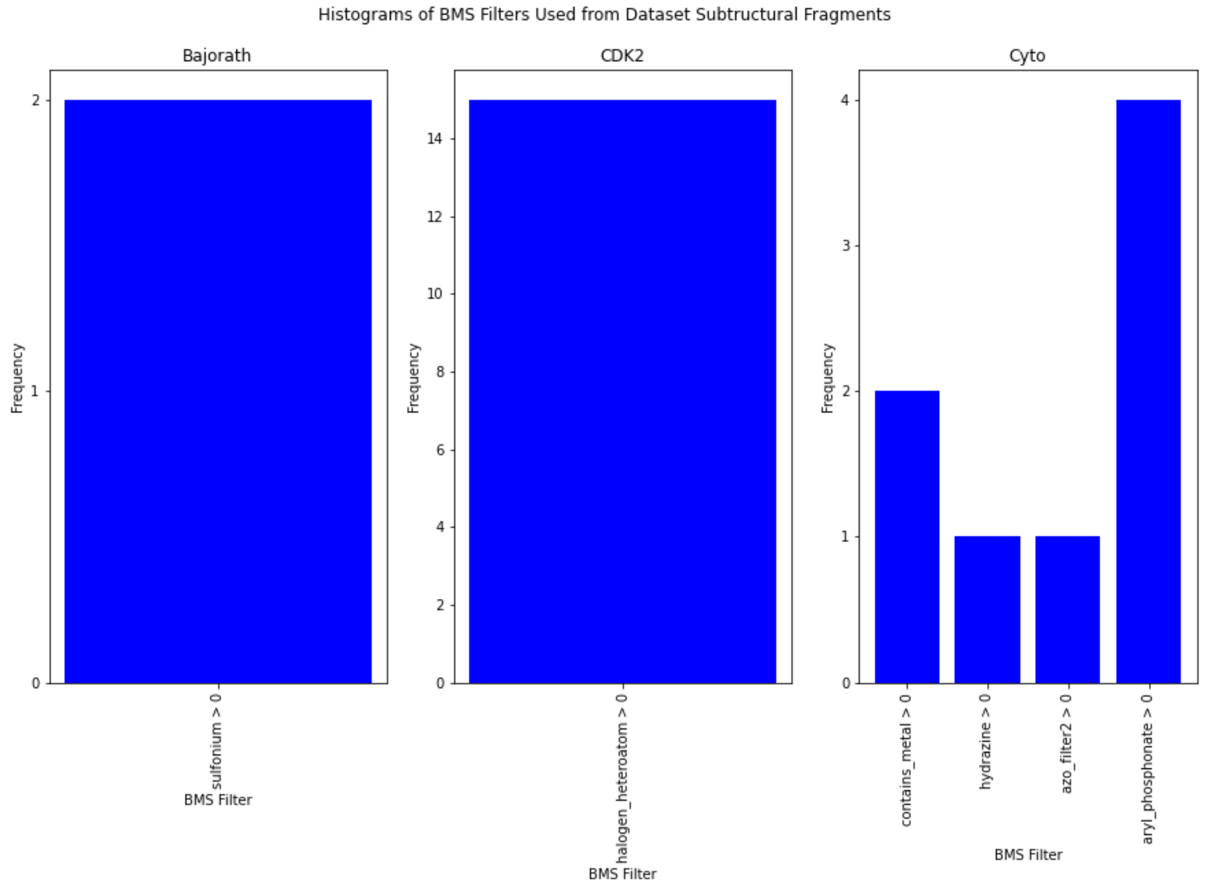


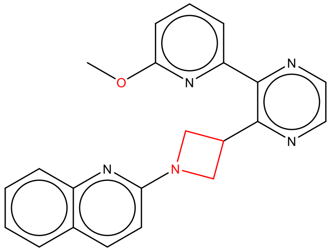
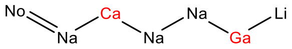
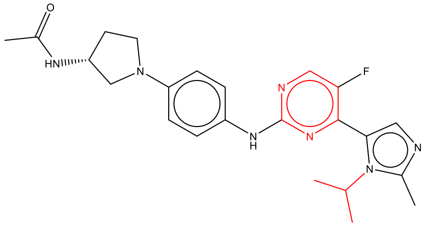
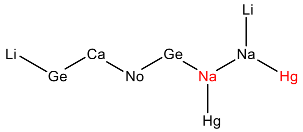
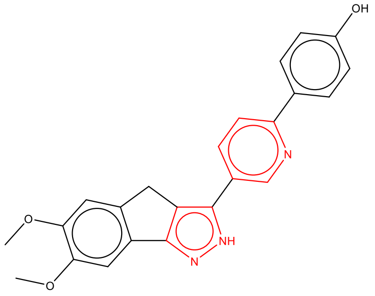
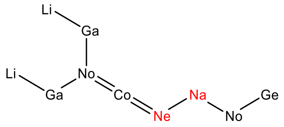
Figure 8-17: BMS filters applied to the single node alteration for each dataset when using the substructural fragments from the dataset

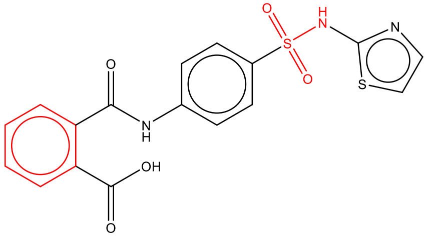
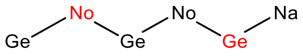
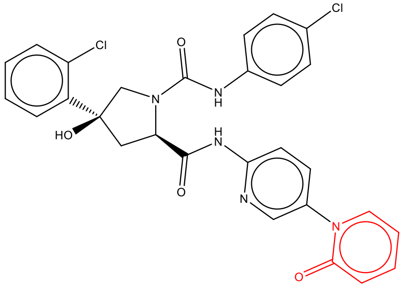
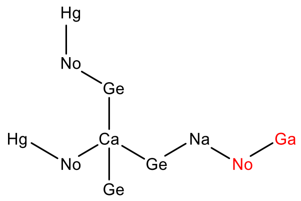
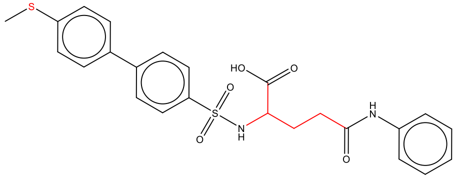
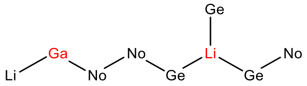
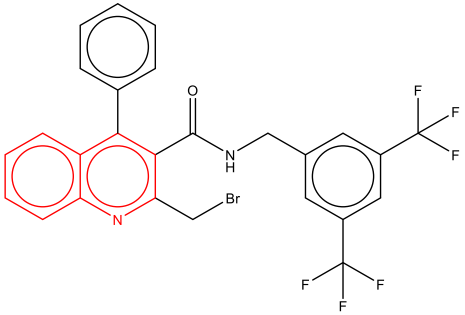
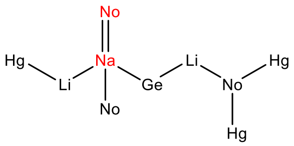
For some of these datasets, the drug-like properties were too strict as they passed the BMS filters, however, failed on the drug-like property. Even though the initial molecule would have also failed. Therefore, these properties could be altered according to the input molecule or what the chemist is interested in.

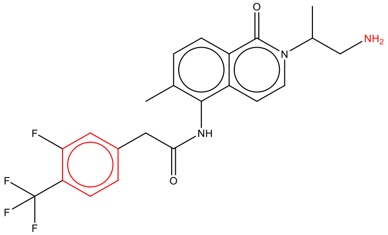
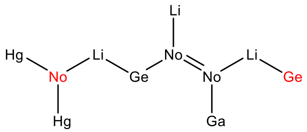
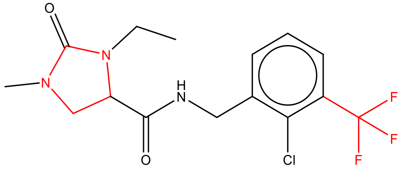
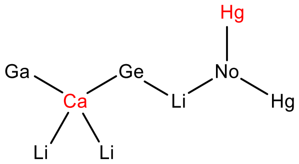
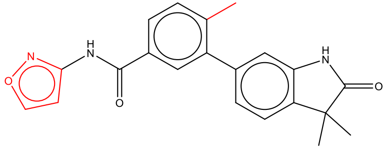
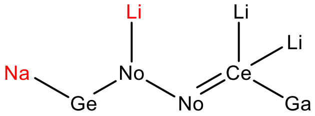
8.3.3 Substructure Replacements based on Multiple Nodes

The same molecules from the single node section then underwent a multiple node transformation. This was the same node from the single node transformation along with another randomly selected node. The number of replacement fragments and molecules generated using each dictionary for all datasets is shown in Table 8-13. In all cases, 100% of the generated molecules were valid.

Table 8-13: Results of the most active molecule from each dataset undergoing multiple node transformation

Dataset	Molecule	RG	Number of Molecules Generated Using…					
			Dataset		ChEMBL		Zinc	
			Replacement Fragments	Number of Molecules Generated	Replacement Fragments	Number of Molecules Generated	Replacement Fragments	Number of Molecules Generated
Bajorath	 CHEMBL3691541 (pIC50 = 11.92)		Ga 6 Ca 29	233	Ga 236 Ca 1054	397193	Ga 278 Ca 1004	351013
CDK2	 50422965 (pIC50 = 9.52)		Na 25 Hg 21	4409	Na 180 Hg 2087	1666679	Na 185 Hg 1178	813663
Chk1	 Chk1N144 (pIC50 = 9.52)		Ne 1 Na 5	35	Ne 55 Na 126	2399	Ne 67 Na 114	3189

Cyto	 4806 (pIC50 = 8.89)		Ge 42 No 14	899	Ge 1055 No 68	87857	Ge 1160 No 58	76681
FactorXa	 50266775 (pIC50 = 10.70)		No 16 Ga 6	79	No 68 Ga 77	3797	No 58 Ga 70	2753
MMP12	 A21B07 (pIC50 = 8.00)		Ga 5 Li 6	93	Ga 236 Li 2166	2852091	Ga 278 Li 1425	1822164
Neurokinin	 CHEMBL281797 (pIC50 = 11.00)		Na 1 No 6	17	Na 51 No 68	4772	Na 53 No 58	3599

P2x7	 CHEMBL2324343 (pIC50 = 11.00)		Ge 6 No 3	58	Ge 646 No 120	117258	Ge 606 No 126	115688
P2x7 Subset	 CHEMBL2218174 (pIC50 = 9.51)		Ca 9 Hg 13	389	Ca 3902 Hg 2087	11174507	Ca 3826 Hg 1178	6573149
P38α	 255896 (pIC50 = 10.40)		Na 18 Li 6	107	Na 54 Li 375	18953	Na 46 Li 210	8555

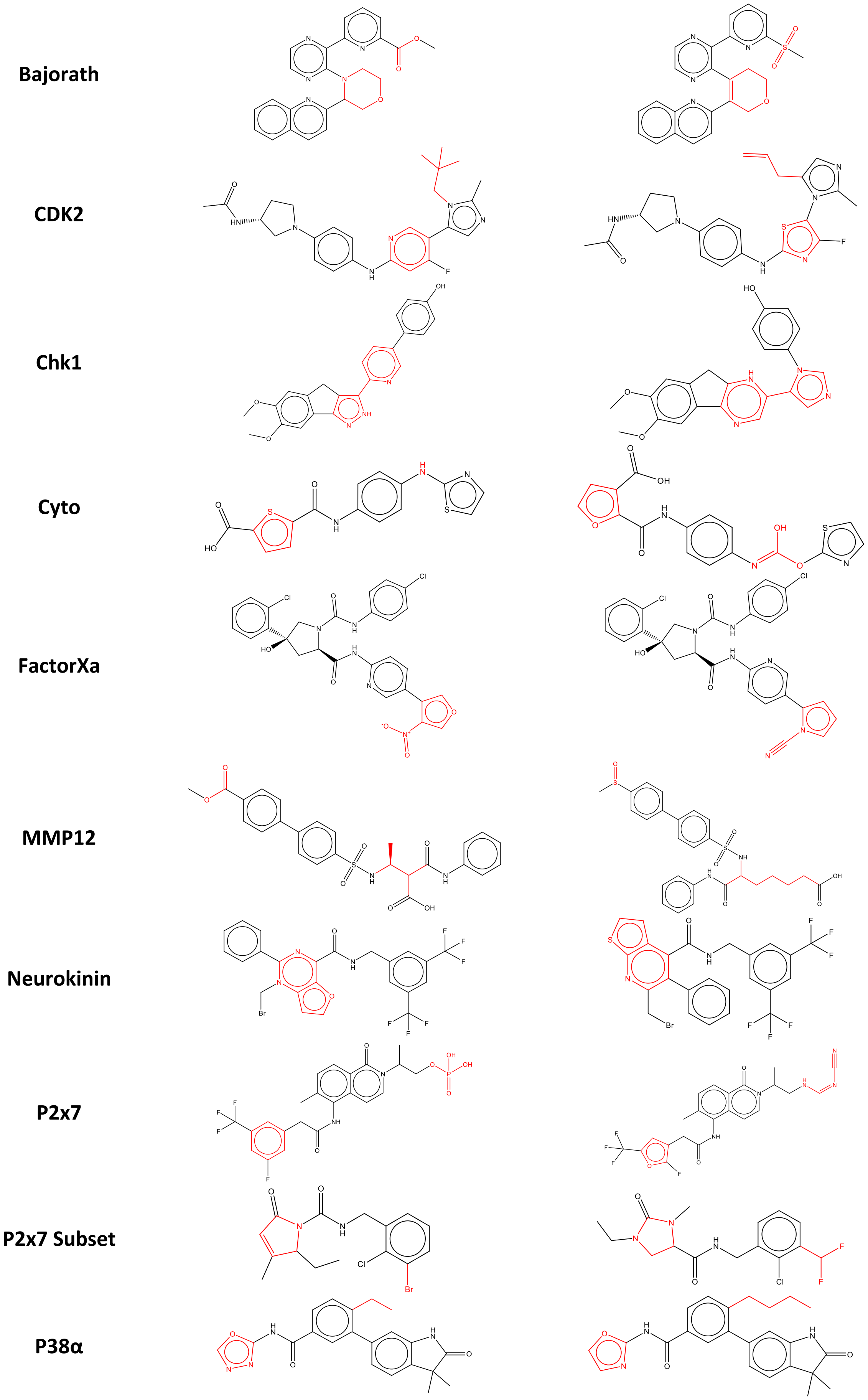


Figure 8-18: Two examples of molecules generated from the multiple node generation

After examining the molecules generated from the multiple node replacements, there are several key issues to note. The Chk1 examples within Figure 8-18 show that when multiple nodes are being replaced, the original substructure for one node is retained provided that the substructure for the other node is replaced. This is because these combinations of substructures have not been seen before. The P2x7 Subset demonstrates that a substructural fragment can be the same, but connected differently from the input molecule, as shown in the previous section. An example of how adjoining bond types can be altered from the initial bonds can be found in Figure 8-18 for FactorXa, where the double bond between the two nodes in the input molecule is not retained and a single bond joins the replacement substructures.

Table 8-14: Comparison of the molecules between each of the methods to establish the overlap

Dataset	Molecules That Are The Same in…		
	Dataset & …		ChEMBL & …
	ChEMBL	Zinc	Zinc
Bajorath	233	191	89464
CDK2	4409	3989	265481
Chk1	35	35	1827
Cyto	854	794	27650
FactorXa	79	79	1356
MMP12	93	93	263027
Neurokinin	17	17	2180
P2x7	58	58	41622
P2x7 Subset	389	233	1143071
P38α	107	107	4814

Table 8-14 shows the number of molecules that are the same between the three different node-substructure dictionaries. As for the single node replacements, all of the molecules generated using the LO dataset substructural fragments are subsets of the molecules generated using the ChEMBL substructural fragments, except for the Cyto dataset. However, this is not the case when comparing the molecules generated from the LO dataset fragments and the Zinc substructural fragments where there are three more datasets that are not subsets, Bajorath, CDK2 and P2x7 Subset. Finally, the overlap between the molecules generated from the two larger substructural fragment dictionaries, ChEMBL and Zinc, varies depending on the dataset. For some datasets the overlap is small, two of note are the MMP12 and P2x7 Subset datasets as these are around 15%. Whereas, for the Neurokinin dataset, the overlap is around 60%. The degree of overlap is dependent on the type of nodes being replaced and the number of connection points with the larger overlaps tending to be for the node types that have fewer substructural fragments extracted, shown in Table 8-4. Therefore, if a user wanted to explore a vast area of chemical space a combination of ChEMBL and Zinc substructural fragments would be more ideal.

8.3.3.1 Filtering Molecules

All the molecules from each experiment were then filtered using the four filtering processes.

8.3.3.1.1 Change in HAC

The generated molecules were filtered using the same HAC cut offs as the previous section, that is, -3 to 3. Table 8-15 demonstrates the number of molecules that were filtered out for each experiment. There was little to no effect on the molecules generated from the LO dataset substructural fragments. However, this is the opposite for the molecules generated from the ChEMBL and Zinc substructural fragments. The large filtration for these two methods is good, as in most instances, the chemists still have many molecules to examine to make their own opinions but are not overwhelmed with as many molecules as they would have been without the filter.

Table 8-15: Number of molecules from the multiple node change that were filtered out for each dataset using the change in HAC filter

Dataset	Number of Molecules Filtered Out…		
	Dataset	ChEMBL	Zinc
Bajorath	70 [30.0%]	321059 [80.8%]	314498 [89.6%]
CDK2	0 [0%]	1320460 [79.2%]	569324 [70.0%]
Chk1	0 [0%]	24 [1.0%]	0 [0%]
Cyto	35 [3.9%]	18663 [21.2%]	16811 [21.9%]
FactorXa	0 -0%]	563 [14.8%]	393 [14.3%]
MMP12	6 [6.5%]	2756063 [96.6%]	1786033 [98.0%]
Neurokinin	0 [0%]	9 [0.2%]	45 [1.3%]
P2x7	10 [17.2%]	70502 [60.1%]	74918 [64.8%]
P2x7 Subset	0 [0%]	9421770 [84.3%]	4996011 [76.0%]
P38α	25 [23.4%]	17646 [93.1%]	7772 [90.8%]

8.3.3.1.2 QSAR Filter

Similar to the single node investigation, the QSAR filter only removed molecules for a few of the datasets and for seven datasets no molecules were filtered regardless of which substructural fragments were used, Table 8-16. For the Cyto dataset, all of the molecules were filtered out when an activity cut off of 6.5 is used. Therefore, the QSAR filter for each dataset should be established for each or a top percentage or cut off should be applied.

Table 8-16: Number of molecules from the multiple node change that were filtered out for each dataset using the QSAR model

Dataset	Number of Molecules Filtered Out…		
	Dataset	ChEMBL	Zinc
Bajorath	0 [0%]	0 [0%]	0 [0%]

CDK2	3522 [79.9%]	1517698 [91.1%]	725682 [89.2%]
Chk1	0 [0%]	0 [0%]	0 [0%]
Cyto	899 [100%]	87857 [100%]	76681 [100%]
FactorXa	0 [0%]	0 [0%]	0 [0%]
MMP12	67 [72.0%]	2847883 [99.9%]	1819971 [99.9%]
Neurokinin	0 [0%]	0 [0%]	0 [0%]
P2x7	0 [0%]	0 [0%]	0 [0%]
P2x7 Subset	0 [0%]	0 [0%]	0 [0%]
P38α	0 [0%]	0 [0%]	0 [0%]

8.3.3.1.3 MedChem Filters

The BMS filter and property filters were applied to all the molecules generated in Table 8-13. The number of filtered molecules for each dataset and node-substructure dictionary can be found in Table 8-17. For the LO dataset substructural fragments at least 30 percent of the molecules were filtered out, apart from for the two smallest datasets Chk1 and P2x7 Subset. For the ChEMBL and Zinc node-substructure dictionaries, a larger percentage of molecules were filtered out and in some instances all the molecules were filtered. This is good as the chemist is presented with a more manageable list of suggested molecules to synthesis next. However, like the single node experiment, the property filter may need to be relaxed in some scenarios as a large proportion are filtered out and for some of the datasets the majority of the compounds in the LO series would also not pass these filters.

Table 8-17: Number of molecules from the multiple node change that were filtered out for each dataset using the BMS and medicinal chemistry filters

Dataset	**Number of Molecules Filtered Out…**		
	Dataset	**ChEMBL**	**Zinc**
Bajorath	76 [32.6%]	274926 [69.2%]	237055 [67.5%]
CDK2	1230 [27.9%]	1533742 [92.0%]	727022 [89.4%]
Chk1	0 [0%]	464 [19.3%]	596 [18.7%]
Cyto	328 [36.5%]	59071 [67.2%]	49818 [65.0%]
FactorXa	79 [100%]	3797 [100%]	2753 [100%]
MMP12	42 [45.2%]	2846480 [99.8%]	1819078 [99.8%]
Neurokinin	17 [100%]	4772 [100%]	3599 [100%]
P2x7	20 [34.5%]	99693 [85.0%]	101678 [87.9%]
P2x7 Subset	2 [2.8%]	9152049 [81.9%]	4896192 [74.5%]
P38α	33 [30.8%]	14385 [75.9%]	5850 [68.4%]

A breakdown of the split between the BMS filters or the property filters is shown for each experiment in Table 8-18. The number in brackets, [], indicates how many BMS filters were used. There were 13, 90 and 83 unique BMS filters used for the dataset, for the ChEMBL and Zinc substructural fragments respectively. The number of molecules that are filtered out for each filter is shown within the

Appendix. For the LO dataset substructural fragments, the most effective filter is the *halogen_heteratom* filter, which is where a halogen atom is connected to a heteroatom contained within a ring. For the ChEMBL and Zinc dictionaries the largest filter is once again the *gte_10_carbon_sb_chain*.

Table 8-18: Table showing the split of molecule that are filtered by either the BMS filters or the property filter for the multiple node alteration. The number in the square brackets equals the number of BMS filters

Dataset	**Filter**	**Number of Molecules Filtered Out…**		
		Dataset	**ChEMBL**	**Zinc**
Bajorath	BMS	49 [2]	229782 [37]	178370 [35]
	Properties	27	45144	58685
CDK2	BMS	990 [1]	1153591 [13]	528535 [11]
	Properties	240	377129	196394
Chk1	BMS	0	464 [5]	580 [6]
	Properties	0	0	16
Cyto	BMS	328 [6]	54769 [43]	46130 [39]
	Properties	0	4302	3688
FactorXa	BMS	38 [4]	2044 [24]	1433 [26]
	Properties	41	1753	1320
MMP12	BMS	0	1792294 [39]	1063831 [41]
	Properties	30	1053392	754780
Neurokinin	BMS	17 [1]	4772 [8]	3599 [3]
	Properties	0	0	0
P2x7	BMS	0	68037 [43]	65434 [40]
	Properties	20	31656	36244
P2x7 Subset	BMS	0	3739469 [32]	1550330 [23]
	Properties	11	5412580	3331523
P38α	BMS	0	10820 [11]	3584 [9]
	Properties	33	3565	2266

8.3.4 Full Enumeration

For each dataset, the molecule with the largest pIC50 value was examined. The first step was to find all possible substructural fragments for each node type with the correct number of neighbours (connection points) to be applicable for this enumeration. An example is displayed within Figure 8-19, based on the node-substructure dictionary from the MMP12 LO series. An indication of the number of molecules that would be generated can be calculated by considering all combinations. The real number is likely to be considerably larger due to the different ways of combining substructures with more than one connection points.

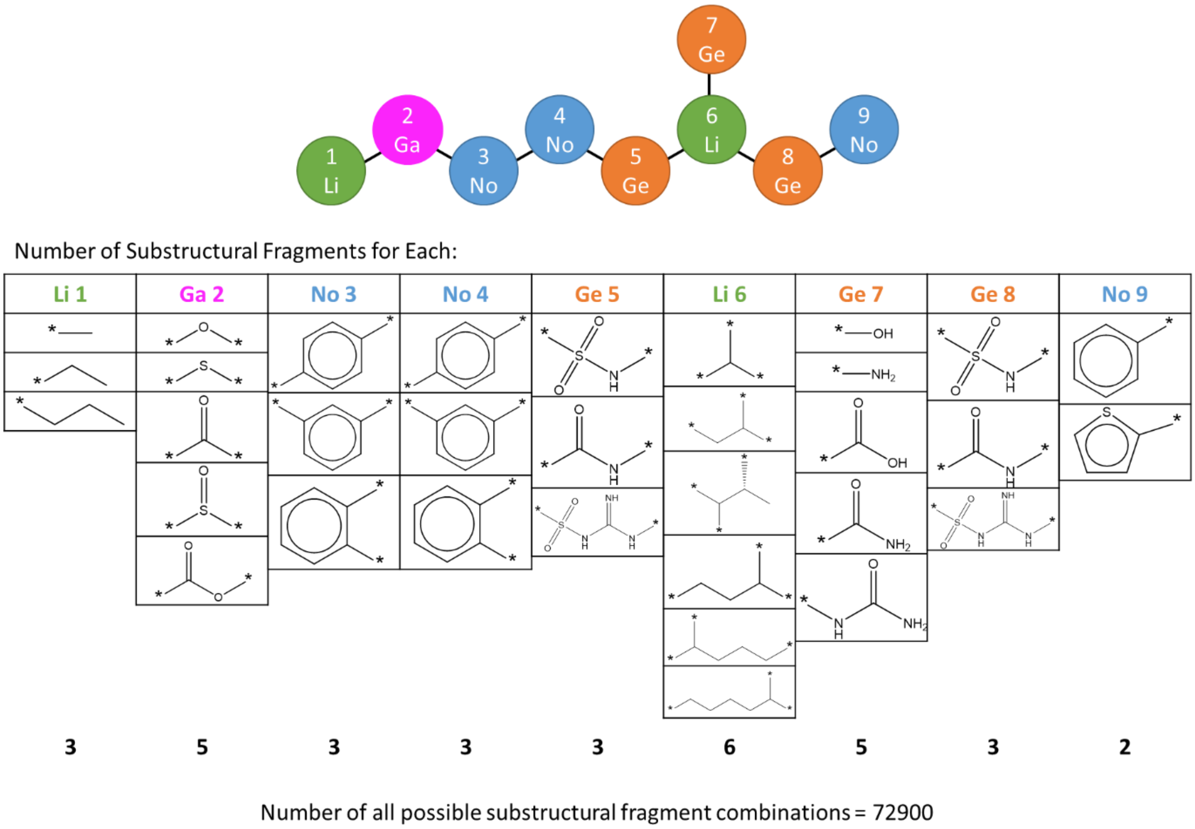
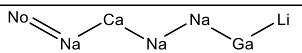
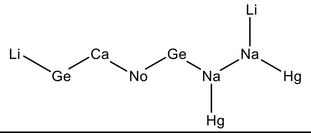
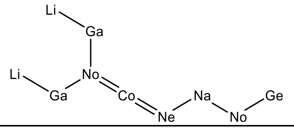
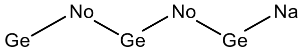
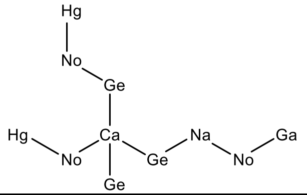
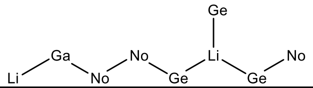
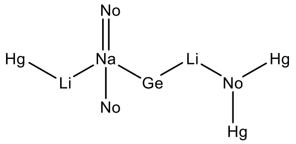
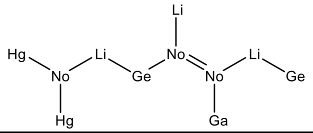
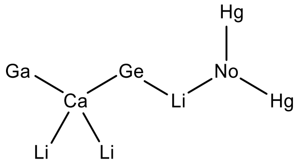
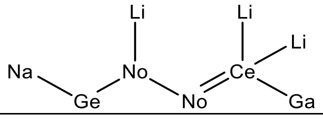

7 Ge
1 Li
2 Ga
3 No
4 No
5 Ge
6 Li
8 Ge
9 No
Number of Substructural Fragments for Each:
Li 1
Ga 2
No 3
No 4
Ge 5
Li 6
Ge 7
Ge 8
No 9
3
5
3
3
3
6
5
3
2
Number of all possible substructural fragment combinations = 72900

Figure 8-19: Extraction of all substructural fragments for RG and fragments from MMP12 dataset

Given the extremely large number of substructural fragments, full enumerations were not carried out. Instead the number of combinations of fragments for each RG were calculated. The number of combinations of each substructural fragment for each node for the overall RG can be seen in Table 8-19. It should be noted that these values do not equate to the number of molecules that would be generated as most fragments can be connected in multiple different ways via their connection points, as mentioned above. Additionally, some of these substructural fragments may not be compatible as the connecting bonds may not be the same and if a fused ring is present the overlapping atoms may not be the same or both rings may not have the same number of connecting atoms. Table 8-19 just gives an indication of the vast size of space that would be explored. Unfortunately, these would take a large amount of computational power to compute.

Table 8-19: Number of combinations of applicable substructural fragments for a full enumeration

Dataset	RG	Number of Combinations of Fragments…		
		Dataset	ChEMBL	Zinc
Bajorath		206,007,648	1.813E+16	8.173E+15
CDK2		6.293E+11	1.583E+27	1.641E+26

Chk1		75,600	7.221E+23	3.022E+23
Cyto		186,701,760	1.795E+14	1.262E+14
FactorXa		9.042E+13	3.728E+28	6.741E+27
MMP12		72,900	1.275E+22	2.510E+21
Neurokinin		1,395,523,584	9.325E+26	3.854E+25
P2x7		13,412,044,800	3.193E+30	2.262E+29
P2x7 Subset		821,340	7.963E+25	5.232E+24
P38α		783,820,800	5.903E+21	8.721E+20

The full enumeration was completed for the top RG within the MMP12 dataset using the substructural fragment dictionary generated from the dataset itself. Even though there were only 72,900 different fragment combinations, this resulted in 933,120 molecules. Therefore, this demonstrates how the number of molecules begins to explode.

A full enumeration using all possible substructural fragments, even from just the dataset, would require a large amount of computation and generate a vast amount of molecules. It would be more beneficial if the user was to craft their own preferences for each node. Or the user needs to set certain requirements to filter the fragment set down even further.

8.4 Verification of Molecular Generation Method

Two verification experiments have been created that aim to understand how effective the molecular generation algorithm is at identifying new molecules that would potentially be useful to a lead optimisation programme. The first analysis examines whether a hold out set can be generated from the main dataset. The second examines whether the hold out set is generated using each of the different substructural fragment sets.

8.4.1 Generating molecules in a hold out set

All molecules had their respective RGs calculated using the parameter “linker and complex”. For each dataset, each RG within the hold out 10% was examined to see whether it was present within the 90% RGs. Table 8-20 shows the number of molecules for which the RG representations are not represented within the 90%. These molecules are removed since the molecular generation method is not applicable to the generation of these due to it being driven by the RG representation of the input molecules. This is a limitation of the method.

Table 8-20: Table showing the split of data within the 90% and 10% set for each dataset and the number of molecules that could not be created

Dataset	90%		10%		Number of Molecules in holdout for which the RG is Not Represented in 90%
	Number of Molecules	Number of Unique RG	Number of Molecules	Number of Unique RG	
Bajorath	1876	981	208	191	66
CDK2	1230	842	137	129	71
Chk1	95	86	11	11	11
Cyto	5675	3999	635	590	328
FactorXa	1760	981	196	181	67
MMP12	1534	563	170	135	39
Neurokinin	1321	800	147	133	62
P2x7	1607	749	179	148	46
P2x7 Subset	622	210	69	60	13
P38α	3280	2038	364	330	147

Hold out sets for each dataset were derived using stratified sampling. For each dataset, first, singletons were removed, that is, molecules with a unique RG representation, and then a stratified split was taken of the remaining molecules to generate a training and a hold out set. The size of the hold out set has to be equal to or greater than the number of unique RGs. The singletons were then recombined with the training set to generate the substructural fragments for each node type. The split in data can be seen in Table 8-21.

Table 8-21: Training and test set split using the stratified sampling approach

Dataset	Singletons		Training Set		Hold Out Set
	Number of Molecules	Number of Unique RG	Number of Molecules	Number of Unique RG	Number of Molecules and Unique RG
Bajorath	702	702	964	418	418
CDK2	681	681	455	231	231
Chk1	90	90	9	7	7
Cyto	3237	3237	1988	1085	1085

FactorXa	653	653	910	393	393
MMP12	275	275	1102	327	327
Neurokinin	559	559	608	301	301
P2x7	482	482	992	312	312
P2x7 Subset	104	104	469	118	118
P38α	144	144	1463	740	740

For each molecule, the nearest molecule in the training set with the same RG was found within the hold out. The nearest molecule was perceived to be the molecule with the fewest number of RG node transformations. The number of substructural changes that would be required to generate the hold out molecule from the molecule in the training data was then calculated based on the node definitions. Most of the molecules in the hold out set can be generated with one or two node changes, shown in Table 8-22. The structure generation method was then applied to the molecule in the training data to see if the hold out molecule could be generated. This was repeated for each training set-hold out set molecule pair from each dataset using the three different node-substructure dictionaries: the one derived from the LO series; the ChEMBL dictionary; and the Zinc dictionary. The number of molecules from the hold out sets that could not be generated from each of the different node substructure dictionaries can be found in Table 8-23.

Table 8-22: Number of node changes to make for each molecule within each datasets test set

Dataset	Molecules in Test Set	Node Changes…		
		One	Two	More Than Two
Bajorath	418	350	42	25
CDK2	231	185	29	16
Chk1	7	5	2	0
Cyto	1085	700	254	117
FactorXa	393	326	39	22
MMP12	327	271	40	16
Neurokinin	301	237	48	13
P2x7	312	234	38	35
P2x7 Subset	118	81	26	10
P38α	740	610	68	38

Table 8-23: The number of molecules that cannot be generated in the hold out set from each of the different substructural dictionaries

Dataset	Number of Molecules in Hold Out Set	Number of Molecules Not Generated Using Substructural Fragments From …		
		Dataset	ChEMBL	Zinc
Bajorath	418	17	0	9
CDK2	231	28	1	12
Chk1	7	5	0	0
Cyto	1085	83	3	67
FactorXa	393	42	1	3

MMP12	327	0	0	0
Neurokinin	301	42	0	19
P2x7	312	10	0	5
P2x7 Subset	118	2	0	4
P38α	740	21	1	8

Two examples from the P2x7 Subset dataset molecules that could not be generated are shown in Figure 8-20. The hold out molecule and its nearest neighbour are shown. The first example requires just one substructure/node replacement (shown in red) and the second requires two replacements (shown in red and blue). The tables underneath show the substructural fragments that are present in the node-substructure dictionary generated from the training data. It can be seen that for the first molecule the required substructural fragment is not contained within the already seen substructural fragments extracted from the training set. For the second molecule, the substructure required as a replacement for the Li node is present, however, that required for the Ge node is not present within the Ge substructural fragments that exist within the training set. This means that it cannot be generated from the node-substructure dictionary from the training data. However, there are many instances where these are present within the ChEMBL and Zinc node-substructure dictionaries and why more molecules can then be generated. Also, all datasets that are extracted from ChEMBL, (Bajorath, Neurokinin, P2x7 and P2x7 Subset) when using the ChEMBL node-substructure dictionary are able to make the hold out molecules. These datasets should have good reproducibility as all fragments are presented within the ChEMBL database.

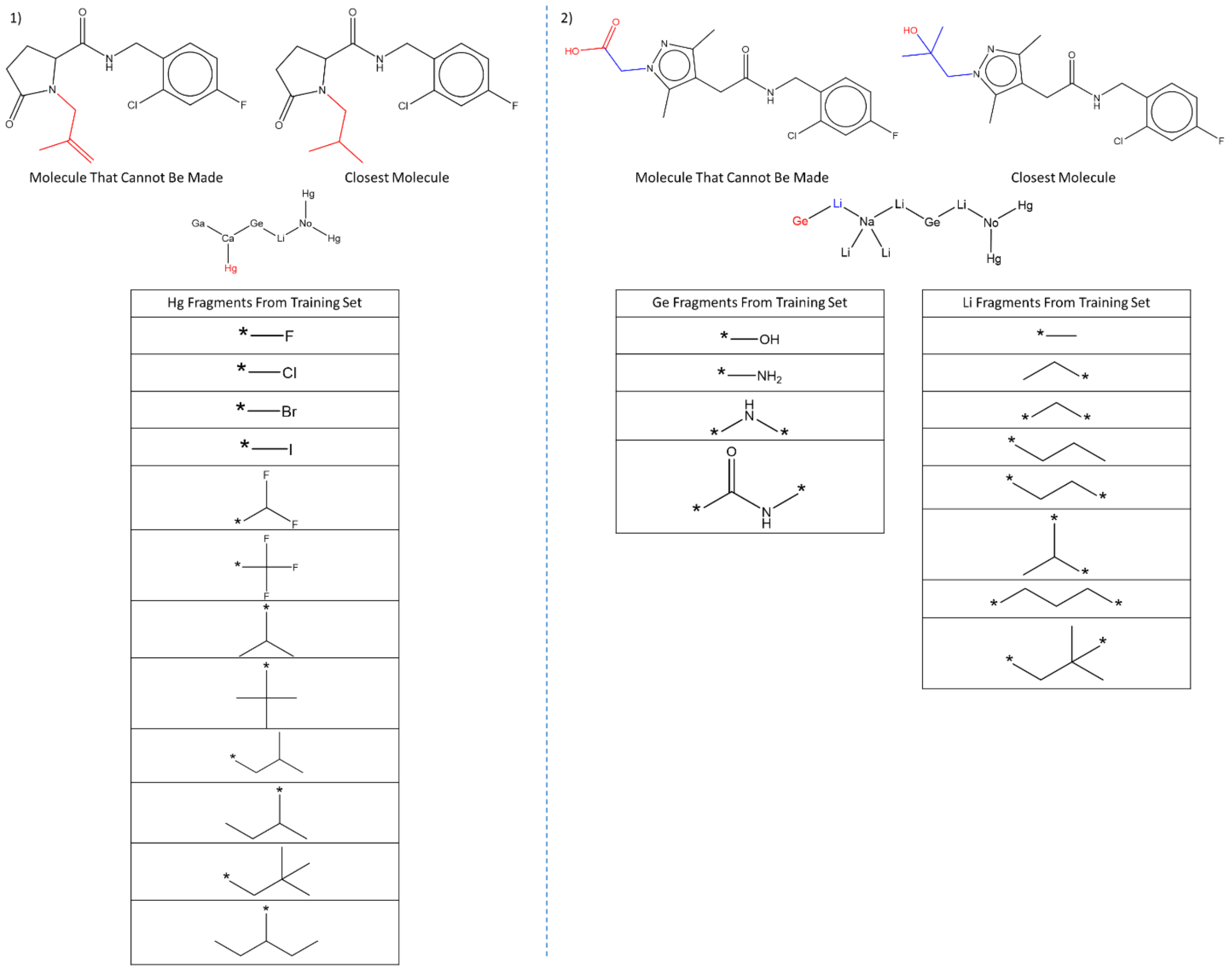

1)
Molecule That Cannot Be Made
Closest Molecule
Hg Fragments From Training Set
2)
Molecule That Cannot Be Made
Closest Molecule
Ge Fragments From Training Set
Li Fragments From Training Set

Figure 8-20: Example of how the two molecules from P2x7 Subset with the dataset substructural fragments cannot be made

8.5 Conclusion

A new molecular generation algorithm has been created based upon RGs and BRICS and BREED. A molecule can have a proportion of it altered according to the underlying RG node structure. This can either be a single or multiple node alteration, with the corresponding RG node fragments being used from either the LO series itself, or from the ChEMBL or Zinc node-substructural dictionaries. An advantage of this methodology is that the RG structure is always retained, unlike Pogány et. al, making it a good technique for scaffold hopping as the pharmacophoric features remain. The RG structure remains the same while introducing variations in the underlying chemical structure. Additionally, this methodology can also act as a way to essentially complete a Free-Wilson matrix for R-groups, however, it can also incorporate changes within the RG core not just the R-groups.

The single node and multi node alterations both can generate numerous molecules. Three different node-substructure dictionaries were used from which the new molecules are generated. There were fewer molecules generated from the LO series node-substructure dictionary as there were fewer substructural fragments. This would allow a chemist to remain within the domain of applicability as the substructures have already been seen. Also, these molecules generally tended to be a subset of the molecules extracted from the molecules generated from the ChEMBL node-substructure dictionary. A large proportion of molecules were different when using either the ChEMBL or Zinc node-substructure dictionaries, which is due to the initial substructural fragments being different. Therefore, the molecules generated are dependent on the node-substructure dictionary used.

Additionally, some ways were examined that allowed the molecules generated to be filtered to provide a more manageable list of suggested molecules. However, some work still needs to be done on this. It may also be worth the substructural fragments going through an additional filtering process. This would reduce the number of molecules generated and allow the molecules to be more appropriate.

Furthermore, a full enumeration was attempted, however, due to the number of nodes and fragments there were a vast number of ways in which these could be combined. To make a full enumeration more achievable then the number of substructural fragments for each node needs to be reduced. This could be achieved by either filtering the fragments or allowing a chemist to choose several fragments for each node that could be of interest. This would reduce the amount of combinations and allow a more computational viable option. Additionally, the chemist would be able to personalise the molecule to their own preferences whilst retaining the RG structure.

9 Conclusions and Future Work

9.1 Conclusions

This thesis has detailed the creation and evaluation of a new visualisation for LO series, two new scores to assess the contribution that new molecules would make to the series, and a new de novo design technique based on an existing LO series. The ability to present information to chemists to provide an insight into relationships within a lead optimisation dataset is important in drug discovery to allow them to make informed choices for the next iteration of the project. The visualisation aims to overcome limitations of existing methods in that Markush structures and SAR tables which are typically used do not allow scaffolds that have small alterations to be considered under the same representation.

Chapter 2 demonstrated how a RG core could be extracted from a clustered dataset. A combination of different molecular descriptors and clustering algorithms were investigated, to provide a pre-clustered dataset to allow the best RG core to be extracted. The RG core represents the relationships between the molecules within a cluster. It is important to have RG cores that reflect the relationships between molecules as this affects the rest of the results within the thesis. While it was shown that it was possible to cluster a dataset it was difficult to identify the most appropriate cut-off for the different methods, whether this is the number of clusters or the similarity value used.

Chapter 3 aimed to assess the quality of the RG cores. RG cores were extracted from both the clustered datasets and the datasets as a whole. In most instances, the most effective method was to extract the RG cores from the whole dataset. These RG cores were more concise and were therefore more representative of the data within the LO dataset. This was also confirmed by the scaffold scores produced, as on average the whole dataset RG cores had larger scores than those produced from the clustered data. Furthermore, the RG cores were compared to Markush structures and Murcko scaffolds where it was shown that the RG cores provided advantages over these existing methods, due to their sizes and the calculated scaffold scores.

Chapter 4 first describes the process of mapping the RGs of the molecules in a LO dataset back onto the RG cores. When molecules have multiple mappings to an RG core a prioritisation process has been created to allow just one RG core to be mapped that best aligns with the rest of the molecules in the LO series. The development of a new visualisation tool is then described in Chapter 5. First, the substructural fragments for each node within the RG core are identified. The nodes within a RG core are then be represented as pie charts, where each segment indicates a different substructural fragment. The visualisation provides an understanding of the relationships between molecules in a LO series, whilst providing information on how well explored the chemical space for each node is.

Chapter 6 introduced several different methodologies that have been analysed in the development of the exploration score. The exploration score should reflect the amount of information that would be added to a LO series by a new molecule based on its structure. The developed score is based on an existing score used within high-throughput screening, the collection model score (Harper, Pickett, & Green, 2004), that has been adapted for the exploration purpose. The mean of all the node scores was calculated to provide a score for the whole RG core. By using the mean this allows the comparison between molecules that map to different RG cores including when the RG cores have different number of nodes with the RG core. The exploration score was compared to a manual ranked list and it performed well.

Chapter 7 describes the development of the second score which is the exploitation score. The exploitation score is based upon the significance of each of the substructural fragments within the node. The significance of each fragment was calculated based on a methodology described by Polishchuk et. al (P. G. Polishchuk et al., 2013). Once again, the mean of the node scores is found for the overall molecule exploitation score. Unfortunately, these scores had large uncertainties, therefore, these are not the most ideal score. However, the exploitation score offers a chemist a base to work from other than just making a random choice.

Finally, Chapter 8 detailed a new algorithm for de novo design. A new molecule can be achieved by replacing either a single substructure or multiple substructures of an input molecules, where the substructures are defined according to RGs generated using a dataset of molecules. The resulting new molecules retain the same RG representation as the input molecule but are constructed from different substructures. The replacement substructures should have the same node type and number of neighbours and are extracted from a pre-defined node-substructure dictionary. These node-substructure dictionaries can be generated from the LO series itself or ChEMBL or Zinc. There are fewer molecules generated when using the node-substructure dictionary from the LO series as would be expected, however, these molecules are likely to be closely related to those in the series. The molecules from the ChEMBL and Zinc dictionaries explore a larger area of chemical space. A full enumeration was attempted, however, there was a large computational cost when using all potential substructural fragments. To make this method more effective, then the fragments for each node must be filtered to a select a subset of fragments.

9.2 Future Work

This thesis has introduced a new visualisation, an exploration score, an exploitation score and a new de novo design tool. There are several ways in which this work could be improved and developed further. The first is that the RG core could be allowed to be more flexible. It could be seen that some of the RG cores were closely related to each other. An optimisation of these cores could be an area of development to achieve more coherent results. An optimisation could be achieved through trying a disconnected core or alternatively having a core that contained a node that could be either of two nodes or just a particular binding style or aromaticity style. Therefore, some nodes could be incorporated together and if the user wanted to delve into greater depths then it could provide a way of showing these different node types. An example of this is when two RG cores differ by one node and this node is an aromatic ring hydrogen bond acceptor node and the other is an aromatic ring hydrogen bond acceptor and donor.

Additionally, it would be good if a user could import a user defined RG core and then all the corresponding data according to that RG core could be extracted and represented. This would allow the user to input known scaffolds that are important for the biological activity or that are of interest to their LO projects.

The scores provided information for the substructural fragments that were incorporated within the RG core, however, no information was provided about the R-groups. Therefore, the R-groups of the molecules could be aligned and introduced into the visualisation and incorporated somehow into both scoring techniques, even if just for each R-group the exploration and exploitation scores are produced for each node and not incorporated into the scores and just presented on the visualisation. The user would then have some indication as to the level of exploration of each node and significance of each substructural fragment.

Following the large uncertainties observed with the significance scores that are used within the exploitation score a more effective exploitation score should be produced. One way in which this might be achieved is to incorporate physicochemical properties after the FP and train the models from these. Improved models may also provide a smaller uncertainty, which may be achieved through more refinement in the hyperparameters.

Finally, within the molecular generation algorithm, as well as filtering the fragment list more which has already been stated, then it would be interesting to create an algorithm that can take a RG core as an input and then produce new molecules. This would mean that a new algorithm would need to go from a RG core to a full RG to then undergo the full enumeration. Also, another piece of future work for the molecular generation step could be to turn it into an active learning problem and the choice of molecule would be the molecule that had the most uncertainty.

Appendix

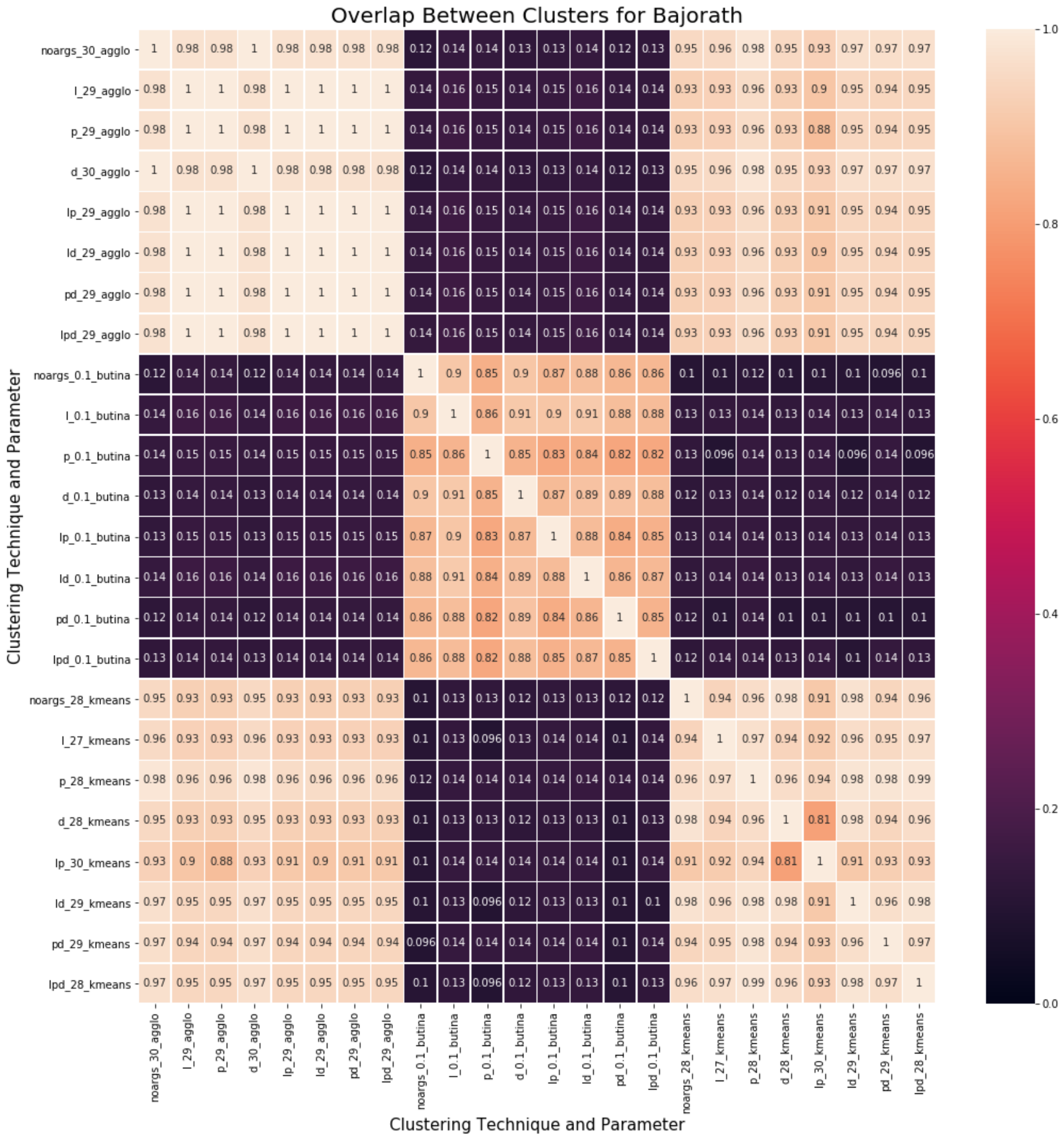
Reduced Graphs

Table 0-1: A comparison of the effect of the different RG parameters on the number of RG (average size of RG rounded to the nearest whole number)

Parameter	Bajorath	CDK2	Chk1	Cyto	FactorXa	Neurokinin	P2x7	P38α
Default	920 (10)	824 (8)	91 (8)	3762 (8)	883 (10)	1451 (9)	822 (9)	1902 (8)
Terminal	1089 (10)	872 (9)	97 (9)	4262 (9)	1010 (11)	1530 (10)	842 (10)	2026 (9)
Complex	1250 (10)	929 (9)	97 (9)	4463 (9)	1087 (11)	1645 (10)	1020 (10)	2246 (9)
Double Bond	929 (10)	825 (8)	91 (8)	3814 (8)	890 (10)	1461 (9)	823 (9)	1904 (8)
Metal	920 (10)	824 (8)	91 (8)	3765 (8)	883 (10)	1451 (9)	822 (9)	1902 (8)
Terminal and Complex	1227 (10)	923 (9)	97 (9)	4442 (9)	1089 (11)	1641 (10)	1018 (10)	2236 (9)
Terminal and Double Bond	1094 (10)	872 (9)	97 (9)	4274 (9)	1012 (11)	1535 (10)	842 (10)	2028 (9)
Terminal and Metal	1089 (10)	872 (9)	97 (9)	4264 (9)	1010 (11)	1530 (10)	842 (10)	2026 (9)
Complex and Double Bond	1255 (10)	929 (9)	97 (9)	4489 (9)	1088 (11)	1649 (10)	1020 (10)	2248 (9)
Complex and Metal	1250 (10)	929 (9)	97 (9)	4465 (9)	1087 (11)	1645 (10)	1020 (10)	2246 (9)
Double Bond and Metal	929 (10)	825 (8)	91 (8)	3816 (8)	890 (10)	1461 (9)	823 (9)	1904 (8)
Terminal, Complex and Double Bond	1232 (10)	923 (9)	97 (9)	4451 (9)	1090 (11)	1646 (10)	1018 (10)	2238 (9)
Terminal, Complex and Metal	1227 (10)	923 (9)	97 (9)	4444 (9)	1089 (11)	1641 (10)	1018 (10)	2236 (9)
Terminal, Double Bond and Metal	1094 (10)	872 (9)	97 (9)	4276 (9)	1012 (11)	1535 (10)	842 (10)	2028 (9)
Complex, Double Bond and Metal	1255 (10)	929 (9)	97 (9)	4491 (9)	1088 (11)	1649 (10)	1020 (10)	2248 (9)
Terminal, Complex, Double Bond and Metal	1232 (10)	923 (9)	97 (9)	4453 (9)	1090 (11)	1646 (10)	1018 (10)	2238 (9)

Clustering

Clustering overlap heatmaps



	noargs_30_agglo	l_29_agglo	p_29_agglo	d_30_agglo	lp_29_agglo	ld_29_agglo	pd_29_agglo	lpd_29_agglo	noargs_0.1_butina	l_0.1_butina	p_0.1_butina	d_0.1_butina	lp_0.1_butina	ld_0.1_butina	pd_0.1_butina	lpd_0.1_butina	noargs_28_kmeans	l_27_kmeans	p_28_kmeans	d_28_kmeans	lp_30_kmeans	ld_29_kmeans	pd_29_kmeans	lpd_28_kmeans
noargs_30_agglo	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.12	0.14	0.14	0.13	0.13	0.14	0.12	0.13	0.95	0.96	0.98	0.95	0.93	0.97	0.97	0.97
l_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.9	0.95	0.94	0.95
p_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.88	0.95	0.94	0.95
d_30_agglo	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.12	0.14	0.14	0.13	0.13	0.14	0.12	0.13	0.95	0.96	0.98	0.95	0.93	0.97	0.97	0.97
lp_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.91	0.95	0.94	0.95
ld_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.9	0.95	0.94	0.95
pd_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.91	0.95	0.94	0.95
lpd_29_agglo	0.98	1	1	0.98	1	1	1	1	0.14	0.16	0.15	0.14	0.15	0.16	0.14	0.14	0.93	0.93	0.96	0.93	0.91	0.95	0.94	0.95
noargs_0.1_butina	0.12	0.14	0.14	0.12	0.14	0.14	0.14	0.14	1	0.9	0.85	0.9	0.87	0.88	0.86	0.86	0.1	0.1	0.12	0.1	0.1	0.1	0.096	0.1
l_0.1_butina	0.14	0.16	0.16	0.14	0.16	0.16	0.16	0.16	0.9	1	0.86	0.91	0.9	0.91	0.88	0.88	0.13	0.13	0.14	0.13	0.14	0.13	0.14	0.13
p_0.1_butina	0.14	0.15	0.15	0.14	0.15	0.15	0.15	0.15	0.85	0.86	1	0.85	0.83	0.84	0.82	0.82	0.13	0.096	0.14	0.13	0.14	0.096	0.14	0.096
d_0.1_butina	0.13	0.14	0.14	0.13	0.14	0.14	0.14	0.14	0.9	0.91	0.85	1	0.87	0.89	0.89	0.88	0.12	0.13	0.14	0.12	0.14	0.12	0.14	0.12
lp_0.1_butina	0.13	0.15	0.15	0.13	0.15	0.15	0.15	0.15	0.87	0.9	0.83	0.87	1	0.88	0.84	0.85	0.13	0.14	0.14	0.13	0.14	0.13	0.14	0.13
ld_0.1_butina	0.14	0.16	0.16	0.14	0.16	0.16	0.16	0.16	0.88	0.91	0.84	0.89	0.88	1	0.86	0.87	0.13	0.14	0.14	0.13	0.14	0.13	0.14	0.13
pd_0.1_butina	0.12	0.14	0.14	0.12	0.14	0.14	0.14	0.14	0.86	0.88	0.82	0.89	0.84	0.86	1	0.85	0.12	0.1	0.14	0.1	0.1	0.1	0.1	0.1
lpd_0.1_butina	0.13	0.14	0.14	0.13	0.14	0.14	0.14	0.14	0.86	0.88	0.82	0.88	0.85	0.87	0.85	1	0.12	0.14	0.14	0.13	0.14	0.1	0.14	0.13
noargs_28_kmeans	0.95	0.93	0.93	0.95	0.93	0.93	0.93	0.93	0.1	0.13	0.13	0.12	0.13	0.13	0.12	0.12	1	0.94	0.96	0.98	0.91	0.98	0.94	0.96
l_27_kmeans	0.96	0.93	0.93	0.96	0.93	0.93	0.93	0.93	0.1	0.13	0.096	0.13	0.14	0.14	0.1	0.14	0.94	1	0.97	0.94	0.92	0.96	0.95	0.97
p_28_kmeans	0.98	0.96	0.96	0.98	0.96	0.96	0.96	0.96	0.12	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.96	0.97	1	0.96	0.94	0.98	0.98	0.99
d_28_kmeans	0.95	0.93	0.93	0.95	0.93	0.93	0.93	0.93	0.1	0.13	0.13	0.12	0.13	0.13	0.1	0.13	0.98	0.94	0.96	1	0.81	0.98	0.94	0.96
lp_30_kmeans	0.93	0.9	0.88	0.93	0.91	0.9	0.91	0.91	0.1	0.14	0.14	0.14	0.14	0.14	0.1	0.14	0.91	0.92	0.94	0.81	1	0.91	0.93	0.93
ld_29_kmeans	0.97	0.95	0.95	0.97	0.95	0.95	0.95	0.95	0.1	0.13	0.096	0.12	0.13	0.13	0.1	0.1	0.98	0.96	0.98	0.98	0.91	1	0.96	0.98
pd_29_kmeans	0.97	0.94	0.94	0.97	0.94	0.94	0.94	0.94	0.096	0.14	0.14	0.14	0.14	0.14	0.1	0.14	0.94	0.95	0.98	0.94	0.93	0.96	1	0.97
lpd_28_kmeans	0.97	0.95	0.95	0.97	0.95	0.95	0.95	0.95	0.1	0.13	0.096	0.12	0.13	0.13	0.1	0.13	0.96	0.97	0.99	0.96	0.93	0.98	0.97	1

Figure 0-1: Bajorath heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap

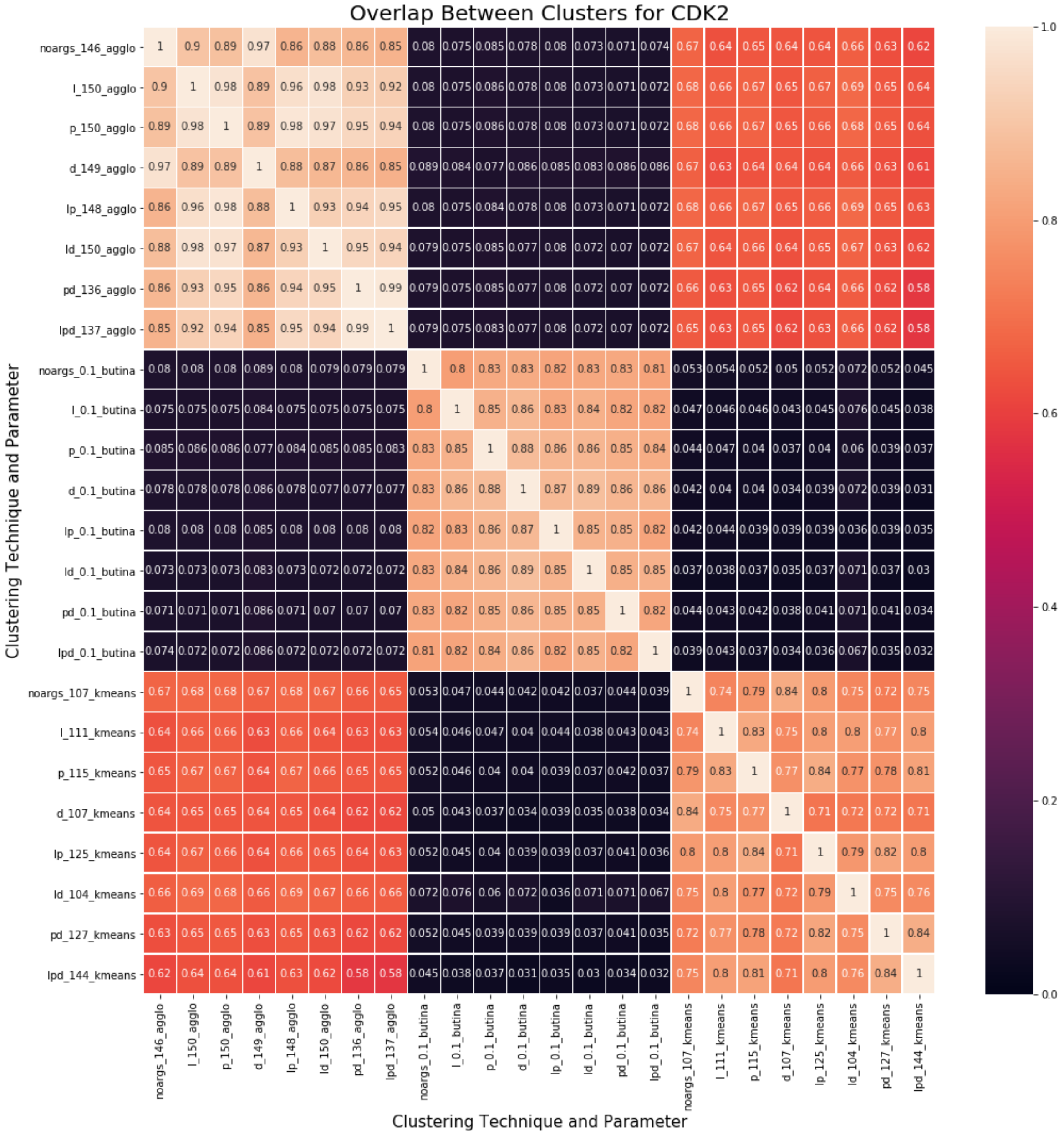
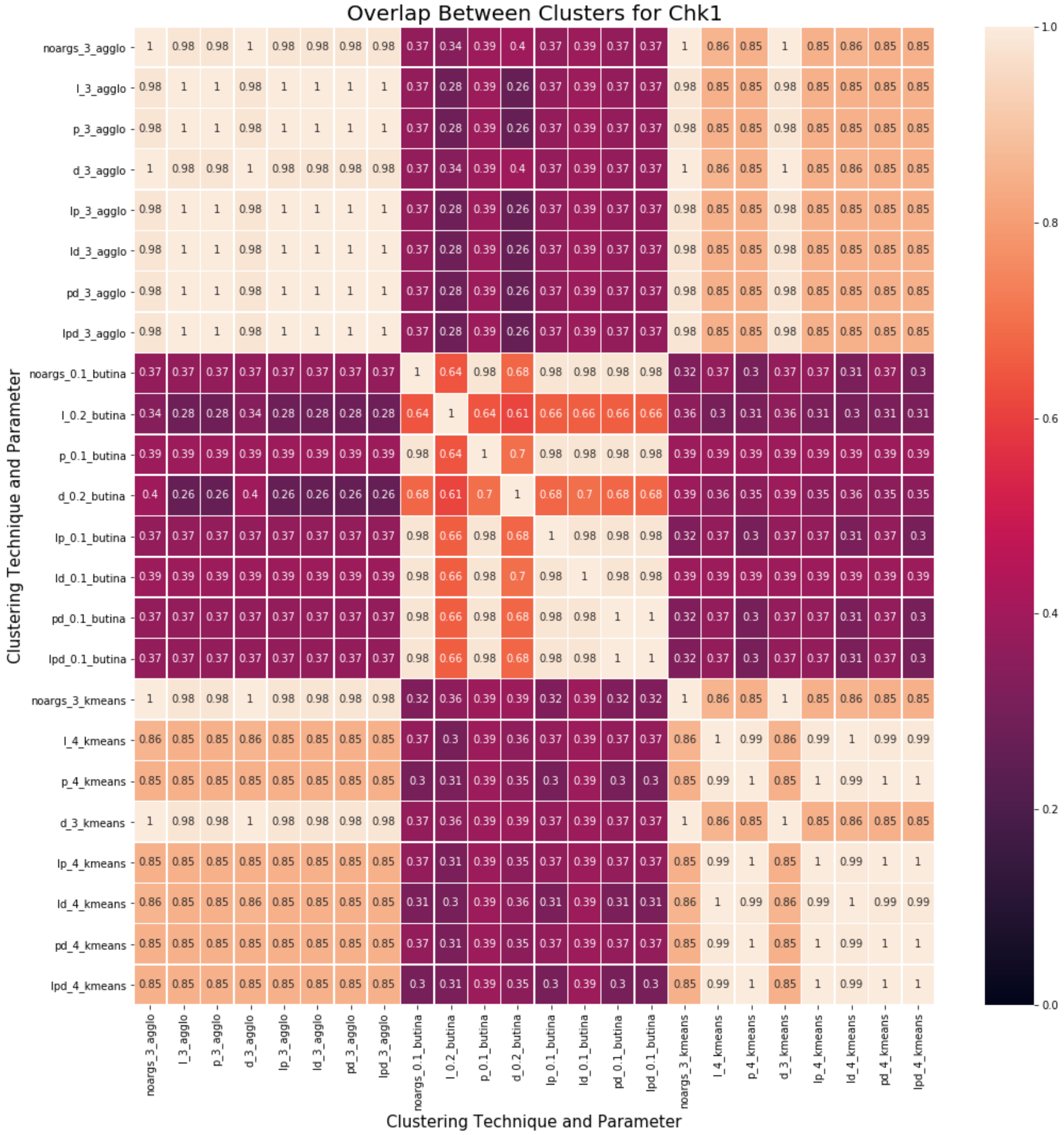
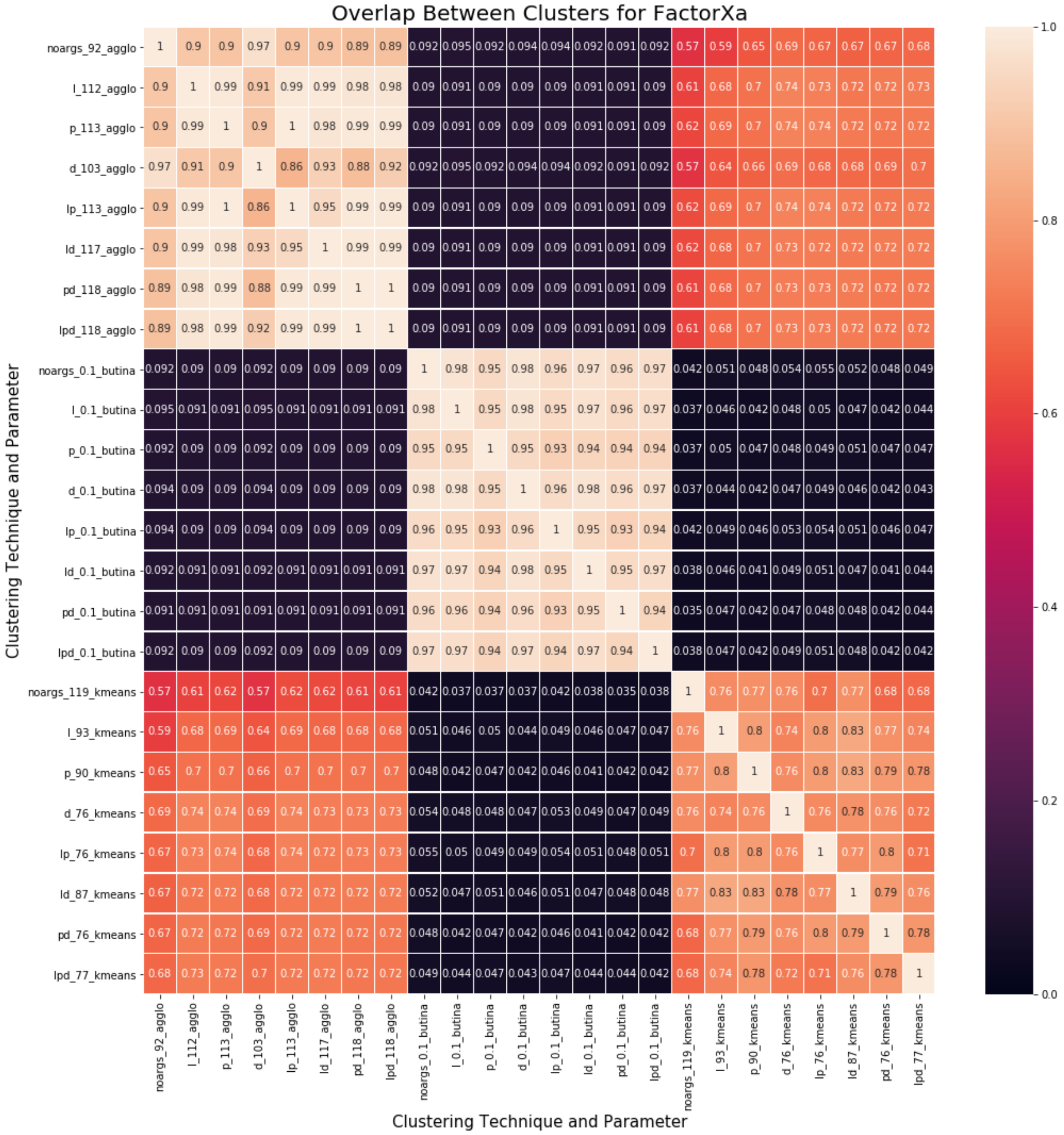


Figure 0-2: CDK2 heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap



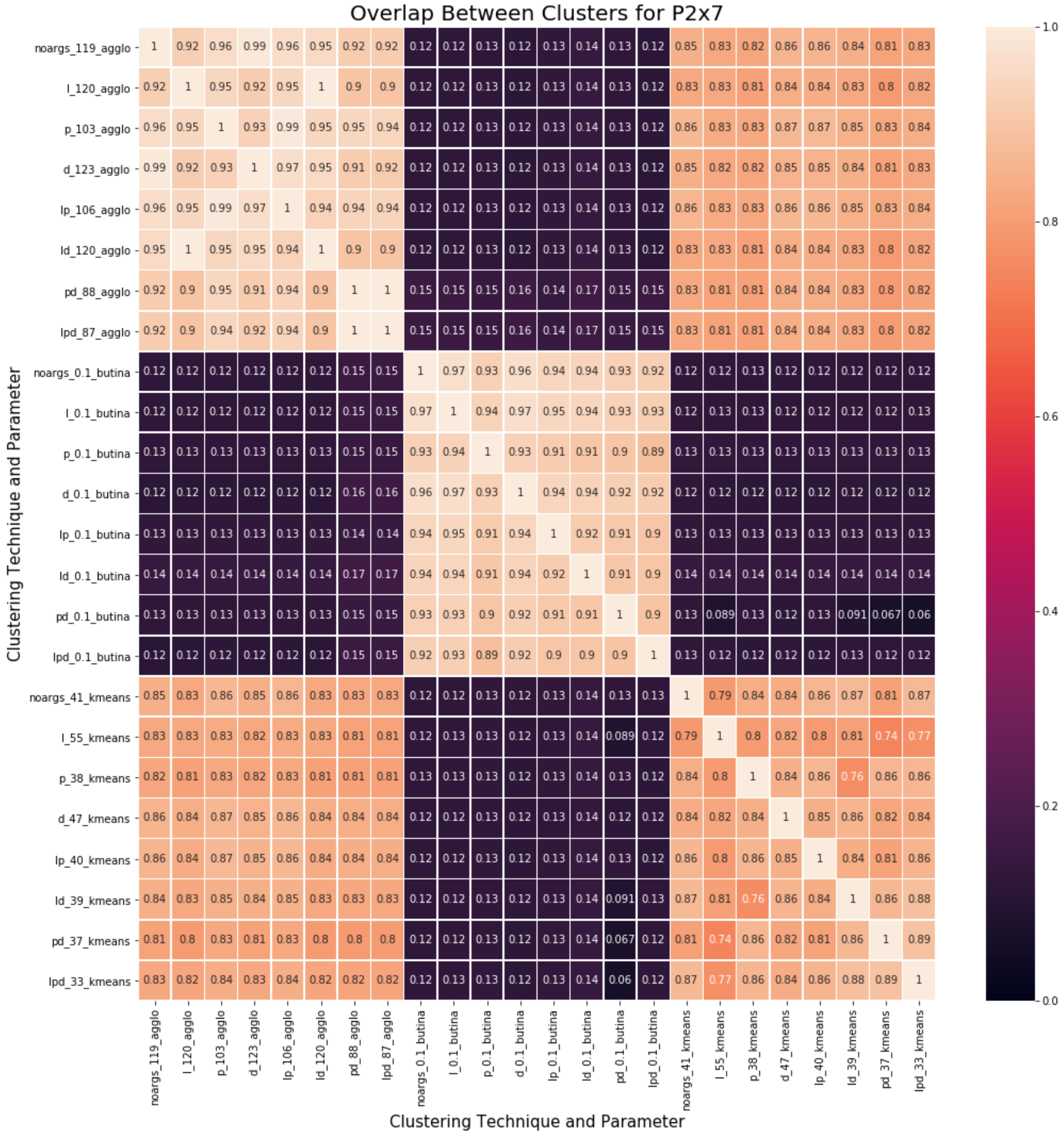
	noargs_3_agglo	l_3_agglo	p_3_agglo	d_3_agglo	lp_3_agglo	ld_3_agglo	pd_3_agglo	lpd_3_agglo	noargs_0.1_butina	l_0.2_butina	p_0.1_butina	d_0.2_butina	lp_0.1_butina	ld_0.1_butina	pd_0.1_butina	lpd_0.1_butina	noargs_3_kmeans	l_4_kmeans	p_4_kmeans	d_3_kmeans	lp_4_kmeans	ld_4_kmeans	pd_4_kmeans	lpd_4_kmeans
noargs_3_agglo	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.37	0.34	0.39	0.4	0.37	0.39	0.37	0.37	1	0.86	0.85	1	0.85	0.86	0.85	0.85
l_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
p_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
d_3_agglo	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.37	0.34	0.39	0.4	0.37	0.39	0.37	0.37	1	0.86	0.85	1	0.85	0.86	0.85	0.85
lp_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
ld_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
pd_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
lpd_3_agglo	0.98	1	1	0.98	1	1	1	1	0.37	0.28	0.39	0.26	0.37	0.39	0.37	0.37	0.98	0.85	0.85	0.98	0.85	0.85	0.85	0.85
noargs_0.1_butina	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	1	0.64	0.98	0.68	0.98	0.98	0.98	0.98	0.32	0.37	0.3	0.37	0.37	0.31	0.37	0.3
l_0.2_butina	0.34	0.28	0.28	0.34	0.28	0.28	0.28	0.28	0.64	1	0.64	0.61	0.66	0.66	0.66	0.66	0.36	0.3	0.31	0.36	0.31	0.3	0.31	0.31
p_0.1_butina	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.98	0.64	1	0.7	0.98	0.98	0.98	0.98	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39
d_0.2_butina	0.4	0.26	0.26	0.4	0.26	0.26	0.26	0.26	0.68	0.61	0.7	1	0.68	0.7	0.68	0.68	0.39	0.36	0.35	0.39	0.35	0.36	0.35	0.35
lp_0.1_butina	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.98	0.66	0.98	0.68	1	0.98	0.98	0.98	0.32	0.37	0.3	0.37	0.37	0.31	0.37	0.3
ld_0.1_butina	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.98	0.66	0.98	0.7	0.98	1	0.98	0.98	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39
pd_0.1_butina	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.98	0.66	0.98	0.68	0.98	0.98	1	1	0.32	0.37	0.3	0.37	0.37	0.31	0.37	0.3
lpd_0.1_butina	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.98	0.66	0.98	0.68	0.98	0.98	1	1	0.32	0.37	0.3	0.37	0.37	0.31	0.37	0.3
noargs_3_kmeans	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.32	0.36	0.39	0.39	0.32	0.39	0.32	0.32	1	0.86	0.85	1	0.85	0.86	0.85	0.85
l_4_kmeans	0.86	0.85	0.85	0.86	0.85	0.85	0.85	0.85	0.37	0.3	0.39	0.36	0.37	0.39	0.37	0.37	0.86	1	0.99	0.86	0.99	1	0.99	0.99
p_4_kmeans	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.3	0.31	0.39	0.35	0.3	0.39	0.3	0.3	0.85	0.99	1	0.85	1	0.99	1	1
d_3_kmeans	1	0.98	0.98	1	0.98	0.98	0.98	0.98	0.37	0.36	0.39	0.39	0.37	0.39	0.37	0.37	1	0.86	0.85	1	0.85	0.86	0.85	0.85
lp_4_kmeans	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.37	0.31	0.39	0.35	0.37	0.39	0.37	0.37	0.85	0.99	1	0.85	1	0.99	1	1
ld_4_kmeans	0.86	0.85	0.85	0.86	0.85	0.85	0.85	0.85	0.31	0.3	0.39	0.36	0.31	0.39	0.31	0.31	0.86	1	0.99	0.86	0.99	1	0.99	0.99
pd_4_kmeans	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.37	0.31	0.39	0.35	0.37	0.39	0.37	0.37	0.85	0.99	1	0.85	1	0.99	1	1
lpd_4_kmeans	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.3	0.31	0.39	0.35	0.3	0.39	0.3	0.3	0.85	0.99	1	0.85	1	0.99	1	1

Figure 0-3: Chk1 heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap



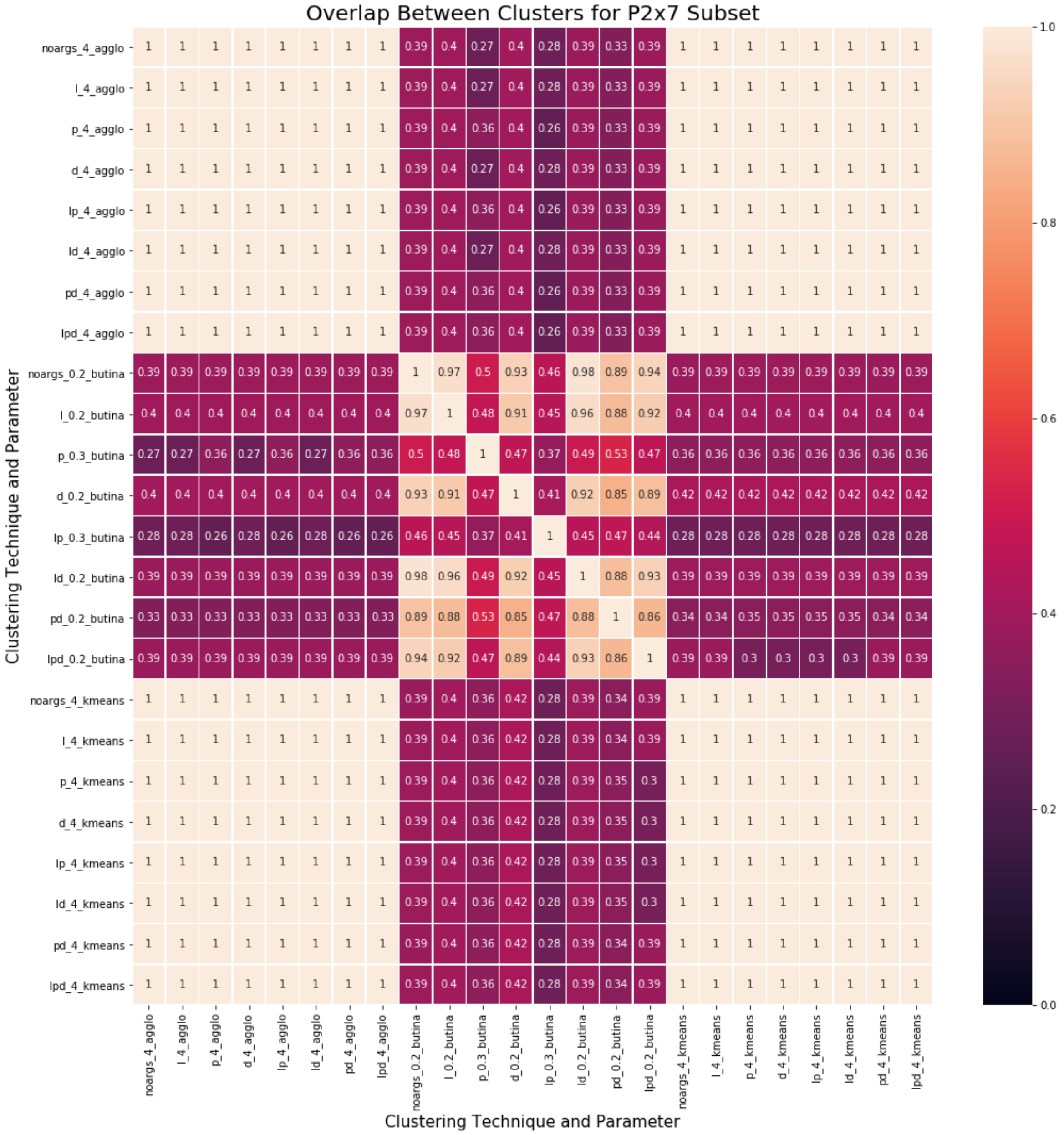
	noargs_92_agglo	l_112_agglo	p_113_agglo	d_103_agglo	lp_113_agglo	ld_117_agglo	pd_118_agglo	lpd_118_agglo	noargs_0.1_butina	l_0.1_butina	p_0.1_butina	d_0.1_butina	lp_0.1_butina	ld_0.1_butina	pd_0.1_butina	lpd_0.1_butina	noargs_119_kmeans	l_93_kmeans	p_90_kmeans	d_76_kmeans	lp_76_kmeans	ld_87_kmeans	pd_76_kmeans	lpd_77_kmeans
noargs_92_agglo	1	0.9	0.9	0.97	0.9	0.9	0.89	0.89	0.092	0.095	0.092	0.094	0.094	0.092	0.091	0.092	0.57	0.59	0.65	0.69	0.67	0.67	0.67	0.68
l_112_agglo	0.9	1	0.99	0.91	0.99	0.99	0.98	0.98	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.61	0.68	0.7	0.74	0.73	0.72	0.72	0.73
p_113_agglo	0.9	0.99	1	0.9	1	0.98	0.99	0.99	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.62	0.69	0.7	0.74	0.74	0.72	0.72	0.72
d_103_agglo	0.97	0.91	0.9	1	0.86	0.93	0.88	0.92	0.092	0.095	0.092	0.094	0.094	0.092	0.091	0.092	0.57	0.64	0.66	0.69	0.68	0.68	0.69	0.7
lp_113_agglo	0.9	0.99	1	0.86	1	0.95	0.99	0.99	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.62	0.69	0.7	0.74	0.74	0.72	0.72	0.72
ld_117_agglo	0.9	0.99	0.98	0.93	0.95	1	0.99	0.99	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.62	0.68	0.7	0.73	0.72	0.72	0.72	0.72
pd_118_agglo	0.89	0.98	0.99	0.88	0.99	0.99	1	1	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.61	0.68	0.7	0.73	0.73	0.72	0.72	0.72
lpd_118_agglo	0.89	0.98	0.99	0.92	0.99	0.99	1	1	0.09	0.091	0.09	0.09	0.09	0.091	0.091	0.09	0.61	0.68	0.7	0.73	0.73	0.72	0.72	0.72
noargs_0.1_butina	0.092	0.09	0.09	0.092	0.09	0.09	0.09	0.09	1	0.98	0.95	0.98	0.96	0.97	0.96	0.97	0.042	0.051	0.048	0.054	0.055	0.052	0.048	0.049
l_0.1_butina	0.095	0.091	0.091	0.095	0.091	0.091	0.091	0.091	0.98	1	0.95	0.98	0.95	0.97	0.96	0.97	0.037	0.046	0.042	0.048	0.05	0.047	0.042	0.044
p_0.1_butina	0.092	0.09	0.09	0.092	0.09	0.09	0.09	0.09	0.95	0.95	1	0.95	0.93	0.94	0.94	0.94	0.037	0.05	0.047	0.048	0.049	0.051	0.047	0.047
d_0.1_butina	0.094	0.09	0.09	0.094	0.09	0.09	0.09	0.09	0.98	0.98	0.95	1	0.96	0.98	0.96	0.97	0.037	0.044	0.042	0.047	0.049	0.046	0.042	0.043
lp_0.1_butina	0.094	0.09	0.09	0.094	0.09	0.09	0.09	0.09	0.96	0.95	0.93	0.96	1	0.95	0.93	0.94	0.042	0.049	0.046	0.053	0.054	0.051	0.046	0.047
ld_0.1_butina	0.092	0.091	0.091	0.092	0.091	0.091	0.091	0.091	0.97	0.97	0.94	0.98	0.95	1	0.95	0.97	0.038	0.046	0.041	0.049	0.051	0.047	0.041	0.044
pd_0.1_butina	0.091	0.091	0.091	0.091	0.091	0.091	0.091	0.091	0.96	0.96	0.94	0.96	0.93	0.95	1	0.94	0.035	0.047	0.042	0.047	0.048	0.048	0.042	0.044
lpd_0.1_butina	0.092	0.09	0.09	0.092	0.09	0.09	0.09	0.09	0.97	0.97	0.94	0.97	0.94	0.97	0.94	1	0.038	0.047	0.042	0.049	0.051	0.048	0.042	0.042
noargs_119_kmeans	0.57	0.61	0.62	0.57	0.62	0.62	0.61	0.61	0.042	0.037	0.037	0.037	0.042	0.038	0.035	0.038	1	0.76	0.77	0.76	0.7	0.77	0.68	0.68
l_93_kmeans	0.59	0.68	0.69	0.64	0.69	0.68	0.68	0.68	0.051	0.046	0.05	0.044	0.049	0.046	0.047	0.047	0.76	1	0.8	0.74	0.8	0.83	0.77	0.74
p_90_kmeans	0.65	0.7	0.7	0.66	0.7	0.7	0.7	0.7	0.048	0.042	0.047	0.042	0.046	0.041	0.042	0.042	0.77	0.8	1	0.76	0.8	0.83	0.79	0.78
d_76_kmeans	0.69	0.74	0.74	0.69	0.74	0.73	0.73	0.73	0.054	0.048	0.048	0.047	0.053	0.049	0.047	0.049	0.76	0.74	0.76	1	0.76	0.78	0.76	0.72
lp_76_kmeans	0.67	0.73	0.74	0.68	0.74	0.72	0.73	0.73	0.055	0.05	0.049	0.049	0.054	0.051	0.048	0.051	0.7	0.8	0.8	0.76	1	0.77	0.8	0.71
ld_87_kmeans	0.67	0.72	0.72	0.68	0.72	0.72	0.72	0.72	0.052	0.047	0.051	0.046	0.051	0.047	0.048	0.048	0.77	0.83	0.83	0.78	0.77	1	0.79	0.76
pd_76_kmeans	0.67	0.72	0.72	0.69	0.72	0.72	0.72	0.72	0.048	0.042	0.047	0.042	0.046	0.041	0.042	0.042	0.68	0.77	0.79	0.76	0.8	0.79	1	0.78
lpd_77_kmeans	0.68	0.73	0.72	0.7	0.72	0.72	0.72	0.72	0.049	0.044	0.047	0.043	0.047	0.044	0.044	0.042	0.68	0.74	0.78	0.72	0.71	0.76	0.78	1

Figure 0-4: FactorXa heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap



	noargs_119_agglo	l_120_agglo	p_103_agglo	d_123_agglo	lp_106_agglo	ld_120_agglo	pd_88_agglo	lpd_87_agglo	noargs_0.1_butina	l_0.1_butina	p_0.1_butina	d_0.1_butina	lp_0.1_butina	ld_0.1_butina	pd_0.1_butina	lpd_0.1_butina	noargs_41_kmeans	l_55_kmeans	p_38_kmeans	d_47_kmeans	lp_40_kmeans	ld_39_kmeans	pd_37_kmeans	lpd_33_kmeans
noargs_119_agglo	1	0.92	0.96	0.99	0.96	0.95	0.92	0.92	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.85	0.83	0.82	0.86	0.86	0.84	0.81	0.83
l_120_agglo	0.92	1	0.95	0.92	0.95	1	0.9	0.9	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.83	0.83	0.81	0.84	0.84	0.83	0.8	0.82
p_103_agglo	0.96	0.95	1	0.93	0.99	0.95	0.95	0.94	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.86	0.83	0.83	0.87	0.87	0.85	0.83	0.84
d_123_agglo	0.99	0.92	0.93	1	0.97	0.95	0.91	0.92	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.85	0.82	0.82	0.85	0.85	0.84	0.81	0.83
lp_106_agglo	0.96	0.95	0.99	0.97	1	0.94	0.94	0.94	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.86	0.83	0.83	0.86	0.86	0.85	0.83	0.84
ld_120_agglo	0.95	1	0.95	0.95	0.94	1	0.9	0.9	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.83	0.83	0.81	0.84	0.84	0.83	0.8	0.82
pd_88_agglo	0.92	0.9	0.95	0.91	0.94	0.9	1	1	0.15	0.15	0.15	0.16	0.14	0.17	0.15	0.15	0.83	0.81	0.81	0.84	0.84	0.83	0.8	0.82
lpd_87_agglo	0.92	0.9	0.94	0.92	0.94	0.9	1	1	0.15	0.15	0.15	0.16	0.14	0.17	0.15	0.15	0.83	0.81	0.81	0.84	0.84	0.83	0.8	0.82
noargs_0.1_butina	0.12	0.12	0.12	0.12	0.12	0.12	0.15	0.15	1	0.97	0.93	0.96	0.94	0.94	0.93	0.92	0.12	0.12	0.13	0.12	0.12	0.12	0.12	0.12
l_0.1_butina	0.12	0.12	0.12	0.12	0.12	0.12	0.15	0.15	0.97	1	0.94	0.97	0.95	0.94	0.93	0.93	0.12	0.13	0.13	0.12	0.12	0.12	0.12	0.13
p_0.1_butina	0.13	0.13	0.13	0.13	0.13	0.13	0.15	0.15	0.93	0.94	1	0.93	0.91	0.91	0.9	0.89	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
d_0.1_butina	0.12	0.12	0.12	0.12	0.12	0.12	0.16	0.16	0.96	0.97	0.93	1	0.94	0.94	0.92	0.92	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
lp_0.1_butina	0.13	0.13	0.13	0.13	0.13	0.13	0.14	0.14	0.94	0.95	0.91	0.94	1	0.92	0.91	0.9	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
ld_0.1_butina	0.14	0.14	0.14	0.14	0.14	0.14	0.17	0.17	0.94	0.94	0.91	0.94	0.92	1	0.91	0.9	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
pd_0.1_butina	0.13	0.13	0.13	0.13	0.13	0.13	0.15	0.15	0.93	0.93	0.9	0.92	0.91	0.91	1	0.9	0.13	0.089	0.13	0.12	0.13	0.091	0.067	0.06
lpd_0.1_butina	0.12	0.12	0.12	0.12	0.12	0.12	0.15	0.15	0.92	0.93	0.89	0.92	0.9	0.9	0.9	1	0.13	0.12	0.12	0.12	0.12	0.13	0.12	0.12
noargs_41_kmeans	0.85	0.83	0.86	0.85	0.86	0.83	0.83	0.83	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.13	1	0.79	0.84	0.84	0.86	0.87	0.81	0.87
l_55_kmeans	0.83	0.83	0.83	0.82	0.83	0.83	0.81	0.81	0.12	0.13	0.13	0.12	0.13	0.14	0.089	0.12	0.79	1	0.8	0.82	0.8	0.81	0.74	0.77
p_38_kmeans	0.82	0.81	0.83	0.82	0.83	0.81	0.81	0.81	0.13	0.13	0.13	0.12	0.13	0.14	0.13	0.12	0.84	0.8	1	0.84	0.86	0.76	0.86	0.86
d_47_kmeans	0.86	0.84	0.87	0.85	0.86	0.84	0.84	0.84	0.12	0.12	0.13	0.12	0.13	0.14	0.12	0.12	0.84	0.82	0.84	1	0.85	0.86	0.82	0.84
lp_40_kmeans	0.86	0.84	0.87	0.85	0.86	0.84	0.84	0.84	0.12	0.12	0.13	0.12	0.13	0.14	0.13	0.12	0.86	0.8	0.86	0.85	1	0.84	0.81	0.86
ld_39_kmeans	0.84	0.83	0.85	0.84	0.85	0.83	0.83	0.83	0.12	0.12	0.13	0.12	0.13	0.14	0.091	0.13	0.87	0.81	0.76	0.86	0.84	1	0.86	0.88
pd_37_kmeans	0.81	0.8	0.83	0.81	0.83	0.8	0.8	0.8	0.12	0.12	0.13	0.12	0.13	0.14	0.067	0.12	0.81	0.74	0.86	0.82	0.81	0.86	1	0.89
lpd_33_kmeans	0.83	0.82	0.84	0.83	0.84	0.82	0.82	0.82	0.12	0.13	0.13	0.12	0.13	0.14	0.06	0.12	0.87	0.77	0.86	0.84	0.86	0.88	0.89	1

Figure 0-5: P2x7 heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap



	noargs_4_agglo	l_4_agglo	p_4_agglo	d_4_agglo	lp_4_agglo	ld_4_agglo	pd_4_agglo	lpd_4_agglo	noargs_0.2_butina	l_0.2_butina	p_0.3_butina	d_0.2_butina	lp_0.3_butina	ld_0.2_butina	pd_0.2_butina	lpd_0.2_butina	noargs_4_kmeans	l_4_kmeans	p_4_kmeans	d_4_kmeans	lp_4_kmeans	ld_4_kmeans	pd_4_kmeans	lpd_4_kmeans
noargs_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.27	0.4	0.28	0.39	0.33	0.39	1	1	1	1	1	1	1	1
l_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.27	0.4	0.28	0.39	0.33	0.39	1	1	1	1	1	1	1	1
p_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.4	0.26	0.39	0.33	0.39	1	1	1	1	1	1	1	1
d_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.27	0.4	0.28	0.39	0.33	0.39	1	1	1	1	1	1	1	1
lp_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.4	0.26	0.39	0.33	0.39	1	1	1	1	1	1	1	1
ld_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.27	0.4	0.28	0.39	0.33	0.39	1	1	1	1	1	1	1	1
pd_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.4	0.26	0.39	0.33	0.39	1	1	1	1	1	1	1	1
lpd_4_agglo	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.4	0.26	0.39	0.33	0.39	1	1	1	1	1	1	1	1
noargs_0.2_butina	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	1	0.97	0.5	0.93	0.46	0.98	0.89	0.94	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39
l_0.2_butina	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.97	1	0.48	0.91	0.45	0.96	0.88	0.92	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
p_0.3_butina	0.27	0.27	0.36	0.27	0.36	0.27	0.36	0.36	0.5	0.48	1	0.47	0.37	0.49	0.53	0.47	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
d_0.2_butina	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.93	0.91	0.47	1	0.41	0.92	0.85	0.89	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
lp_0.3_butina	0.28	0.28	0.26	0.28	0.26	0.28	0.26	0.26	0.46	0.45	0.37	0.41	1	0.45	0.47	0.44	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
ld_0.2_butina	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.98	0.96	0.49	0.92	0.45	1	0.88	0.93	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39
pd_0.2_butina	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.89	0.88	0.53	0.85	0.47	0.88	1	0.86	0.34	0.34	0.35	0.35	0.35	0.35	0.34	0.34
lpd_0.2_butina	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.94	0.92	0.47	0.89	0.44	0.93	0.86	1	0.39	0.39	0.3	0.3	0.3	0.3	0.39	0.39
noargs_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.34	0.39	1	1	1	1	1	1	1	1
l_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.34	0.39	1	1	1	1	1	1	1	1
p_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.35	0.3	1	1	1	1	1	1	1	1
d_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.35	0.3	1	1	1	1	1	1	1	1
lp_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.35	0.3	1	1	1	1	1	1	1	1
ld_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.35	0.3	1	1	1	1	1	1	1	1
pd_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.34	0.39	1	1	1	1	1	1	1	1
lpd_4_kmeans	1	1	1	1	1	1	1	1	0.39	0.4	0.36	0.42	0.28	0.39	0.34	0.39	1	1	1	1	1	1	1	1

Figure 0-6: P2x7 subset Neurokinin heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap

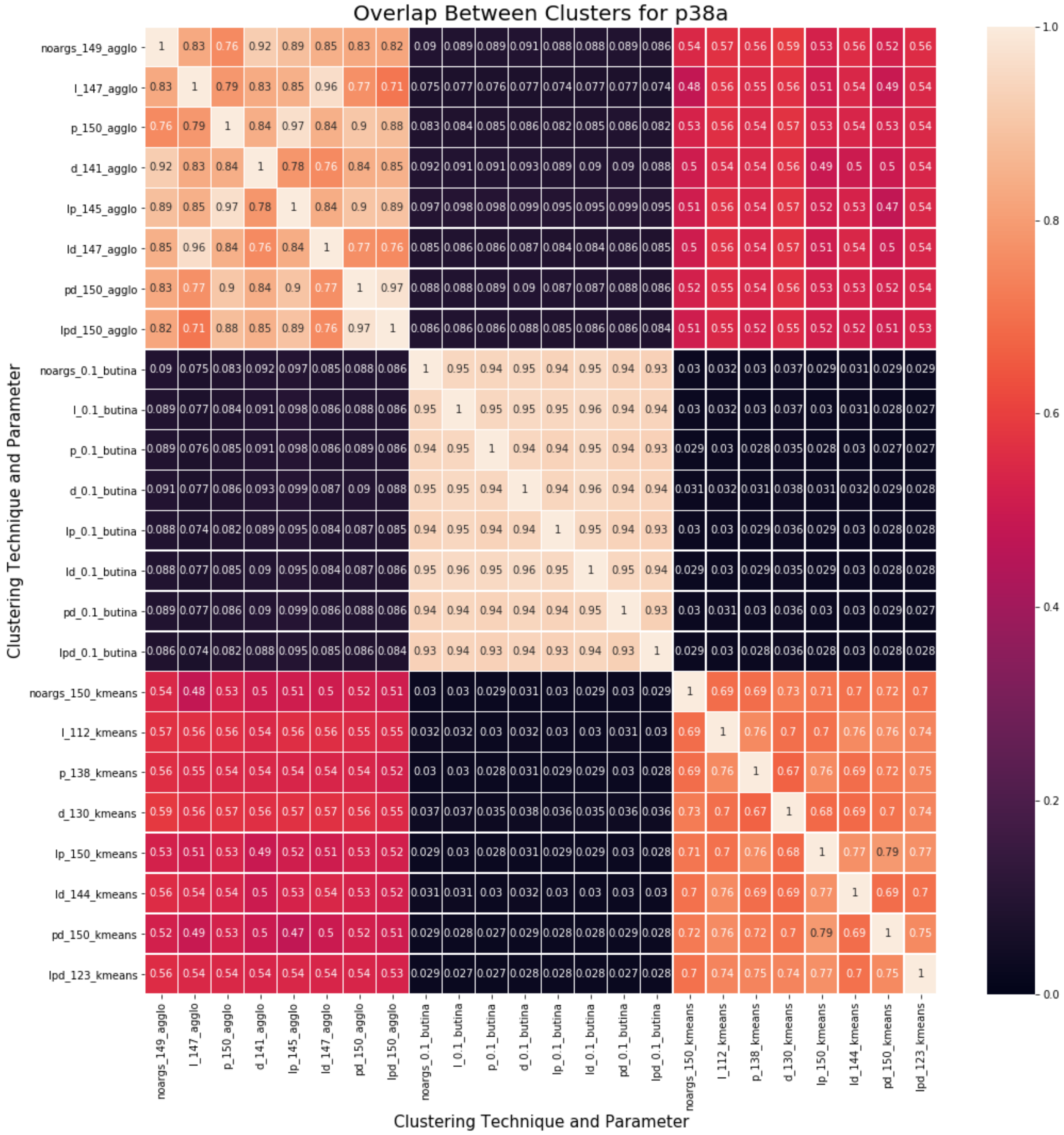


Figure 0-7: p38α heatmap of the overlap between the clusters all of the different top results from the different parameters and clustering techniques. The closer to one the more overlap

Bajorath

*Table 0-2: Bajorath table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		Dunn delta 1		Dunn delta 2		Davies Bouldin delta 1		Davies Bouldin delta 2		Calinski Harabasz		Ball-Hall	Kelley	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	28	0.493	28	0.823	28	1.188	28	1.945	49	1.411	39	1665.290	25	5	1088.117
	Butina	0.1	-0.008	0.1	0.040	0.1	0.990	0.1	49.383	0.4	1.184	0.7	5.089	0.2	0.1	1274.500
	K-means	29	0.469	21	0.680	2	0.992	21	2.185	43	1.589	39	2407.769	14	59	551.698
RG (default) connected	Agglomerative	30	0.474	2	0.688	30	1.178	3	2.070	147	1.281	29	431.020	15	3	411.701
	Butina	0.1	-0.043	0.1	0.067	0.1	0.957	0.1	25.692	0.8	1.602	0.8	0.152	0.2	0.2	308.060
	K-means	28	0.457	2	0.455	29	1.041	29	3.382	145	1.460	34	466.300	14	118	242.458
RG (default) disconnected	Agglomerative	145	0.233	7	0.106	2	1	149	5.485	146	1.315	57	134.682	19	5	338.172
	Butina	0.2	-0.031	-	-	0.2	0.867	-	-	0.3	1.509	0.8	0.152	0.4	0.6	338.756
	K-means	148	0.205	149	0.178	2	1	149	6.493	148	1.515	2	580.139	11	66	240.615
RGFP (default)	Agglomerative	40	0.331	2	0.684	28	1.033	147	1.962	147	1.356	40	260.480	14	19	400.960
	Butina	0.1	-0.097	-	-	-	-	0.5	5.361	0.5	1.673	0.5	0.155	0.2	0.1	418.787
	K-means	38	0.327	16	0.393	2	1	149	2.676	149	1.669	46	406.774	14	72	259.626
Chemical graph connected	Agglomerative	31	0.621	28	0.744	30	1.250	29	1.835	57	1.276	30	2145.631	23	21	1067.316
	Butina	0.1	0.010	-	-	0.1	0.972	-	-	0.5	1.203	0.9	6.290	0.2	0.1	1274.500
	K-means	30	0.619	14	0.470	29	1.025	29	2.358	30	1.473	41	2696.857	15	86	615.169

Chemical graph disconnected	Agglomerative	47	0.430	2	0.271	2	0.990	2	5.779	148	1.340	39	888.079	14	10	1118.358
	Butina	0.4	-0.088	-	-	0.4	0.959	-	-	0.8	1.558	0.9	1.854	0.5	0.5	1246.440
	K-means	35	0.381	-	-	2	1.061	128	11.965	4	0.905	35	960.648	11	102	631.734

CDK2

*Table 0-3: CDK2 table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		**Dunn delta 1**		**Dunn delta 2**		**Davies Bouldin delta 1**		**Davies Bouldin delta 2**		**Calinski Harabasz**		**Ball-Hall**	**Kelley**	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	150	0.340	-	-	-	-	2	1.281	2	1.002	136	67.731	133	2	2.000
	Butina	0.1	-0.087	-	-	-	-	-	-	0.7	1.464	0.6	2.138	0.2	0.1	727.653
	K-means	117	0.295	2	0.538	2	0.982	2	3.147	148	1.675	2	372.872	138	65	470.306
RG (default) connected	Agglomerative	146	0.330	2	0.667	2	1.000	145	2.149	149	1.286	83	45.479	15	2	412.000
	Butina	0.1	-0.059	-	-	-	-	0.1	16.037	0.6	1.614	0.6	0.129	0.2	0.1	336.703
	K-means	107	0.293	107	0.179	2	1.000	107	4.469	148	1.667	2	165.648	12	76	279.357
RG (default) disconnected	Agglomerative	146	0.230	2	0.333	2	1.000	149	4.663	128	1.292	136	64.305	17	6	274.272
	Butina	0.2	0.038	-	-	0.2	1.000	-	-	0.4	1.742	0.8	0.270	0.3	0.5	405.021
	K-means	145	0.196	112	0.167	2	1.000	149	6.090	149	1.583	5	220.763	10	57	175.342
RGFP (default)	Agglomerative	149	0.182	2	0.700	2	1.000	150	2.160	150	1.561	20	26.680	132	3	405.669
	Butina	0.1	-0.071	-	-	-	-	0.4	3.863	0.4	1.541	0.2	0.126	0.2	0.2	461.762
	K-means	142	0.159	140	0.425	2	1.000	149	2.888	2	2.000	3	144.070	139	142	364.007

Chemical graph connected	Agglomerative	149	0.462	3	0.552	3	1.006	148	2.064	149	1.252	149	87.749	21	2	684.000
	Butina	0.2	-0.172	-	-	-	-	-	-	0.8	1.490	0.7	2.377	0.3	0.8	1335.862
	K-means	88	0.438	2	0.214	13	0.971	149	3.976	140	1.591	104	369.631	15	37	390.912
Chemical graph disconnected	Agglomerative	144	0.356	2	0.350	3	1.035	150	4.249	150	1.331	-	-	-	9	591.096
	Butina	0.3	-0.064	-	-	0.3	0.949	-	-	0.3	1.664	0.9	0.840	0.5	0.5	460.538
	K-means	59	0.248	46	0.057	5	0.959	141	8.243	3	1.179	-	-	-	117	357.108

Chk1

*Table 0-4: Chk1 table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		Dunn delta 1		Dunn delta 2		Davies Bouldin delta 1		Davies Bouldin delta 2		Calinski Harabasz		Ball-Hall	Kelley	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	3	0.334	3	0.783	3	1.150	34	1.724	34	1.145	3	137.690	6	3	44.597
	Butina	0.1	-0.009	-	-	-	-	0.1	3.318	0.1	1.016	0.4	0.230	0.2	0.1	2.000
	K-means	3	0.334	3	0.783	3	1.150	39	1.949	40	1.320	5	150.776	5	17	36.435
RG (default) connected	Agglomerative	3	0.363	5	0.661	4	1.091	33	1.966	33	1.061	3	81.459	6	3	36.139
	Butina	0.1	-0.038	-	-	-	-	0.1	3.339	0.1	1.028	0.5	0.147	0.2	0.1	2.000
	K-means	3	0.363	2	0.640	11	1.099	36	2.183	32	1.194	6	106.156	6	24	26.829
RG (default) disconnected	Agglomerative	3	0.351	6	0.389	6	1.167	30	2.763	29	1.028	3	68.762	6	10	29.945
	Butina	0.3	0.016	-	-	0.3	1.053	-	-	0.7	1.533	0.6	0.351	0.5	0.3	45.500
	K-means	4	0.346	7	0.250	4	1.249	35	3.006	35	1.211	5	71.006	5	8	18.578
	Agglomerative	33	0.198	3	0.740	5	1.075	33	1.809	33	1.269	33	54.773	6	6	36.417

RGFP (default)	Butina	0.1	-0.061	0.1	0.286	0.1	0.946	0.4	3.953	0.5	1.607	0.5	0.196	0.2	0.1	59.291
	K-means	4	0.197	3	0.740	3	1.028	33	1.924	33	1.340	34	109.151	6	23	34.445
Chemical graph connected	Agglomerative	3	0.435	3	0.808	3	1.180	6	1.783	34	1.055	3	156.170	4	4	34.024
	Butina	0.2	-0.085	0.2	0.038	0.2	0.980	0.7	23.509	0.5	1.498	0.7	0.142	0.3	0.4	42.409
	K-means	3	0.435	3	0.808	3	1.180	36	2.083	39	1.126	3	156.170	5	9	26.925
Chemical graph disconnected	Agglomerative	29	0.337	7	0.292	7	1.127	38	2.168	38	1.006	34	79.259	6	8	36.043
	Butina	0.5	-0.160	-	-	-	-	0.5	21.529	0.5	1.612	0.8	0.180	0.6	0.6	38.711
	K-means	4	0.331	19	0.244	10	1.111	38	2.968	40	1.181	36	84.558	6	21	33.221

Cyto

*Table 0-5: Cyto table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		**Dunn delta 1**		**Dunn delta 2**		**Davies Bouldin delta 1**		**Davies Bouldin delta 2**		**Kelley**	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**
M2FP	Agglomerative	2	0.111	2	0.909	2	1	2	1.833	9	1.616	2	3185.000
	Butina	0.1 (9)	-0.017	-	-	-	-	0.1	4.086	0.9	1.347	0.1	3526.830
	K-means	150	0.087	2	0.176	2	1	146	7.743	139	1.885	137	2624.225
RG (default) connected	Agglomerative	135	0.123	-	-	-	-	2	1.056	2	1	2	2
	Butina	0.1 (7)	-0.074	-	-	-	-	0.9	8.016	0.9	1.457	0.1	1535.803
	K-means	102	0.101	69	0.110	2	1	145	1512.866	148	1865.893	140	1379.629

RG (default) disconnected	Agglomerative	2	0.258	2	0.500	2	1	2	3	2	1.5	5	1189.142
	Butina	0.3 (7)	-0.125	-	-	-	-	0.1	3.086	0.1	0.925	0.3	907.263
	K-means	150	0.145	33	0.088	2	1	2	0	2	0	95	899.808
RGFP (default)	Agglomerative	2	0.073	2	0.667	2	1	4	2.411	4	1.739	6	1.818
	Butina	0.1 (27)	-0.088	-	-	-	-	0.9	3.912	0.9	1.339	0.1	2004.177
	K-means	149	0.048	46	0.250	2	1	150	4.878	128	1.924	3	1624.015

FactorXa

*Table 0-6: FactorXa table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		**Dunn delta 1**		**Dunn delta 2**		**Davies Bouldin delta 1**		**Davies Bouldin delta 2**		**Calinski Harabasz**		**Ball-Hall**	**Kelley**	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	112	0.327	2	0.733	3	1.000	2	2.237	149	1.407	78	221.075	135	121	966.175
	Butina	3	-0.057	-	-	-	-	-	-	0.7	1.627	0.3	0.081	0.2	0.1	979.000
	K-means	71	0.294	116	0.214	11	0.992	146	4.713	148	1.681	8	486.413	140	143	541.413
RG (default) connected	Agglomerative	92	0.378	-	-	-	-	2	1.284	2	1.012	92	101.199	137	2	2.000
	Butina	0.1	0.008	0.1	0.115	0.1	0.995	0.7	13.631	0.7	1.604	0.4	0.124	0.2	0.2	199.131
	K-means	119	0.310	126	0.152	2	0.995	144	4.323	146	1.585	15	228.571	140	97	243.342
RG (default) disconnected	Agglomerative	150	0.243	2	0.263	2	0.978	150	4.992	149	1.258	68	93.994	136	45	402.822
	Butina	0.3	0.040	0.3	0.071	0.3	0.978	0.3	24.955	0.4	1.647	0.9	0.247	0.5	0.7	438.636

	K-means	138	0.215	117	0.153	2	1.003	148	6.503	149	1.573	2	292.942	142	86	127.288
RGFP (default)	Agglomerative	150	0.244	19	0.384	2	1.000	150	2.257	150	1.438	4	84.727	137	16	396.194
	Butina	0.1	-0.029	0.1	0.208	0.1	0.978	0.5	4.527	0.5	1.570	0.3	0.144	0.2	0.1	250.853
	K-means	92	0.198	101	0.408	2	1.000	149	2.994	149	1.714	10	151.413	140	124	307.499
Chemical graph connected	Agglomerative	138	0.475	2	0.502	2	0.991	150	2.345	150	1.177	71	260.229	135	76	936.862
	Butina	0.2	-0.133	-	-	-	-	-	-	0.8	1.597	0.9	0.161	0.3	0.2	1203.855
	K-means	83	0.458	4	0.088	2	1.022	137	5.712	150	1.587	83	791.987	140	82	448.386
Chemical graph disconnected	Agglomerative	137	0.358	2	0.186	2	0.984	150	5.545	150	1.278	137	286.442	140	3	895.361
	Butina	0.4	-0.121	-	-	-	-	-	-	0.5	1.579	0.9	0.163	0.6	0.5	529.314
	K-means	149	0.273	14	0.035	2	1.053	3	11.004	3	1.200	10	589.996	141	140	430.221

Neurokinin

*Table 0-7: Neurokinin table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		Dunn delta 1		Dunn delta 2		Davies Bouldin delta 1		Davies Bouldin delta 2		Calinski Harabasz		Ball-Hall	Kelley	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	2	0.139	-	-	-	-	2	1.042	2	1	137	15.635	20	2	2
	Butina	0.1	0.009	-	-	0.1	0.984	-	-	0.7	1.569	0.4	6.045	0.2	0.1	930.176
	K-means	140	0.181	142	0.195	2	1	140	5.355	149	1.789	2	1197.014	13	108	788.135
	Agglomerative	145	0.218	-	-	-	-	2	1.125	2	1	145	42.518	29	2	2

RG (default) connected	Butina	0.1	-0.044	-	-	-	-	-	-	0.7	1.614	0.7	0.160	0.2	0.1	697.602
	K-means	136	0.183	131	0.141	2	1	134	6.848	149	1.799	2	544.150	12	135	509.522
RG (default) disconnected	Agglomerative	2	0.326	-	-	-	-	2	1.667	2	1	146	58.128	30	2	2
	Butina	0.1	-0.018	-	-	0.1	1	-	-	0.6	1.637	0.7	0.439	0.4	0.2	602.629
	K-means	2	0.207	138	0.144	2	1	145	8.159	145	1.702	2	1405.862	10	78	337.511
RGFP (default)	Agglomerative	149	0.092	-	-	-	-	2	1.273	2	1	51	23.972	21	2	2
	Butina	0.1	-0.040	-	-	-	-	0.1	20.540	0.4	1.556	0.1	0.365	0.2	0.1	471.767
	K-means	147	0.098	146	0.386	2	1	147	3.612	142	1.871	2	846.657	11	103	678.009
Chemical graph connected	Agglomerative	149	0.244	2	1	2	1	6	1	6	1	90	54.169	56	6	6
	Butina	0.4	-0.295	-	-	-	-	-	-	0.9	1.594	0.4	2.607	0.5	0.4	1088.380
	K-means	129	0.256	2	0.032	2	1	14	29.751	139	1.729	2	1412.825	13	90	729.762
Chemical graph disconnected	Agglomerative	2	0.458	2	1	2	1	6	1	6	1	125	98.653	21	6	6
	Butina	0.2	0.157	-	-	0.2	1.000	-	-	0.4	1.714	0.5	8.768	0.4	0.7	898.759
	K-means	2	0.277	14	0.037	2	1	14	36.478	130	1.802	4	2269.532	10	147	504.826

P2x7

*Table 0-8: P2x7 table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		Dunn delta 1		Dunn delta 2		Davies Bouldin delta 1		Davies Bouldin delta 2		Calinski Harabasz		Ball-Hall	Kelley	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	67	0.260	-	-	-	-	2	1.314	2	1.006	46	181.373	23	2	2

	Butina	0.1	0.007	0.1	0.081	0.1	0.984	0.1	23.984	0.7	1.598	0.6	0.131	0.2	0.2	950.647
	K-means	64	0.251	131	0.210	3	0.997	147	5.503	139	1.756	2	584.958	13	105	570.017
RG (default) connected	Agglomerative	119	0.357	-	-	-	-	2	1.109	2	1.016	147	91.965	23	2	2
	Butina	0.1	-0.019	-	-	0.1	0.98	-	-	0.7	1.621	0.5	0.276	0.2	0.2	316.950
	K-means	41	0.288	92	0.187	2	1	149	4.836	149	1.566	3	220.233	11	96	180.057
RG (default) disconnected	Agglomerative	2	0.334	2	0.556	2	0.979	3	1.755	3	1.028	123	95.719	14	3	3
	Butina	0.2	-0.042	-	-	0.2	1	-	-	0.3	1.439	0.7	0.176	0.3	0.5	325.228
	K-means	146	0.258	109	0.157	2	1	149	5.651	145	1.481	3	333.035	10	68	156.084
RGFP (default)	Agglomerative	148	0.201	2	0.769	2	1	149	2.059	149	1.432	80	47.545	25	2	411
	Butina	0.1	-0.060	-	-	-	-	0.5	4.346	0.5	1.592	0.4	2.581	0.2	0.2	507.295
	K-means	83	0.177	83	0.457	2	1	138	2.822	145	1.738	2	245.570	14	46	325.903
Chemical graph connected	Agglomerative	2	0.206	-	-	-	-	2	1.531	3	0.997	78	265.125	39	2	2.000
	Butina	0.1	-0.043	-	-	-	-	0.1	26.794	0.1	1.033	0.7	0.529	0.8	0.1	2
	K-means	2	0.105	-	-	2	1	-	-	144	1.489	2	1507.468	14	34	650.010
Chemical graph disconnected	Agglomerative	2	0.195	2	0.040	2	1.025	2	30.823	146	1.208	126	309.908	14	3	844.780
	Butina	0.3	-0.015	-	-	0.3	0.949	-	-	0.4	1.767	0.9	0.182	0.6	0.6	784.970
	K-means	10	0.096	-	-	2	1	-	-	133	1.589	2	783.878	11	68	505.918

p38α

*Table 0-9: p38α table of the results from the clustering validity indexes (*or Tanimoto cut off)*

		Silhouette		**Dunn delta 1**		**Dunn delta 2**		**Davies Bouldin delta 1**		**Davies Bouldin delta 2**		**Calinski Harabasz**		**Ball-Hall**	**Kelley**	
Molecular Descriptor	**Clustering Algorithm**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Value**	**Clusters ***	**Clusters ***	**Value**
M2FP	Agglomerative	150	0.237	2	0.690	2	0.987	143	2.597	149	1.550	146	81.251	135	2	1822
	Butina	0.1	-0.037	-	-	-	-	-	-	0.7	1.541	0.5	0.149	0.2	0.1	2232.963
	K-means	139	0.232	146	0.181	2	1.001	147	5.668	147	1.798	3	600.727	138	104	1284.756
RG (default) connected	Agglomerative	149	0.252	2	0.500	2	1	145	3.167	145	1.457	33	79.455	130	5	823.771
	Butina	0.1	-0.057	-	-	-	-	0.1	18.031	0.7	1.635	0.7	0.197	0.2	0.1	655.673
	K-means	150	0.228	144	0.136	2	1	144	7.919	150	1.768	6	336.412	141	142	572.329
RG (default) disconnected	Agglomerative	2	0.278	-	-	-	-	2	2.667	2	1	137	86.089	126	2	2
	Butina	0.2	-0.064	-	-	0.2	0.929	-	-	0.2	1.722	0.8	0.163	0.5	0.3	706.157
	K-means	149	0.160	128	0.142	2	1	150	8.687	150	1.705	2	969.437	142	149	425.887
RGFP (default)	Agglomerative	148	0.122	2	0.833	2	1	4	2.002	4	1.665	34	34.827	133	14	913.106
	Butina	0.1	-0.051	-	-	-	-	0.4	4.155	0.4	1.646	0.4	0.208	0.2	0.1	877.901
	K-means	135	0.106	109	0.359	2	1	135	4.023	147	1.888	3	467.113	140	4	529.647

Table 0-10: Bajorath different molecular descriptors purity and v-measures

Molecular Descriptor	Clustering Algorithm	Cluster	Silhouette Score	Purity	V-measure	Mean of Purity and V-measure
M2FP	Agglomerative	28	0.493	0.994	0.972	0.809
	K-means	29	0.469	0.993	0.953	0.771
RG (default) connected	Agglomerative	30	0.474	0.994	0.959	0.646
	K-means	28	0.457	0.971	0.946	0.627
RG (default) disconnected	Agglomerative	145	0.233	0.875	0.717	0.074
	K-means	148	0.205	0.860	0.685	0.048
RGFP (default)	Agglomerative	40	0.331	0.994	0.939	0.662
	K-means	38	0.327	0.990	0.915	0.626
Chemical graph connected	Agglomerative	31	0.621	0.994	0.953	0.789
	K-means	30	0.619	0.993	0.949	0.779
Chemical graph disconnected	Agglomerative	47	0.430	0.993	0.914	0.279
	K-means	35	0.381	0.895	0.848	0.263

Table 0-11: P2x7 different molecular descriptors purity and v-measures

Molecular Descriptor	Clustering Algorithm	Cluster	Silhouette Score	Purity	V-measure	Mean of Purity and V-measure
M2FP	Agglomerative	67	0.260	0.817	0.843	0.830
	K-means	64	0.251	0.838	0.773	0.806
RG (default) connected	Agglomerative	119	0.357	0.926	0.904	0.915
	K-means	41	0.288	0.836	0.863	0.850
RG (default) disconnected	Agglomerative	2	0.334	0.123	0.004	0.063
	K-means	146	0.258	0.763	0.696	0.730
RGFP (default)	Agglomerative	148	0.201	0.872	0.814	0.843
	K-means	83	0.177	0.817	0.771	0.794
Chemical graph connected	Agglomerative	2	0.2056	0.123	0.001	0.062
	K-means	2	0.105	0.171	0.268	0.219

Chemical graph disconnected	Agglomerative	2	0.195	0.122	0.003	0.062
	K-means	10	0.096	0.420	0.509	0.464

RG Core Evaluation

Table 0-12: Table displaying the cores extracted from the two different processes for the Bajorath dataset, the number in the bracket is the number of examples of each core once re-examined

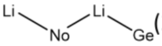
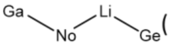
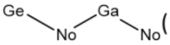
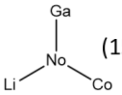
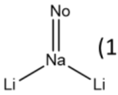
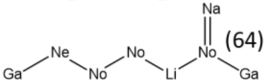
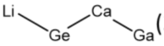
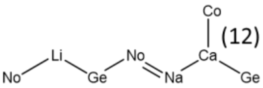
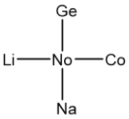
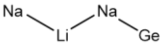
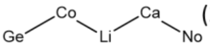
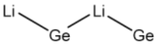
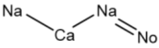
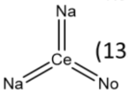
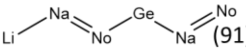
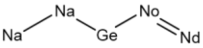
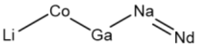
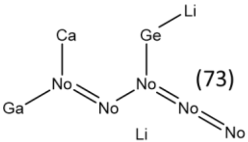
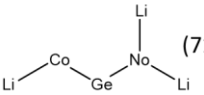
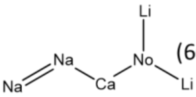
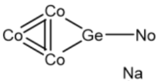
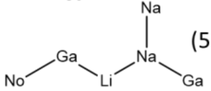
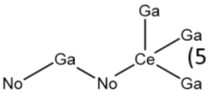
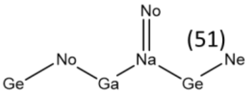
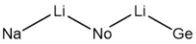
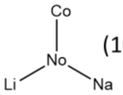
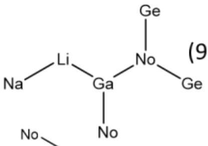
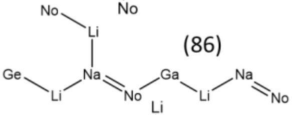
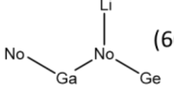
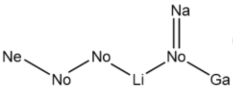
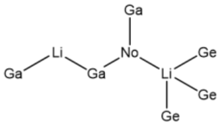
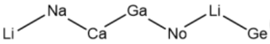
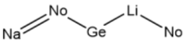
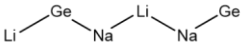
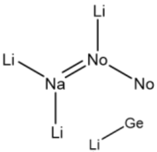
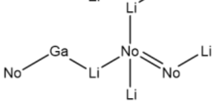
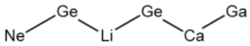
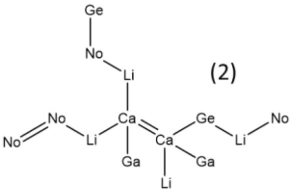
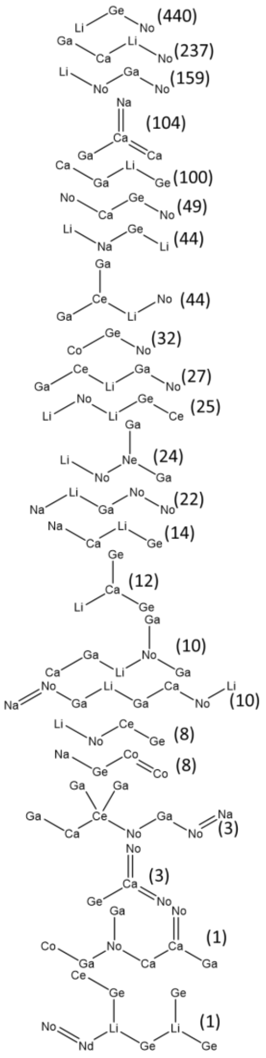
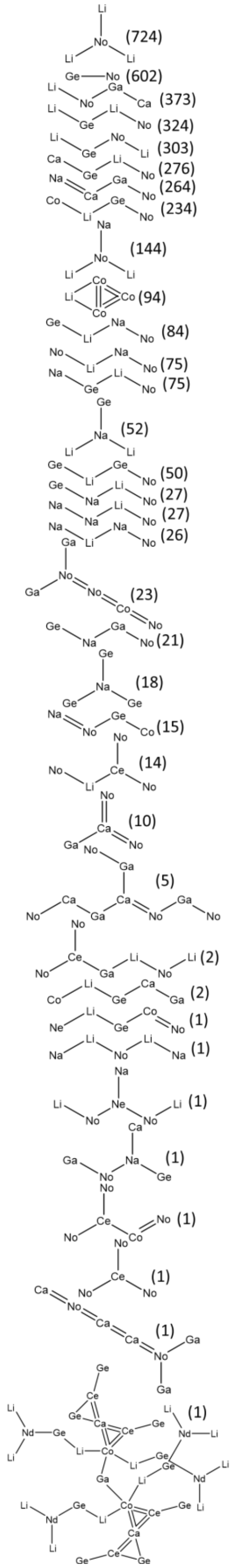
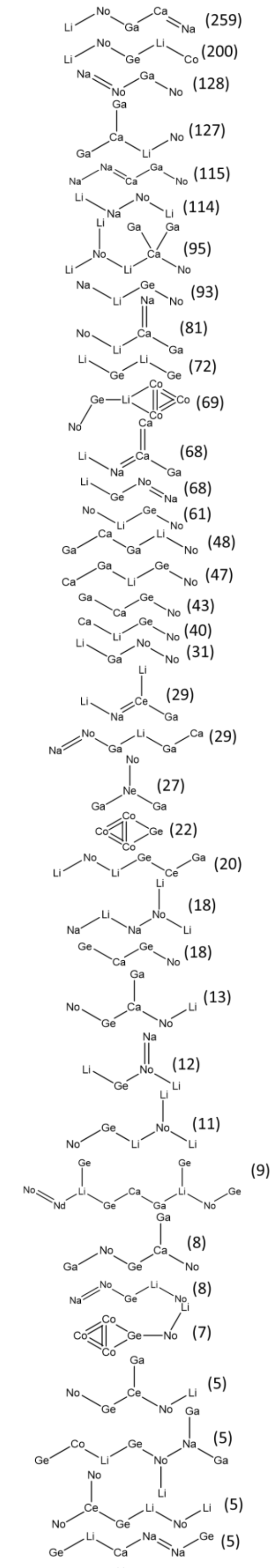
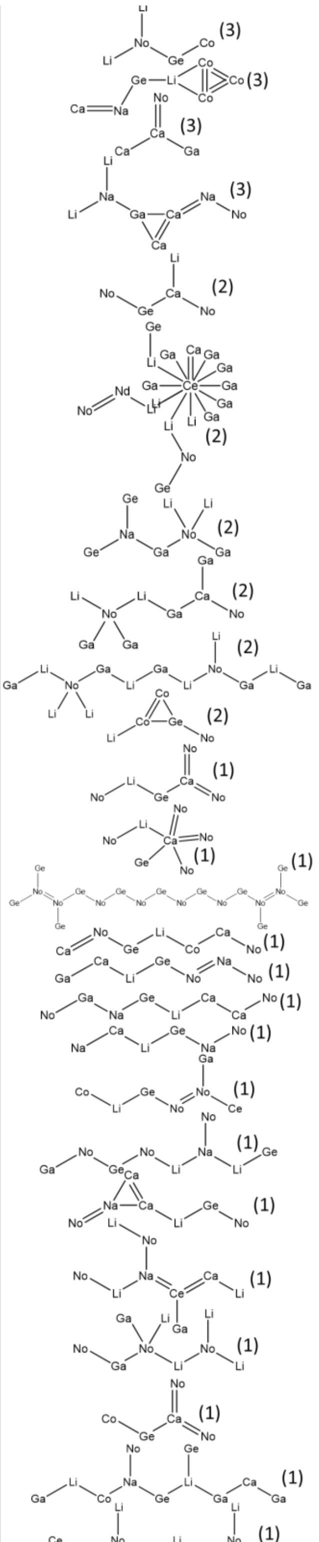
Additional Cores Extracted From The Whole Dataset [9]	Cores Extracted From Both Methods [15]	Additional Cores Extracted From the Clustered Dataset [14]
 (278)  (167)  (163)  (152)  (139)  (64)  (32)  (12)  (3)	 (429)  (226)  (152)  (146)  (135)  (91)  (88)  (81)  (73)  (71)  (60)  (56)  (53)  (52)  (51)	 (183)  (100)  (98)  (86)  (66)  (65)  (62)  (57)  (55)  (54)  (53)  (47)  (3)  (2)

Table 0-13: Table displaying the cores extracted from the two different processes for the P2x7 dataset, the number in the bracket is the number of examples of each core once re-examined

Additional Cores Extracted From The Whole Dataset [23]	Cores Extracted From Both Methods [35]	Additional Cores Extracted From the Clustered Dataset [62]
 (440) (237) (159) (104) (100) (49) (44) (44) (32) (27) (25) (24) (22) (14) (12) (10) (10) (8) (8) (3) (3) (1) (1)	 (724) (602) (373) (324) (303) (276) (264) (234) (144) (94) (84) (75) (75) (52) (50) (27) (27) (26) (23) (21) (18) (15) (14) (10) (5) (2) (2) (1) (1) (1) (1) (1) (1) (1) (1)	 (259) (200) (128) (127) (115) (114) (95) (93) (81) (72) (69) (68) (68) (61) (48) (47) (43) (40) (31) (29) (29) (27) (22) (20) (18) (18) (13) (12) (11) (9) (8) (8) (7) (5) (5) (5) (5)

(3)
(3)
(3)
(3)
(2)
(2)
(2)
(2)
(2)
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

Table 0-14: RG core comparison for each dataset

Dataset	Number of Clusters	RG Core Extraction					
		Whole Dataset			Clustered Dataset		
		Number of RG Cores	Mean Core Size	Number of Singletons	Number of RG Cores	Mean Core Size	Number of Singletons
Bajorath	28	24	5.33	0	29	6.31	0
CDK2	150	114	4.81	35	195	6.25	89
Chk1	3	9	4.67	3	9	5.22	3
Cyto	2	182	3.68	41	181	3.72	16
FactorXa	112	42	4.83	9	126	7.07	31
Neurokinin	2	86	3.67	21	85	3.71	8
P2x7	67	58	5.26	13	97	6.17	16
P2x7 Subset	4	7	5.00	1	6	5.00	1
p38α	150	125	4.66	30	245	5.56	88

Table 0-15: Difference between the fMCS method and the method presented in this chapter on a cluster level and a whole dataset level. Threshold Level set to 1.

Dataset	Unique Number of Cores from			Comparison of RG cores for…						
				RG extraction method and fMCS for clustered data				RG extraction method for whole dataset and fMCS for clustered data		
	Cluster Method	Whole Method	fMCS Method	Number of Clusters that Cores Exactly Match	Number of Cores the Same	Number of Unique Cores to Cluster Method	Number of Unique Cores to fMCS Method	Number of Cores the Same	Number of Unique Cores to Whole Method	Number of Unique Cores to fMCS Method
Bajorath	29	24	28	27	27	2	1	14	10	14
CDK2	195	115	138	117	114	81	24	29	86	109
Chk1	9	9	3	0	1	8	2	1	8	2
Cyto	181	182	0	0	0	181	0	0	182	0
FactorXa	126	42	108	69	92	34	86	16	26	102
Neurokinin	85	86	0	0	0	85	0	0	86	0

P2x7	97	58	58	32	41	56	17	12	46	46
P2x7 Subset	6	7	4	2	2	4	2	1	6	3
p38α	245	125	113	83	92	153	21	31	94	82

Exploration Scores

Table 0-16: Results for each node for this experiment for 1-Prior Prob, Change in Entropy, KL Divergence

Row No.	Prior Distributions (E, F, G)	1-Prior Prob	Change in Entropy	KL Divergence
15	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(1), (1), (1)	(0.145), (0.152), (0.128)	(0.047) (0.047), (0.047)
17	(11, 6, **3**), (14, 3, 3), (7, 6, 4, 3)	(1), (0.85), (1)	(0.145), (0.045), (0.128)	(0.047) (0.006), (0.047)
16	(11, 6, 3), (14, 3, **3**), (7, 6, 4, 3)	(0.85), (1), (1)	(0.038), (0.152), (0.128)	(0.006) (0.047), (0.047)
18	(11, 6, 3), (14, 3, 3), (7, 6, 4, **3**)	(1), (1), (0.85)	(0.145), (0.152), (0.021)	(0.047) (0.047), (0.006)
19	(11, 6, **3**), (14, 3, **3**), (7, 6, 4, **3**)	(0.85), (0.85), (0.85)	(0.038), (0.045), (0.021)	(0.006), (0.006), (0.006)
20	(11, 6, **3**), (14, 3, **3**), (7, 6, **4**, 3)	(0.85), (0.85), (0.80)	(0.038), (0.045), (0.009)	(0.006), (0.006), (0.004)
21	(11, **6**, 3), (14, 3, **3**), (7, 6, 4, **3**)	(0.7), (0.85), (0.85)	(0.008), (0.045), (0.021)	(0.003), (0.006), (0.006)
22	(11, 6, **3**), (14, 3, **3**), (7, **6**, 4, 3)	(0.85), (0.85), (0.7)	(0.038), (0.045), (-0.009)	(0.006), (0.006), (0.003)
23	(11, **6**, 3), (14, 3, **3**), (7, **6**, 4, 3)	(0.7), (0.85), (0.7)	(0.008), (0.045), (-0.009)	(0.003), (0.006), (0.003)
24	(**11**, 6, 3), (14, 3, **3**), (7, 6, 4, **3**)	(0.45), (0.85), (0.85)	(-0.019), (0.045), (0.021)	(0.001), (0.006), (0.006)
25	(11, 6, **3**), (**14**, 3, 3), (7, 6, 4, **3**)	(0.85), (0.3), (0.85)	(0.038), (-0.023), (0.021)	(0.006), (0.0005), (0.006)
26	(**11**, 6, 3), (**14**, 3, 3), (7, 6, 4, **3**)	(0.45), (0.3), (0.85)	(-0.019), (-0.023), (0.021)	(0.001), (0.0005), (0.006)
27	(**11**, 6, 3), (**14**, 3, 3), (**7**, 6, 4, 3)	(0.45), (0.3), (0.65)	(-0.019), (-0.023), (-0.016)	(0.001), (0.0005), (0.002)

Table 0-17: Combined Overall Scores for this experiment for 1-Prior Prob, Change in Entropy and KL Divergence

Row No.:Node No	1-Prior Prob			Change in Entropy			KL Divergence		
	Summed	Multiplied	Mean	Summed	Multiplied	Mean	Summed	Multiplied	Mean
15	3.00	1.00	1.00	0.425	0.003	0.142	0.140	0.0001	0.047
17	2.85	0.850	0.950	0.318	0.0008	0.106	0.099	1.23E-05	0.033
16	2.85	0.850	0.950	0.318	0.0007	0.106	0.099	1.23E-05	0.033
18	2.85	0.850	0.950	0.318	0.0005	0.106	0.099	1.23E-05	0.033
19	2.55	0.614	0.850	0.104	3.57E-05	0.035	0.017	1.83E-07	0.006
20	2.50	0.578	0.833	0.092	1.50E-05	0.031	0.016	1.36E-07	0.005
21	2.40	0.506	0.800	0.074	7.84E-06	0.025	0.014	8.34E-08	0.005
22	2.40	0.506	0.800	0.074	-1.5E-05	0.025	0.014	8.38E-08	0.005
23	2.25	0.417	0.750	0.045	-3.30E-06	0.015	0.011	3.83E-08	0.004

24	2.15	0.325	0.717	0.047	-1.80E-05	0.016	0.012	3.16E-08	0.004
25	2.00	0.217	0.667	0.036	-1.80E-05	0.012	0.012	1.75E-08	0.004
26	1.60	0.115	0.533	-0.021	8.81E-06	-0.007	0.007	3.03E-09	0.002
27	1.40	0.088	0.467	-0.057	-6.60E-06	-0.019	0.004	1.13E-09	0.001

Table 0-18: Results for each node for this experiment for the Collection Model Score Variations

Row No.	Prior Distributions (E, F, G)	E1	E3	E4
15	(11, 6, 3), (14, 3, 3), (7, 6, 4, 3)	(0.300), (0.300), (0.300)	(0.002), (0.003), (0.001)	(0.0002), (0.0002), (0.0002)
17	(11, 6, **3**), (14, 3, 3), (7, 6, 4, 3)	(0.300), (0.103), (0.300)	(0.002), (0.002), (0.001)	(0.0002), (0.0002), (0.0002)
16	(11, 6, 3), (14, 3, **3**), (7, 6, 4, 3)	(0.103), (0.300), (0.300)	(0.001), (0.003), (0.001)	(0.0002), (0.0002), (0.0002)
18	(11, 6, 3), (14, 3, 3), (7, 6, 4, **3**)	(0.300), (0.300), (0.103)	(0.002), (0.003), (0.0006)	(0.0002), (0.0002), (0.0001)
19	(11, 6, **3**), (14, 3, **3**), (7, 6, 4, **3**)	(0.103), (0.103), (0.103)	(0.001), (0.002), (0.0006)	(0.0002), (0.0002), (0.0001)
20	(11, 6, **3**), (14, 3, **3**), (7, 6, **4**, 3)	(0.103), (0.103), (0.072)	(0.001), (0.002), (0.0003)	(0.0002), (0.0002), (0.0001)
21	(11, **6**, 3), (14, 3, **3**), (7, 6, 4, **3**)	(0.035), (0.103), (0.103)	(0.0005), (0.002), (0.0006)	(0.0001), (0.0002), (0.0001)
22	(11, 6, **3**), (14, 3, **3**), (7, **6**, 4, 3)	(0.103), (0.103), (0.035)	(0.001), (0.002), (-0.0002)	(0.0002), (0.0002), (9.76E-05)
23	(11, **6**, 3), (14, 3, **3**), (7, **6**, 4, 3)	(0.035), (0.103), (0.035)	(0.0005), (0.002), (-0.0002)	(0.0001), (0.0002), (9.76E-05)
24	(**11**, 6, 3), (14, 3, **3**), (7, 6, 4, **3**)	(0.006), (0.103), (0.103)	(-0.0008), (0.002), (0.0006)	(4.55E-05), (0.0002), (0.0001)
25	(11, 6, **3**), (**14**, 3, 3), (7, 6, 4, **3**)	(0.103), (0.002), (0.103)	(0.001), (-0.0009), (0.0006)	(0.0002), (1.95E-05), (0.0001)
26	(**11**, 6, 3), (**14**, 3, 3), (7, 6, 4, **3**)	(0.006), (0.002), (0.103)	(-0.0008), (-0.0009), (0.0006)	(4.55E-05), (1.95E-05), (0.0001)
27	(**11**, 6, 3), (**14**, 3, 3), (**7**, 6, 4, 3)	(0.006), (0.002), (0.025)	(-0.0008), (-0.0009), (-0.0005)	(4.55E-05), (1.95E-05), (8.31E-05)

Table 0-19: Combined Overall Scores for this experiment for the Collection Model Score Variations

Row No.	E1			E3			E4		
	Sum	Multiplied	Mean	Sum	Multiplied	Mean	Sum	Multiplied	Mean
15	0.900	0.027	0.300	0.006	7.99E-09	0.002	0.0006	8.39E-12	0.0002
17	0.703	0.009	0.234	0.005	5.50E-09	0.002	0.0006	6.74E-12	0.0002
16	0.703	0.009	0.234	0.005	4.83E-09	0.002	0.0006	6.61E-12	0.0002
18	0.703	0.009	0.234	0.005	3.22E-09	0.002	0.0006	6.41E-12	0.0002
19	0.309	0.001	0.103	0.004	1.34E-09	0.001	0.0005	4.06E-12	0.0002
20	0.278	0.001	0.093	0.003	7.03E-10	0.001	0.0005	3.64E-12	0.0002
21	0.241	0.000	0.080	0.003	5.11E-10	0.001	0.0004	2.96E-12	0.0001
22	0.241	0.000	0.080	0.003	-5.45E-10	0.001	0.0004	2.81E-12	0.0001
23	0.173	0.000	0.058	0.002	-2.07E-10	0.0007	0.0004	2.05E-12	0.0001

24	0.212	6.28E-05	0.071	0.002	-7.85E-10	0.0006	0.0004	1.14E-12	0.0001
25	0.208	2.15E-05	0.069	0.001	-6.36E-10	0.0003	0.0003	4.43E-13	0.0001
26	0.111	1.24E-06	0.037	-0.001	3.72E-10	-0.0004	0.0002	1.25E-13	6.87E-05
27	0.033	2.98E-07	0.011	-0.002	-3.19E-10	-0.0007	0.0001	7.37E-14	4.94E-05

Applying the Scores to Real Molecules

Table 0-20: Node breakdown for Molecule 1

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	1	0.017	0.003	0.300	0.056	0.0001	7.98E-07
Blue	1	0.018	0.003	0.300	0.092	0.0002	8.33E-07
Molecule Score							
Total Summed	2	0.035	0.006	0.600	0.148	0.0003	1.63E-06
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.5	0.009	0.001	0.150	0.037	7.60E-05	4.08E-07

Table 0-21: Node breakdown for Molecule 2

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	0.985	0.009	0.0003	0.050	0.009	0.0001	7.81E-07
Blue	0.506	-0.0004	4.83E-06	0.000	0.000	-8.72E-06	2.72E-07
Molecule Score							
Total Summed	1.491	0.009	0.0003	0.050	0.009	0.0001	1.05E-06
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.373	0.002	6.63E-05	0.013	0.002	3.17E-05	2.63E-07

Table 0-22: Node breakdown for Molecule 3

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	0.982	0.008	0.0002	0.035	0.007	0.0001	7.78E-07
Blue	0.506	-0.0004	4.83E-06	0.000	0.000	-8.72E-05	2.72E-07
Molecule Score							
Total Summed	1.488	0.008	0.0002	0.035	0.007	0.0001	1.05E-06
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.372	0.002	5.63E-05	0.009	0.002	3.14E-05	2.62E-07

Table 0-23: Node breakdown for Molecule 4

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	0.728	0.0006	1.24E-05	0.000	0.000	4.29E-05	4.89E-07
Blue	0.527	-0.0002	5.23E-06	0.000	0.000	-2.04E-06	2.96E-07
Molecule Score							
Total Summed	1.255	0.0004	1.76E-05	0.000	0.000	4.09E-05	7.85E-07
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.314	0.0001	4.41E-06	0.000	0.000	1.02E-05	1.96E-07

Table 0-24: Node breakdown for Molecule 5

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	0.437	-0.002	4E-06	0.000	0.000	-5.18E-05	1.60E-07
Blue	0.527	-0.0002	5.23E-06	0.000	0.000	-2.04E-06	2.96E-07
Molecule Score							
Total Summed	0.964	-0.002	9.23E-06	0.000	0.000	-5.39E-05	4.56E-07
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.241	-0.001	2.31E-06	0.000	0.000	-1.35E-05	1.14E-07

Table 0-25: Node breakdown for Molecule 6

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Pink	0	0	0	0	0	0	0
Red	0	0	0	0	0	0	0
Purple	0.437	-0.002	4E-06	0.000	0.000	-5.18E-05	1.60E-07
Blue	0.506	-0.0004	4.83E-06	0.000	0.000	-8.72E-06	2.72E-07
Molecule Score							
Total Summed	0.943	-0.002	8.83E-06	0.000	0.000	-6.06E-05	4.32E-07
Total Multiplied	0	0	0	0.000	0	0.000	0.000
Total Mean	0.236	-0.001	2.21E-06	0.000	0.000	-1.51E-05	1.08E-07

Cross Core Comparison

Table 0-26: Node breakdown for Molecule 7

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Red	0	0	0	0.001	0.001	0	0
Blue	0.188	-0.017	0.0004	0.003	0.002	-0.0007	1.55E-05
Orange	0.125	-0.015	0.0003	0.002	0.001	-0.0006	6.88E-06
Pink	0	0	0	0.001	0.001	0	0
Green	1	0.202	0.058	0.3	0.200	0.005	0.0004
Molecule Score							
Total Summed	1.313	0.170	0.059	0.307	0.307	0.003	0.0004
Total Multiplied	0	0	0	1.76E-12	1.76E-12	0	0
Total Mean	0.263	0.034	0.012	0.061	0.061	0.0007	8.29E-05

Table 0-27: Node breakdown for Molecule 8

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Orange	0	0	0	5.57E-07	5.6E-07	0	0
Purple	0.081	-0.007	4.54E-05	1.62E-06	5.6E-07	-0.0002	4.51E-07
Blue	0.081	-0.007	4.54E-05	1.62E-06	9.0E-07	-0.0002	4.51E-07
Cyan	0.919	0.049	0.003	0.103	9.0E-07	0.002	7.02E-05
Molecule Score							
Total Summed	1.081	0.035	0.003	0.103	0.103	0.001	7.11E-05
Total Multiplied	0	0	0	1.51E-19	1.51E-19	0	0
Total Mean	0.270	0.009	0.0008	0.026	0.026	0.0004	1.78E-05

Table 0-28: Node breakdown for Molecule 9

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Cyan	0.5	-0.033	0.0007	0.002	0.0005	-0.0008	2.20E-05
Green	0	0	0	1.38E-05	1.4E-05	0	0
Olive	0	0	0	1.38E-05	1.4E-05	0	0
Blue	0.857	0.050	0.003	0.072	0.041	0.002	0.0001
Molecule Score							
Total Summed	1.357	0.017	0.004	0.074	0.074	0.001	0.0001
Total Multiplied	0	0	0	2.79E-14	2.79E-14	0	0
Total Mean	0.339	0.004	0.001	0.019	0.019	0.0003	3.33E-05

Table 0-29: Node breakdown for Molecule 10

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Cyan	0.5	-0.033	0.0007	0.002	0.0005	-0.0008	2.20E-05
Green	0	0	0	1.38E-05	1.4E-05	0	0
Olive	0	0	0	1.38E-05	1.4E-05	0	0
Blue	0.143	-0.009	0.0001	5.75E-05	3.3E-05	-0.0004	3.09E-06
Molecule Score							
Total Summed	0.643	-0.042	0.0008	0.002	0.002	-0.001	2.51E-05
Total Multiplied	0	0	0	2.22E-17	2.23E-17	0	0
Total Mean	0.161	-0.011	0.0002	0.0005	0.0005	-0.0003	6.28E-06

Table 0-30: Node breakdown for Molecule 11

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Orange	0	0	0	5.57E-07	5.6E-07	0	0
Purple	0.081	-0.007	4.54E-05	1.62E-06	9.0E-07	-0.0002	4.51E-07
Blue	0.081	-0.007	4.54E-05	1.62E-06	9.0E-07	-0.0002	4.51E-07
Cyan	0.135	-0.010	7.65E-05	3.31E-06	1.50E-06	-0.0003	1.16E-06
Molecule Score							
Total Summed	0.297	-0.024	0.0002	7.12E-06	7.12E-06	-0.0007	2.06E-06
Total Multiplied	0	0	0	4.86E-24	4.86E-24	0	0
Total Mean	0.074	-0.006	4.18E-05	1.78E-06	1.80E-06	-0.0002	5.15E-07

Table 0-31: Node breakdown for Molecule 12

Node	1-Prior Prob	Change in Entropy	KL Divergence	E1	E2	E3	E4
Orange	0	0	0	0.0001	0.0001	0	0
Green	0	0	0	0.0001	0.0001	0	0
Pink	0	0	0	0.0001	0.0001	0	0
Blue	0.136	-0.011	0.0002	0.0003	0.0002	-0.0005	4.47E-06
Molecule Score							
Total Summed	0.136	-0.011	0.0002	0.0007	0.0007	-0.0005	4.47E-06
Total Multiplied	0	0	0	5.52E-16	5.52E-16	0	0
Total Mean	0.034	-0.003	5.00E-05	0.0002	0.0002	-0.0001	1.12E-06

Molecular Generation

Node Extraction

Table 0-32: Table showing all of the number of substructural fragments extracted for each node type from ChEMBL and Zinc for RG made with linker parameter

Nodes	ChEMBL	Zinc	Comparison of ChEMBL and Zinc nodes (same)
	Linker	Linker	Linker
Acyclic inert - Li	11843	7877	2579
Acyclic HBA - Ga	848	981	372
Acyclic HBD - Gd	20	12	5
Aromatic NHB - No	557	546	419
Aromatic HBA- Na	580	549	425
Aromatic HBD - Nd	234	247	171
Aliphatic HBD - Cd	30	9	0
Aliphatic HBA - Ca	2103161	16862	6672
Aliphatic NHB - Co	4565603	11879	4565
Acyclic HBA HBD - Ge	2556	2663	990
Aromatic HBA HBD - Ne	200	204	137
Aliphatic HBA HBD - Ce	2282292	11249	2680
Hydrophobic - Hg	0	0	0

Table 0-33: Table showing all of the number of substructural fragments extracted for each node type for each dataset for RG made with linker parameter

Nodes	Bajorath	CDK2	Chk1	Cyto	FactorXa	MMP12	Neurokinin	P2x7	P2x7 Subset	P38α
Acyclic inert - Li	87	83	22	328	95	23	53	51	23	89
Acyclic HBA - Ga	18	27	10	76	19	8	20	15	6	22
Acyclic HBD - Gd	0	0	0	0	0	0	1	1	0	0
Aromatic NHB - No	29	51	12	93	43	12	26	22	12	70
Aromatic HBA- Na	81	77	8	101	57	9	40	62	42	107
Aromatic HBD - Nd	4	22	3	32	9	1	5	4	2	17
Aliphatic HBD - Cd	0	0	0	1	0	0	0	0	0	0
Aliphatic HBA - Ca	91	107	12	1369	125	5	128	83	34	125
Aliphatic NHB - Co	27	53	8	793	35	2	57	47	6	63
Acyclic HBA HBD - Ge	20	34	12	101	62	8	28	17	5	27
Aromatic HBA HBD - Ne	12	16	6	19	11	1	15	15	1	23

Aliphatic HBA HBD - Ce	29	502	9	204	38	0	58	26	3	58
Hydrophobic - Hg	0	0	0	0	0	0	0	0	0	0

BMS Filter Breakdown

Single Node Alterations

Dataset

('halogen_heteroatom > 0', 15), ('aryl_phosphonate > 0', 4), ('contains_metal > 0', 2), ('sulfonium > 0', 2), ('azo_filter2 > 0', 1), ('hydrazine > 0', 1)

ChEMBL

('boron_warhead > 0', 41), ('azo_filter2 > 0', 8), ('CH2_S#O_3_ring > 0', 1), ('peroxide > 0', 42), ('aziridine_diazirine > 0', 13), ('crown_ether > 0', 54), ('phosphite > 0', 5), ('hydrazine > 0', 21), ('sulfonium > 0', 58), ('sulf_D3_nitrogen > 0', 4), ('activated_S#O_3_ring > 0', 16), ('polysulfide > 0', 6), ('sulf_D2_nitrogen > 0', 14), ('contains_metal > 0', 59), ('hyperval_sulfur > 0', 8), ('halogen_heteroatom > 0', 172), ('quat_N_N > 0', 57), ('oxonium > 0', 13), ('disulfide_acyclic > 0', 9), ('azo_amino > 0', 25), ('sulf_D2_oxygen_D2 > 0', 2), ('alpha_dicarbonyl > 0', 61), ('keto_def_heterocycle > 0', 30), ('activated_diazo > 0', 3), ('diazo_carbonyl > 0', 4), ('gte_3_COOH > 0', 1), ('hydroxamate_warhead > 0', 2), ('phosphorus_sulfur_bond > 0', 19), ('thio_hydroxamate > 0', 13), ('aryl_thiocarbonyl > 0', 21), ('anhydride > 0', 5), ('nitrone > 0', 49), ('bad_boron > 0', 2), ('azo_filter3 > 0', 2), ('thioester > 0', 2), ('acyclic_imide > 0', 10), ('diamino_sulfide > 0', 2), ('hydrazothiourea > 0', 4), ('hetero_silyl > 0', 3), ('quat_N_acyl > 0', 63), ('phosphorous_nitrogen_bond > 0', 18), ('gte_2_free_phos > 0', 4), ('aryl_phosphonate > 0', 118), ('NO_phosphonate > 0', 6), ('aldehyde > 0', 6), ('nitrosamine > 0', 15), ('oxime > 0', 41), ('sulfonyl_heteroatom > 0', 8), ('bad_cations > 0', 8), ('polyene > 0', 356), ('gte_10_carbon_sb_chain > 0', 2249), ('non_ring_acetal > 0', 4), ('trisub_bis_act_olefin > 0', 26), ('activated_acetylene > 0', 142), ('phosphonium > 0', 17), ('thiocarbonate > 0', 28), ('non_ring_CH2O_acetal > 0', 30), ('acrylate > 0', 137), ('gte_2_N_quats > 0', 42), ('gte_8_CF2_or_CH2 > 0', 49), ('primary_halide_sulfate > 0', 7), ('4halo_pyridine_3EWG > 0', 1), ('2halo_pyridine_3EWG > 0', 1), ('thiopyrylium > 0', 3), ('2halo_pyridine_5EWG > 0', 1), ('secondary_halide_sulfate > 0', 14)

Zinc

('boron_warhead > 0', 15), ('CH2_S#O_3_ring > 0', 2), ('aziridine_diazirine > 0', 11), ('crown_ether > 0', 60), ('phosphite > 0', 10), ('hydrazine > 0', 25), ('thio_hydroxamate > 0', 19), ('sulf_D3_nitrogen > 0', 8), ('activated_S#O_3_ring > 0', 10), ('sulf_D2_nitrogen > 0', 17), ('contains_metal > 0', 10), ('halogen_heteroatom > 0', 179), ('quat_N_N > 0', 59), ('oxonium > 0', 22), ('hyperval_sulfur > 0', 16), ('alpha_dicarbonyl > 0', 52), ('diazo_carbonyl > 0', 11), ('keto_def_heterocycle > 0', 45), ('gte_3_COOH > 0', 5), ('acyl_cyanide > 0', 2), ('activated_diazo > 0', 1), ('hydroxamate_warhead > 0', 4), ('anhydride > 0', 17), ('aryl_thiocarbonyl > 0', 10), ('nitrone > 0', 22), ('azo_filter3 > 0', 2), ('thioester > 0', 2), ('acyclic_imide > 0', 9), ('hydrazothiourea > 0', 2), ('diamino_sulfide > 0', 4), ('thiocarbonate > 0', 3), ('phosphorous_nitrogen_bond > 0', 17), ('sulfonium > 0', 11), ('gte_2_free_phos > 0', 3), ('aryl_phosphonate > 0', 33), ('NO_phosphonate > 0', 1), ('azo_filter2 > 0', 9), ('aldehyde > 0', 6), ('nitrosamine > 0', 3), ('oxime > 0', 62), ('sulfonyl_heteroatom > 0', 16), ('azo_amino > 0', 16), ('polyene > 0', 385), ('phosphonium > 0', 20), ('non_ring_CH2O_acetal > 0', 32), ('acrylate > 0', 64), ('gte_2_N_quats > 0', 45), ('bad_boron > 0', 4), ('trisub_bis_act_olefin > 0', 17), ('activated_acetylene > 0', 113), ('gte_8_CF2_or_CH2 > 0', 37), ('quat_N_acyl > 0', 53), ('gte_10_carbon_sb_chain > 0', 1065), ('non_ring_acetal > 0', 18), ('bad_cations > 0', 1), ('primary_halide_sulfate > 0', 9), ('4halo_pyridine_3EWG > 0', 1), ('thiopyrylium > 0', 15), ('2halo_pyridine_3EWG > 0', 1), ('2halo_pyridine_5EWG > 0', 1), ('gte_7_total_hal > 0', 1), ('gte_3_iodine > 0', 1), ('secondary_halide_sulfate > 0', 4)

Multiple Node Alterations

Dataset

('halogen_heteroatom > 0', 990), ('acyl_imidazole > 0', 120), ('quat_N_acyl > 0', 105), ('aryl_phosphonate > 0', 52), ('keto_def_heterocycle > 0', 39), ('contains_metal > 0', 26), ('azide > 0', 19), ('primary_halide_sulfate > 0', 17), ('aldehyde > 0', 17), ('azo_filter2 > 0', 13), ('hydrazine > 0', 13), ('sulfonium > 0', 10), ('quat_N_N > 0', 1)

ChEMBL

('keto_def_heterocycle > 0', 7460), ('azo_filter2 > 0', 1692), ('carbodiimide_iso#thio#cyanate > 0', 21680), ('anhydride > 0', 136283), ('nitrone > 0', 27508), ('nitrosamine > 0', 114947), ('sulfite_sulfate_ester > 0', 28863), ('sulfonyl_anhydride > 0', 32983), ('disulfide_acyclic > 0', 160261), ('hydrazine > 0', 11983), ('boron_warhead > 0', 44474), ('sulfonium > 0', 40612), ('thiosulfoxide > 0', 32736), ('alpha_dicarbonyl > 0', 159208), ('hetero_silyl > 0', 9094), ('thioester > 0', 56207), ('activated_diazo > 0', 21674), ('contains_metal > 0', 396498), ('sulf_D2_oxygen_D2 > 0', 33158), ('CH2_S#O_3_ring > 0', 294), ('azo_amino > 0', 23744), ('diamino_sulfide > 0', 10757), ('thiocarbonate > 0', 49834), ('aldehyde > 0', 22383), ('peroxide > 0', 117147), ('thio_xanthate > 0', 8343), ('crown_ether > 0', 24354), ('gte_2_N_quats > 0', 11694), ('aryl_thiocarbonyl > 0', 44070), ('aziridine_diazirine > 0', 3406), ('diazo_carbonyl > 0', 128181), ('polysulfide > 0', 1062), ('sulf_D2_nitrogen > 0', 45664), ('phosphite > 0', 5035), ('aryl_phosphonate > 0', 16655), ('sulf_D3_nitrogen > 0', 8890), ('activated_S#O_3_ring > 0', 4704), ('bad_boron > 0', 13127), ('oxonium > 0', 16516), ('alpha_halo_amine > 0', 4138), ('secondary_halide_sulfate > 0', 80612), ('bad_cations > 0', 15332), ('halogen_heteroatom > 0', 369674), ('hyperval_sulfur > 0', 23076), ('polyene > 0', 328267), ('gte_8_CF2_or_CH2 > 0', 258014), ('quat_N_N > 0', 227350), ('gte_10_carbon_sb_chain > 0', 3355292), ('thio_hydroxamate > 0', 14266), ('quat_N_acyl > 0', 38274), ('hydroxamate_warhead > 0', 1358), ('gte_3_COOH > 0', 49), ('acyl_pyrazole > 0', 3230), ('oxime > 0', 3607), ('sulfonyl_heteroatom > 0', 232), ('phosphorus_sulfur_bond > 0', 2165), ('acyl_imidazole > 0', 3230), ('azo_filter3 > 0', 356), ('gte_2_free_phos > 0', 6664), ('hydrazothiourea > 0', 336), ('NO_phosphonate > 0', 728), ('acyclic_imide > 0', 540), ('phosphorous_nitrogen_bond > 0', 105984), ('diazonium > 0', 54), ('azide > 0', 728), ('acyl_activated_NO > 0', 11650), ('acyl_cyanide > 0', 256), ('activated_acetylene > 0', 41748), ('acrylate > 0', 40278), ('trisub_bis_act_olefin > 0', 3925), ('phosphonium > 0', 3859), ('non_ring_acetal > 0', 760), ('non_ring_CH2O_acetal > 0', 5710), ('alpha_halo_heteroatom > 0', 44), ('primary_halide_sulfate > 0', 3525), ('phosphorane > 0', 131), ('phosphorus_phosphorus_bond > 0', 131), ('2halo_pyridine_5EWG > 0', 572), ('2halo_pyridine_3EWG > 0', 572), ('4halo_pyridine_3EWG > 0', 572), ('thiopyrylium > 0', 1155), ('betalactam > 0', 116100), ('gte_7_total_hal > 0', 6), ('alpha_halo_carbonyl > 0', 3876), ('activated_vinyl_ester > 0', 38760), ('halo_olefin_bis_EWG > 0', 2854), ('beta_lactone > 0', 34830), ('gte_3_iodine > 0', 6), ('carbonyl_halide > 0', 7740), ('isonitrile > 0', 245)

Zinc

('keto_def_heterocycle > 0', 9468), ('anhydride > 0', 108104), ('phosphonium > 0', 10412), ('nitrone > 0', 40213), ('nitrosamine > 0', 43798), ('sulfite_sulfate_ester > 0', 39955), ('sulfonyl_anhydride > 0', 17175), ('disulfide_acyclic > 0', 7134), ('thio_hydroxamate > 0', 19878), ('hydrazine > 0', 21730), ('boron_warhead > 0', 3911), ('alpha_dicarbonyl > 0', 90509), ('sulfonyl_heteroatom > 0', 44684), ('thioester > 0', 72495), ('activated_diazo > 0', 36186), ('contains_metal > 0', 56492), ('sulf_D2_oxygen_D2 > 0', 22732), ('CH2_S#O_3_ring > 0', 586), ('azo_filter3 > 0', 14514), ('diamino_sulfide > 0', 14484), ('thiocarbonate > 0', 22742), ('crown_ether > 0', 21354), ('thio_xanthate > 0', 17010), ('gte_2_N_quats > 0', 16378), ('aryl_thiocarbonyl > 0', 22144), ('aziridine_diazirine > 0', 2827), ('hydrazothiourea > 0', 6410), ('diazo_carbonyl > 0', 64972), ('sulf_D2_nitrogen > 0', 23481), ('phosphite > 0', 19606), ('aryl_phosphonate > 0', 23064), ('sulf_D3_nitrogen > 0', 1584), ('hyperval_sulfur > 0', 71470), ('activated_S#O_3_ring > 0', 2930), ('phosphorous_nitrogen_bond > 0', 45826), ('bad_boron > 0', 24256), ('gte_7_total_hal > 0', 6701), ('oxonium > 0', 18921), ('alpha_halo_amine > 0', 2385), ('secondary_halide_sulfate > 0', 23868), ('halogen_heteroatom > 0', 241112), ('gte_8_CF2_or_CH2 > 0', 144631), ('polyene > 0', 215666), ('quat_N_N > 0', 149183), ('gte_10_carbon_sb_chain > 0', 1293677), ('gte_3_iodine > 0', 6701), ('azo_filter2 > 0', 1105), ('quat_N_acyl > 0', 66853), ('hydroxamate_warhead > 0', 969), ('gte_3_COOH > 0', 205), ('acyl_pyrazole > 0', 3318), ('sulfonium > 0', 2543), ('oxime > 0', 5404), ('acyl_imidazole > 0', 3318), ('azo_amino > 0', 1554), ('aldehyde > 0', 859), ('gte_2_free_phos > 0', 1891), ('NO_phosphonate > 0', 46), ('acyclic_imide > 0', 414), ('acyl_cyanide > 0', 592), ('thiopyrylium > 0', 7167), ('carbodiimide_iso#thio#cyanate > 0', 138), ('diazonium > 0', 46), ('azide > 0', 638), ('acyl_activated_NO > 0', 13052), ('activated_acetylene > 0', 33109), ('bad_cations > 0', 482), ('acrylate > 0', 18752), ('trisub_bis_act_olefin > 0', 2839), ('non_ring_acetal > 0', 3744), ('non_ring_CH2O_acetal > 0', 6676), ('primary_halide_sulfate > 0', 3040), ('2halo_pyridine_5EWG > 0', 503), ('2halo_pyridine_3EWG > 0', 503), ('peroxide > 0', 288), ('4halo_pyridine_3EWG > 0', 503), ('halo_imino > 0', 221), ('gte_4_basic_N > 0', 442), ('betalactam > 0', 45318), ('activated_vinyl_ester > 0', 30296), ('beta_lactone > 0', 16185), ('acyl_123_triazole > 0', 5410), ('isonitrile > 0', 44)

Bibliography

Abdi, H. (2010). Partial least squares regression and projection on latent structure regression (PLS Regression). *WIREs Computational Statistics*, *2*(1), 97–106. https://doi.org/10.1002/wics.51

Agrafiotis, D. K., Bandyopadhyay, D., & Farnum, M. (2007). Radial Clustergrams: Visualizing the Aggregate Properties of Hierarchical Clusters. *Journal of Chemical Information and Modeling*, *47*(1), 69–75. https://doi.org/10.1021/ci600427x

Agrafiotis, D. K., Shemanarev, M., Connolly, P. J., Farnum, M., & Lobanov, V. S. (2007). SAR Maps: A New SAR Visualization Technique for Medicinal Chemists. *J. Med. Chem.*, *50*(24), 5926–5937. https://doi.org/10.1021/jm070845m

Agrawal, K. P., Garg, S., & Patel, P. (2015). Performance Measures for Densed and Arbitrary Shaped Clusters. *International Journal of Computer Science & Communication*, *6*, 338–350. https://doi.org/10.090592/IJCSC.2015.637

Alvarsson, J., Lampa, S., Schaal, W., Andersson, C., Wikberg, J. E. S., & Spjuth, O. (2016). Large-scale ligand-based predictive modelling using support vector machines. *Journal of Cheminformatics*, *8*(1), 39. https://doi.org/10.1186/s13321-016-0151-5

Arif, S. M., Holliday, J. D., & Willett, P. (2009). Analysis and use of fragment-occurrence data in similarity-based virtual screening. *Journal of Computer-Aided Molecular Design*, *23*(9), 655–668. https://doi.org/10.1007/s10822-009-9285-0

Arif, S. M., Holliday, J. D., & Willett, P. (2010). Inverse Frequency Weighting of Fragments for Similarity-Based Virtual Screening. *Journal of Chemical Information and Modeling*, *50*(8), 1340–1349. https://doi.org/10.1021/ci1001235

Arús-Pous, J., Patronov, A., Bjerrum, J., Tyrchan, C., Reymond, J.-L., Chen, H., & Engkvist, O. (2020). SMILES-Based Deep Generative Scaffold Decorator for De-Novo Drug Design. *ChemRxiv.* https://doi.org/10.26434/chemrxiv.11638383.v1

Auman, J. T., Boorman, G. A., Wilson, R. E., Travlos, G. S., & Paules, R. S. (2007). Heat map visualization of high-density clinical chemistry data. *Physiological Genomics*, *31*(2), 352–356. https://doi.org/10.1152/physiolgenomics.00276.2006

Azuaje, F., & Bolshakova, N. (2002). Chapter13 Clustering Genomic Expression Data: Design and Evaluation Principles. In *Understanding and Using Microarray Analysis Techniques: A Practical Guide*. Springer.

Ball, G. H., & Hall, D. J. (1965). *Isodata: A Novel Method Of Data Analysis And Pattern Classification*.

Barker, E. J., Gardiner, E. J., Gillet, V. J., Kitts, P., & Morris, J. (2003). Further Development of Reduced Graphs for Identifying Bioactive Compounds. *Journal of Chemical Information and Computer Sciences*, *43*(2), 346–356. https://doi.org/10.1021/ci0255937

Barrow, H. G., & Burstall, R. M. (1975). Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques. *Information Processing Letters*, *4*, 83–84.

Bessonov, Y. (1985). On the solution of a problem on the search for the best intersection of graphs on the basis of an analysis of the projections of the subgraphs of the modular product. *Vychisl. Sistemy*, *121*, 34.

Birchall, K., & Gillet, V. J. (2010). Reduced Graphs and Their Applications in Chemoinformatics. In *Chemoinformatics and Computational Chemical Biology* (pp. 197–212). Humana Press, Totowa, NJ. https://doi.org/10.1007/978-1-60761-839-3_8

Birchall, K., Gillet, V. J., Harper, G., & Pickett, S. D. (2006). Training Similarity Measures for Specific Activities: Application to Reduced Graphs. *Journal of Chemical Information and Modeling*, *46*(2), 577–586. https://doi.org/10.1021/ci050465e

Birchall, K., Gillet, V. J., Harper, G., & Pickett, S. D. (2008). Evolving Interpretable Structure−Activity Relationships. 1. Reduced Graph Queries. *Journal of Chemical Information and Modeling*, *48*(8), 1543–1557. https://doi.org/10.1021/ci8000502

Bohacek, R. S., & McMartin, C. (1994). Multiple Highly Diverse Structures Complementary to Enzyme Binding Sites: Results of Extensive Application of a de Novo Design Method Incorporating Combinatorial Growth. *Journal of the American Chemical Society*, *116*(13), 5560–5571. https://doi.org/10.1021/ja00092a006

Bora, D. J., & Gupta, A. K. (2014). A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm. *International Journal of Computer Trends and Technology*, *10*(2).

Bosc, N., Atkinson, F., Felix, E., Gaulton, A., Hersey, A., & Leach, A. R. (2019). Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. *Journal of Cheminformatics*, *11*(1), 4. https://doi.org/10.1186/s13321-018-0325-4

Brecher, J. (2008). Graphical representation standards for chemical structure diagrams (IUPAC Recommendations 2008). *Pure and Applied Chemistry*, *80*(2), 277–410. https://doi.org/10.1351/pac200880020277

Bro, R., & Smilde, A. K. (2014). Principal component analysis. *Anal. Methods*, *6*(9), 2812–2831. https://doi.org/10.1039/C3AY41907J

Brown, N., Ertl, P., Lewis, R., Luksch, T., Reker, D., & Schneider, N. (2020). Artificial intelligence in chemistry and drug design. *Journal of Computer-Aided Molecular Design*, *34*(7), 709–715. https://doi.org/10.1007/s10822-020-00317-x

Brown, R. D., & Martin, Y. C. (1996). Use of Structure−Activity Data To Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *Journal of Chemical Information and Computer Sciences*, *36*(3), 572–584. https://doi.org/10.1021/ci9501047

Bunin, B., Siesel, B., Morales, G., & Bajorath, J. (2007). *Chemoinformatics*. Springer Netherlands. https://doi.org/10.1007/1-4020-5001-1

Burbidge, R., Trotter, M., Buxton, B., & Holden, S. (2001). Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers & Chemistry*, *26*(1), 5–14. https://doi.org/10.1016/S0097-8485(01)00094-8

Butina, D. (1999). Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *Journal of Chemical Information and Computer Sciences*, *39*(4), 747–750. https://doi.org/10.1021/ci9803381

Calinski, T., & Harabasz, J. (1974). A Dendrite Method for Cluster Analysis. *Communications in Statistics - Theory and Methods*, *3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Chen, B., Zhang, T., Bond, T., & Gan, Y. (2015). Development of quantitative structure activity relationship (QSAR) model for disinfection byproduct (DBP) research: A review of methods and resources. *Journal of Hazardous Materials*, *299*, 260–279. https://doi.org/10.1016/j.jhazmat.2015.06.054

Chen, H., Carlsson, L., Eriksson, M., Varkonyi, P., Norinder, U., & Nilsson, I. (2013). Beyond the Scope of Free-Wilson Analysis: Building Interpretable QSAR Models with Machine Learning Algorithms. *Journal of Chemical Information and Modeling*, *53*(6), 1324–1336. https://doi.org/10.1021/ci4001376

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., … Tropsha, A. (2014). QSAR Modeling: Where Have You Been? Where Are You Going To? *Journal of Medicinal Chemistry*, *57*(12), 4977–5010. https://doi.org/10.1021/jm4004285

Chowdary, N. S., Sri, D., Prasanna, L., Sudhakar, P., & Sarathi, S. (2014). *International Journal of Computer Science and Mobile Computing Evaluating and Analyzing Clusters in Data Mining using Different Algorithms*. *International Journal of Computer Science and Mobile Computing* (Vol. 3).

Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., & Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer*, *8*(1), 37–49. https://doi.org/10.1038/nrc2294

Dalby, A., Nourse, J. G., Hounshell, W. D., Gushurst, A. K. I., Grier, D. L., Leland, B. A., & Laufer, J. (1992). Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *Journal of Chemical Information and Computer Sciences*, *32*(3), 244–255. https://doi.org/10.1021/ci00007a012

Davies, D., & Bouldin, D. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(2), 224–227.

Dearden, J. C., Cronin, M. T. D., & Kaiser, K. L. E. (2009). How not to develop a quantitative structure–activity or structure–property relationship (QSAR/QSPR). *SAR and QSAR in Environmental Research*, *20*(3–4), 241–266. https://doi.org/10.1080/10629360902949567

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., & Rarey, M. (2008). On the Art of Compiling and Using "Drug-Like" Chemical Fragment Spaces. *ChemMedChem*, *3*(10), 1503–1507. https://doi.org/10.1002/cmdc.200800178

Dossetter, A. G., Griffen, E. J., & Leach, A. G. (2013). Matched Molecular Pair Analysis in drug discovery. *Drug Discovery Today*, *18*(15–16), 724–731. https://doi.org/10.1016/j.drudis.2013.03.003

Dragon. (n.d.). Talete srl, Milano, Italy.

Duesbury, E., Holliday, J. D., & Willett, P. (2017). Maximum Common Subgraph Isomorphism Algorithms: A Review. *MATCH Communications in Mathematical and in Computer Chemistry*, *77*(2), 213–232.

Dunn, J. C. (1974). Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, *4*(1), 95–104. https://doi.org/10.1080/01969727408546059

Gardiner, E. J., Gillet, V. J., Willett, P., & Cosgrove, D. A. (2007). Representing Clusters Using a Maximum Common Edge Substructure Algorithm Applied to Reduced Graphs and Molecular Graphs. *Journal of Chemical Information and Modeling*, *47*(2), 354–366. https://doi.org/10.1021/ci600444g

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., … Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(D1), D1100–D1107. https://doi.org/10.1093/nar/gkr777

Gillet, V. J., Downs, G. M., Ling, A., Lynch, M. F., Venkataram, P., & Wood, J. V. (1987). Computer Storage and Retrieval of Generic Chemical Structures in Patents. 8. Reduced Chemical Graphs and Their Applications in Generic Chemical Structure Retrievalt. *J. Chem. Inf. Comput. Sci.*, *27*(306), 126–137.

Gillet, V. J., Holliday, J. D., & Willett, P. (2015). Chemoinformatics at the University of Sheffield 2002-2014. *Molecular Informatics*, *34*(9), 598–607. https://doi.org/10.1002/minf.201500004

Gillet, V. J., Willett, P., & Bradshaw, J. (2003). Similarity Searching Using Reduced Graphs. *Journal of Chemical Information and Computer Sciences*, *43*(2), 338–345. https://doi.org/10.1021/ci025592e

Griffen, E., Leach, A. G., Robb, G. R., & Warner, D. J. (2011). Matched Molecular Pairs as a Medicinal Chemistry Tool. *Journal of Medicinal Chemistry*, *54*(22), 7739–7750. https://doi.org/10.1021/jm200452d

Gunera, J., & Kolb, P. (2015). Fragment-based similarity searching with infinite color space. *Journal of Computational Chemistry*, *36*(21), 1597–1608. https://doi.org/10.1002/jcc.23974

Gupta-Ostermann, D., Hu, Y., & Bajorath, J. (2012). Introducing the LASSO Graph for Compound Data Set Representation and Structure–Activity Relationship Analysis. *Journal of Medicinal Chemistry*, *55*(11), 5546–5553. https://doi.org/10.1021/jm3004762

Gupta, A., & Zhou, H.-X. (2021). Machine Learning-Enabled Pipeline for Large-Scale Virtual Drug Screening. *Journal of Chemical Information and Modeling*, (Mm), acs.jcim.1c00710. https://doi.org/10.1021/acs.jcim.1c00710

Gütlein, M., Karwath, A., & Kramer, S. (2012). CheS-Mapper - Chemical Space Mapping and Visualization in 3D. *Journal of Cheminformatics*, *4*(1), 7. https://doi.org/10.1186/1758-2946-4-7

Halkidi, M. (2001). *On Clustering Validation Techniques*. *Journal of Intelligent Information Systems* (Vol. 17).

Hann, M., & Green, R. (1999). Chemoinformatics — a new name for an old problem? *Current Opinion in Chemical Biology*, *3*(4), 379–383. https://doi.org/10.1016/S1367-5931(99)80057-X

Hansch, C., & Fujita, T. (1963). *ρ-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure*. *J. Biol. Chem* (Vol. 39).

Hansch, C., Maloney, P. P., Fujita, T., & Muir, R. M. (1962). Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. *Nature*, *194*(4824), 178–180. https://doi.org/10.1038/194178b0

Harper, G., Bravi, G. S., Pickett, S. D., Hussain, J., & Green, D. V. S. (2004). The Reduced Graph Descriptor in Virtual Screening and Data-Driven Clustering of High-Throughput Screening Data. *Journal of Chemical Information and Computer Sciences*, *44*(6), 2145–2156. https://doi.org/10.1021/ci049860f

Harper, G., Pickett, S., & Green, D. (2004). (Research Papers) Design of a Compound Screening Collection for use in High Throughput Screening. *Combinatorial Chemistry & High Throughput Screening*, *7*(1), 63–70. https://doi.org/10.2174/138620704772884832

Hasan, S., Bonde, B. K., Buchan, N. S., & Hall, M. D. (2012). Network analysis has diverse roles in drug discovery. *Drug*

Discovery Today, *17*(15–16), 869–874. https://doi.org/10.1016/j.drudis.2012.05.006

Heller, S. R., McNaught, A., Pletnev, I., Stein, S., & Tchekhovskoi, D. (2015). InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, *7*(1), 23. https://doi.org/10.1186/s13321-015-0068-4

Ho, T. K. (1995). Random Decision Forests. Retrieved from https://dl.acm.org/doi/10.5555/844379.844681

Ho, T. K. (1998). The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844. https://doi.org/10.1109/34.709601

Hu, Y., Zhang, B., Vogt, M., & Bajorath, J. (2015). AnalogExplorer2 – Stereochemistry sensitive graphical analysis of large analog series. *F1000Research*, *4*, 1031. https://doi.org/10.12688/f1000research.7146.1

Hussain, J., & Rea, C. (2010). Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets. *Journal of Chemical Information and Modeling*, *50*(3), 339–348. https://doi.org/10.1021/ci900450m

Irwin, J. J., & Shoichet, B. K. (2005). ZINC--a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, *45*(1), 177–182. https://doi.org/10.1021/ci049714+

Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., … Sayle, R. A. (2020). ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *Journal of Chemical Information and Modeling*, *60*(12), 6065–6073. https://doi.org/10.1021/acs.jcim.0c00675

Ivanciuc, O. (2007). Applications of Support Vector Machines in Chemistry. In *Reviews in Computational Chemistry* (Vol. 23, pp. 291–400). https://doi.org/10.1002/9780470116449.ch6

Ivanenkov, Y. A., Savchuk, N. P., Ekins, S., & Balakin, K. V. (2009). Computational mapping tools for drug discovery. *Drug Discovery Today*, *14*(15–16), 767–775. https://doi.org/10.1016/j.drudis.2009.05.016

James, C. A., Weininger, D., & Delany, J. (2020). Daylight Theory Manual. Retrieved September 23, 2020, from https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

Jauhiainen, S., & Kärkkäinen, T. (2017). *A Simple Cluster Validation Index with Maximal Coverage*.

Java Universal Network/Graph Framework. (2020). Retrieved December 18, 2020, from http://jung.sourceforge.net/

Jin, W., Barzilay, R., & Jaakkola, T. (2018). Junction Tree Variational Autoencoder for Molecular Graph Generation.

Juneau, P. (2015). Quantification of Heat Map Data Displays for High-Throughput Analysis. *Journal of Pharmacogenomics & Pharmacoproteomics*, *06*(02), 1–7. https://doi.org/10.4172/2153-0645.1000146

Kayastha, S., Kunimoto, R., Horvath, D., Varnek, A., & Bajorath, J. (2017). From bird's eye views to molecular communities: two-layered visualization of structure–activity relationships in large compound data sets. *Journal of Computer-Aided Molecular Design*, *31*(11), 961–977. https://doi.org/10.1007/s10822-017-0070-1

Kelley, L. A., Gardner, S. P., & Sutcliffe, M. J. (1996). *An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally related subfamilies*. *Protein Engineering* (Vol. 9).

Kenny, P. W., & Sadowski, J. (2005). Structure Modification in Chemical Databases. In T. I. Oprea (Ed.), *Chemoinformatics in Drug Discovery* (pp. 271–285). Wiley-VCH. https://doi.org/10.1002/3527603743.ch11

Kruger, F., Stiefl, N., & Landrum, G. A. (2020). rdScaffoldNetwork: The Scaffold Network Implementation in RDKit. *Journal of Chemical Information and Modeling*, *60*(7), 3331–3335. https://doi.org/10.1021/acs.jcim.0c00296

Kuźmin, V. E., Artemenko, A. G., & Muratov, E. N. (2008). Hierarchical QSAR technology based on the Simplex representation of molecular structure. *Journal of Computer-Aided Molecular Design*, *22*(6–7), 403–421. https://doi.org/10.1007/s10822-008-9179-6

Kuźmin, Victor E., Artemenko, A. G., Polischuk, P. G., Muratov, E. N., Hromov, A. I., Liahovskiy, A. V., … Makan, S. Y. (2005). Hierarchic system of QSAR models (1D-4D) on the base of simplex representation of molecular structure. *Journal of Molecular Modeling*, *11*(6), 457–467. https://doi.org/10.1007/s00894-005-0237-x

Langevin, M., Minoux, H., Levesque, M., & Bianciotto, M. (2020). Scaffold-Constrained Molecular Generation. *Journal of Chemical Information and Modeling*, *60*(12), 5637–5646. https://doi.org/10.1021/acs.jcim.0c01015

Leach, A. G., Jones, H. D., Cosgrove, D. A., Kenny, P. W., Ruston, L., MacFaul, P., … Law, B. (2006). Matched Molecular Pairs as a Guide in the Optimization of Pharmaceutical Properties; a Study of Aqueous Solubility, Plasma Protein Binding and Oral Exposure. *Journal of Medicinal Chemistry*, *49*(23), 6672–6682. https://doi.org/10.1021/jm0605233

Leach, A. R., & Gillet, V. J. (2007). *An Introduction to Chemoinformatics*. Springer.

Lewell, X. Q., Judd, D. B., Watson, S. P., & Hann, M. M. (1998). *RECAP - Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry*.

Lim, J., Hwang, S.-Y., Moon, S., Kim, S., & Kim, W. Y. (2020). Scaffold-based molecular design with a graph generative model. *Chemical Science*, *11*(4), 1153–1164. https://doi.org/10.1039/C9SC04503A

Lounkine, E., Wawer, M., Wassermann, A. M., & Bajorath, J. (2010). SARANEA: A Freely Available Program To Mine Structure−Activity and Structure−Selectivity Relationship Information in Compound Data Sets. *Journal of Chemical Information and Modeling*, *50*(1), 68–78. https://doi.org/10.1021/ci900416a

Lu, J., & Carlson, H. A. (2016). ChemTreeMap: an interactive map of biochemical similarity in molecular datasets. *Bioinformatics*, *32*(23), 3584–3592. https://doi.org/10.1093/bioinformatics/btw523

Maggiora, G. M., & Bajorath, J. (2014). Chemical space networks: A powerful new paradigm for the description of chemical space. *Journal of Computer-Aided Molecular Design*, *28*(8), 795–802. https://doi.org/10.1007/s10822-014-9760-0

Maggiora, G. M., & Shanmugasundaram, V. (2004). Molecular Similarity Measures. In J. Bajorath (Ed.), *Chemoinformatics: Concepts, Methods, and Tools for Drug Discovery* (pp. 1–50). Totowa, NJ: Humana Press. https://doi.org/10.1385/1-59259-802-1:001

Maggiora, G., Vogt, M., Stumpfe, D., & Bajorath, J. (2014). Molecular Similarity in Medicinal Chemistry. *Journal of Medicinal Chemistry*, *57*(8), 3186–3204. https://doi.org/10.1021/jm401411z

Makeneni, S., Thieker, D. F., & Woods, R. J. (2018). Applying Pose Clustering and MD Simulations To Eliminate False Positives in Molecular Docking. *Journal of Chemical Information and Modeling*, *58*(3), 605–614. https://doi.org/10.1021/acs.jcim.7b00588

Matveieva, M., Cronin, M. T. D., & Polishchuk, P. (2019). Interpretation of QSAR Models: Mining Structural Patterns Taking into Account Molecular Context. *Molecular Informatics*, *38*(3), 1800084. https://doi.org/10.1002/minf.201800084

Mauri, A., Consonni, V., Pavan, M., & Todeschini, R. (2006). DRAGON SOFTWARE : AN EASY APPROACH TO, *56*, 237–248.

Medina-Franco, J. L., Martínez-Mayorga, K., Giulianotti, M. A., Houghten, R. A., & Pinilla, C. (2008). Visualization of the Chemical Space in Drug Discovery. *Current Computer-Aided Drug Design*, *4*, 322–333.

Mills, I., Cvitas, T., Homann, K., Kallay, N., & Kuchitsu, K. (1993). *International Union Of Pure and Applied Chemistry Physical Chemistry Division*.

Mok, N. Y., & Brown, N. (2017). Applications of Systematic Molecular Scaffold Enumeration to Enrich Structure–Activity Relationship Information. *Journal of Chemical Information and Modeling*, *57*(1), 27–35. https://doi.org/10.1021/acs.jcim.6b00386

Murtagh, F. (1985). Multidimensional Clustering Algorithm. *COMPSTAT L*, 9–30.

Naming and Indexing of Chemical Substances for Chemical Abstracts TM 2007 Edition A publication of Chemical Abstracts Service. (2008).

Nishibata, Y., & Itai, A. (1991). Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. *Tetrahedron*, *47*(43), 8985–8990. https://doi.org/10.1016/S0040-4020(01)86503-0

O'Boyle, N. M. (2012). Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics*, *4*(1), 22. https://doi.org/10.1186/1758-2946-4-22

Ostergard, P. R. J. (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, *120*, 197–207.

Pearce, B. C., Sofia, M. J., Good, A. C., Drexler, D. M., & Stock, D. A. (2006). An Empirical Process for the Design of High-Throughput Screening Deck Filters. *Journal of Chemical Information and Modeling*, *46*(3), 1060–1068. https://doi.org/10.1021/ci050504m

Pierce, A. C., Rao, G., & Bemis, G. W. (2004). BREED: Generating Novel Inhibitors through Hybridization of Known Ligands. Application to CDK2, P38, and HIV Protease. *Journal of Medicinal Chemistry*, *47*(11), 2768–2775.

https://doi.org/10.1021/jm030543u

Pogány, P., Arad, N., Genway, S., & Pickett, S. D. (2019). De Novo Molecule Design by Translating from Reduced Graphs to SMILES. *Journal of Chemical Information and Modeling*, *59*(3), 1136–1146. https://doi.org/10.1021/acs.jcim.8b00626

Polishchuk, P. (2017). Interpretation of Quantitative Structure–Activity Relationship Models: Past, Present, and Future. *Journal of Chemical Information and Modeling*, *57*(11), 2618–2639. https://doi.org/10.1021/acs.jcim.7b00274

Polishchuk, P. (2020). CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, *12*(1), 28. https://doi.org/10.1186/s13321-020-00431-w

Polishchuk, P. G., Kuźmin, V. E., Artemenko, A. G., & Muratov, E. N. (2013). Universal Approach for Structural Interpretation of QSAR/QSPR Models. *Molecular Informatics*, *32*(9–10), 843–853. https://doi.org/10.1002/minf.201300029

Polishchuk, P., Tinkov, O., Khristova, T., Ognichenko, L., Kosinskaya, A., Varnek, A., & Kuźmin, V. (2016). Structural and Physico-Chemical Interpretation (SPCI) of QSAR Models and Its Comparison with Matched Molecular Pair Analysis. *Journal of Chemical Information and Modeling*, *56*(8), 1455–1469. https://doi.org/10.1021/acs.jcim.6b00371

Rarey, M., & Dixon, J. S. (1998). Feature trees: A new molecular similarity measure based on tree matching. *Journal of Computer-Aided Molecular Design*, *12*, 471–490.

Ray, L. C., & Kirsch, R. A. (1957). Finding Chemical Records by Digital Computers. *Science*, *126*(3278), 814–819. https://doi.org/10.1126/science.126.3278.814

Raymond, J. W., Gardiner, E. J., & Willett, P. (2002). RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. *The Computer Journal*, *45*, 631–644.

RDKit: Open-Source Chemoinformatics. (2018).

Reymond, J.-L., & Awale, M. (2012). Exploring Chemical Space for Drug Discovery Using the Chemical Universe Database. *ACS Chemical Neuroscience*, *3*(9), 649–657. https://doi.org/10.1021/cn3000422

Rodríguez Benítez, A., Dürr, S. L., & Narayan, A. R. H. (2020). Radial Scope: A New Visualization Tool for Structure–Data Relationships. *Trends in Chemistry*, *2*(7), 587–589. https://doi.org/10.1016/j.trechm.2020.04.003

Rogers, D., & Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754. https://doi.org/10.1021/ci100050t

Rousseeuw, P. (1987). Silhouettes: A Graphical Aid To The Interpretation And Validation Of Cluster Analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Saldívar-González, F. I., Naveja, J. J., Palomino-Hernández, O., & Medina-Franco, J. L. (2017). Getting SMARt in drug discovery: chemoinformatics approaches for mining structure–multiple activity relationships. *RSC Advances*, *7*(2), 632–641. https://doi.org/10.1039/C6RA26230A

Schuffenhauer, A., Ertl, P., Roggo, S., Wetzel, S., Koch, M. A., & Waldmann, H. (2007). The Scaffold Tree − Visualization of the Scaffold Universe by Hierarchical Scaffold Classification. *Journal of Chemical Information and Modeling*, *47*(1), 47–58. https://doi.org/10.1021/ci600338x

Scott, O. B., & Edith Chan, A. W. (2020). ScaffoldGraph: an open-source library for the generation and analysis of molecular scaffold networks and scaffold trees. *Bioinformatics*, *36*(12), 3930–3931. https://doi.org/10.1093/bioinformatics/btaa219

Shannon, C. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*, 379–423.

Sheridan, R. P. (2019). Interpretation of QSAR Models by Coloring Atoms According to Changes in Predicted Activity: How Robust Is It? *Journal of Chemical Information and Modeling*, *59*(4), 1324–1337. https://doi.org/10.1021/acs.jcim.8b00825

Sheridan, R. P., Wang, W. M., Liaw, A., Ma, J., & Gifford, E. M. (2016). Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling*, *56*(12), 2353–2360. https://doi.org/10.1021/acs.jcim.6b00591

Stepniewska-Dziubinska, M., Zielenkiewicz, P., & Siedlecki, P. (2017). DeCAF—Discrimination, Comparison, Alignment Tool for 2D PHarmacophores. *Molecules*, *22*(7), 1128. https://doi.org/10.3390/molecules22071128

Stewart, K. D., Shiroda, M., & James, C. A. (2006). Drug Guru: A computer software program for drug design using medicinal chemistry rules. *Bioorganic & Medicinal Chemistry*, *14*(20), 7011–7022. https://doi.org/10.1016/j.bmc.2006.06.024

Stiefl, N., Watson, I. A., Baumann, K., & Zaliani, A. (2006). ErG: 2D Pharmacophore Descriptions for Scaffold Hopping. *Journal of Chemical Information and Modeling*, *46*(1), 208–220. https://doi.org/10.1021/ci050457y

Stiefl, N., & Zaliani, A. (2006). A Knowledge-Based Weighting Approach to Ligand-Based Virtual Screening. *Journal of Chemical Information and Modeling*, *46*(2), 587–596. https://doi.org/10.1021/ci050324c

Stumpfe, D., & Bajorath, J. (2016). Recent developments in SAR visualization. *MedChemComm*, *7*(6), 1045–1055. https://doi.org/10.1039/C6MD00108D

Taylor, R. (1995). Simulation Analysis of Experimental Design Strategies for Screening Random Compounds as Potential New Drugs and Agrochemicals. *Journal of Chemical Information and Computer Sciences*, *35*(1), 59–67. https://doi.org/10.1021/ci00023a009

Thomas, K. (2016, March). The Price of Health: The Cost of Developing New Medicines. *Guardian*.

Topliss, J. G. (1972). Utilization of operational schemes for analog synthesis in drug design. *Journal of Medicinal Chemistry*, *15*(10), 1006–1011. https://doi.org/10.1021/jm00280a002

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Walters, P. (2020). rd_filters.

Warr, W. A. (2011). Representation of chemical structures. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1*(4), 557–579. https://doi.org/10.1002/wcms.36

Wawer, M., Lounkine, E., Wassermann, A. M., & Bajorath, J. (2010). Data structures and computational tools for the extraction of SAR information from large compound sets. *Drug Discovery Today*, *15*(15–16), 630–639. https://doi.org/10.1016/j.drudis.2010.06.004

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, *28*(1), 31–36. https://doi.org/10.1021/ci00057a005

Wetzel, S., Klein, K., Renner, S., Rauh, D., Oprea, T. I., Mutzel, P., & Waldmann, H. (2009). Interactive exploration of chemical space with Scaffold Hunter. *Nature Chemical Biology*, *5*(8), 581–583. https://doi.org/10.1038/nchembio.187

Willett, P. (1987). *Similarity and clustering in chemical information systems*. Letchworth: Research Studies Press.

Willett, P. (2005). Searching Techniques for Databases of Two- and Three-Dimensional Chemical Structures. *Journal of Medicinal Chemistry*, *48*(13), 4183–4199. https://doi.org/10.1021/jm0582165

Wilson, J. W., & Free, S. M. (1964). Structure-Activity Studies. *Journal of Medicinal Chemistry*, *7*(4), 395–399.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining : Practical Machine Learning Tools and techniques*. Elsevier Inc.

Wollenhaupt, S., & Baumann, K. (2014). inSARa: Intuitive and Interactive SAR Interpretation by Reduced Graphs and Hierarchical MCS-Based Network Navigation. *Journal of Chemical Information and Modeling*, *54*(6), 1578–1595. https://doi.org/10.1021/ci4007547

Wood, D. R. (1997). An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, *21*, 211–217.

Yoshimori, A., & Bajorath, J. (2020). The SAR Matrix Method and an Artificially Intelligent Variant for the Identification and Structural Organization of Analog Series, SAR Analysis, and Compound Design. *Molecular Informatics*, *39*(12), 2000045. https://doi.org/10.1002/minf.202000045

Zahoránszky-kőhalmi, G., Bologa, C. G., & Oprea, T. I. (2016). Impact of similarity threshold on the topology of molecular similarity networks and clustering outcomes. *J Cheminform*, *8*, 1–17. https://doi.org/10.1186/s13321-016-0127-5

Zhang, B., Hu, Y., & Bajorath, J. (2014). AnalogExplorer: A New Method for Graphical Analysis of Analog Series and Associated Structure–Activity Relationship Information. *Journal of Medicinal Chemistry*, *57*(21), 9184–9194.

https://doi.org/10.1021/jm501391g