



Enhancing Reaction-based *de novo* Design using Machine Learning

A thesis submitted to the University of Sheffield in fulfilment of
the requirements for the degree of Doctor of Philosophy

by

Gian Marco Ghiandoni

This work was sponsored by



The University of Sheffield
Information School - Faculty of Social Sciences
December 2019



ML

Acknowledgements

Reaching the end of a doctoral study is something you do not do on your own. There are several people who deserve to be acknowledged for the achievement of such a personal milestone.

First, I want to thank my supervisors, Prof Val Gillet, Prof Beining Chen, and Dr Mike Bodkin: Val, *in primis*, for her precious experience and feedback, and most importantly, for trusting in me during all the time I have spent at Sheffield as a student; Beining, for her support and presence during this journey; Mike, for his continuous encouragement, for the stimulating discussions, and for giving me the opportunity to work in close contact with many skilled scientists. These people gave me the chance to challenge myself in this adventure.

The "Reaction Vector Cowboys", my colleagues James Webster and Dr James Wallace, for the close collaboration we have established and managed to preserve during all these years. Most of the concepts that have been formulated and developed in this work are the fruit of hours of discussions with these two exceptional scientists. Next to them, I want to thank Dr Dimitar Hristozov, for his great contribution to my research and the reaction vector project; Dr Antonio de la Vega de León and Dr Alessandro Checco, for their help, especially at the beginning of my studies; Dr Matthew Seddon, Dr Christina Founti, Dr Lucyantie Mazalan, Dr Philip Reeve, Jessica Stacey, Arshnous Marandi, and all the people I have worked with at the University of Sheffield.

I also want to thank the people who supported me from far away: My mother, grandmother, sisters, and my two uncles, without whom this could not have been possible. My friends Mario, Marco, Riccardo, Chris, and Tommy for their steady support; Margherita, for encouraging me to pursue my dreams; Luca, for helping and protecting my family.

Special acknowledgements are due to Prof Peter Willett, for supporting my research with his vast knowledge; Prof Jon Sayers, allowing me to work with him and the Sheffield Medical School; Dr Richard Mead, for providing me with an exceptional case study to work on and for putting his interest in the development of our techniques; Dr Stuart Flanagan, for carrying out the syntheses of the compounds designed in this work with great commitment; Dr Daniel Lowe, for providing me with the data for my experiments; Marion Leclerc, for her artistic contribution to this work.

Finally, I would like to thank Evotec U.K. and the Engineering and Physical Sciences Research Council (EPSRC) for their financial support and assistance.

Abstract

De novo design is a branch of chemoinformatics that is concerned with the rational design of molecular structures with desired properties, which specifically aims at achieving suitable pharmacological and safety profiles when applied to drug design. Scoring, construction, and search methods are the main components that are exploited by *de novo* design programs to explore the chemical space to encourage the cost-effective design of new chemical entities. In particular, construction methods are concerned with providing strategies for compound generation to address issues such as drug-likeness and synthetic accessibility.

Reaction-based *de novo* design consists of combining building blocks according to transformation rules that are extracted from collections of known reactions, intending to restrict the enumerated chemical space into a manageable number of synthetically accessible structures. The reaction vector is an example of a representation that encodes topological changes occurring in reactions, which has been integrated within a structure generation algorithm to increase the chances of generating molecules that are synthesisable.

The general aim of this study was to enhance reaction-based *de novo* design by developing machine learning approaches that exploit publicly available data on reactions. A series of algorithms for reaction standardisation, fingerprinting, and reaction vector database validation were introduced and applied to generate new data on which the entirety of this work relies. First, these collections were applied to the validation of a new ligand-based design tool. The tool was then used in a case study to design compounds which were eventually synthesised using very similar procedures to those suggested by the structure generator.

A reaction classification model and a novel hierarchical labelling system were then developed to introduce the possibility of applying transformations by class. The model was augmented with an algorithm for confidence estimation, and was used to classify two datasets from industry and the literature. Results from the classification suggest that the model can be used effectively to gain insights on the nature of reaction collections.

Classified reactions were further processed to build a reaction class recommendation model capable of suggesting appropriate reaction classes to apply to molecules according to their fingerprints. The model was validated, then integrated within the reaction vector-based design framework, which was assessed on its performance against the baseline algorithm. Results from the *de novo* design experiments indicate that the use of the recommendation model leads to a higher synthetic accessibility and a more efficient management of computational resources.

Table of Contents

Acknowledgements	I
Abstract	III
Table of Contents	V
List of Figures	XI
List of Tables	XXI
Table of Common Acronyms	XXV
Preface	XXVII
Chapter 1: Chemical Representations	1
1.1. Introduction	1
1.2. Molecular Representation	1
1.2.1. Molecular Graph Theory	4
1.2.2. Molecular Search Methods	5
1.3. Reaction Representation	10
1.3.1. Reaction Mapping	11
1.4. Reaction Databases	13
1.5. Reaction Search Methods	14
1.6. Reaction Classification	15
1.6.1. Model-driven Methods	15
1.6.2. Data-driven Methods	18
1.7. Conclusions	22
Chapter 2: De novo Molecular Design	23
2.1. Introduction	23
2.2. The Molecular Design Route	23
2.3. De novo Design Components	25
2.4. Scoring Components	25
2.4.1. Structure-based Scoring	26
2.4.2. Ligand-based Scoring	28
2.5. Construction Components	30
2.5.1. Atom-based Construction	30
2.5.2. Fragment-based Construction	31
2.6. Search Components	34
2.6.1. Stochastic Search	34
2.6.2. Deterministic Search	36
2.7. Artificial Intelligence in de novo Design	37

2.8.	Conclusions	42
Chapter 3:	Reaction Vectors	43
3.1.	Introduction	43
3.2.	The Concept of Reaction Vector	43
3.3.	Structure Generation Using Reaction Vectors.....	46
3.3.1.	Original Algorithm.....	47
3.3.2.	Revised Algorithm.....	49
3.3.3.	Handling Multiple Reactants.....	52
3.4.	Conclusions	53
Chapter 4:	Machine Learning.....	55
4.1.	Introduction	55
4.2.	Supervised Classification	55
4.2.1.	Classification Problems	57
4.3.	Classification Algorithms	58
4.3.1.	k-Nearest Neighbours	59
4.3.2.	Decision Trees.....	60
4.3.3.	Support Vector Machine	62
4.4.	Confidence Estimation using Conformal Prediction	64
4.5.	Multi-label Classification.....	67
4.5.1.	Multi-label Approaches	67
4.5.2.	Multi-label Classification in Chemoinformatics	73
4.6.	Conclusions	74
Chapter 5:	Reaction Data	77
5.1.	Introduction	77
5.2.	Reaction Standardisation	78
5.2.1.	Introduction	78
5.2.2.	Methods	78
5.2.3.	KNIME Implementation	82
5.3.	Dynamic Reaction Fingerprints.....	83
5.3.1.	Introduction	83
5.3.2.	Method.....	83
5.4.	US Pharmaceutical Patents.....	85
5.4.1.	Introduction	85
5.4.2.	Standardisation	87
5.4.3.	Dataset Intersections.....	91
5.4.4.	Database Validation and Encoding	95

5.4.5.	Fingerprint Encoding.....	97
5.4.6.	Label Optimisation.....	99
5.5.	External Data.....	102
5.5.1.	Introduction	102
5.5.2.	Standardisation	103
5.5.3.	Database and Fingerprint Encoding.....	104
5.6.	Conclusions	105
Chapter 6: Pseudoretrosynthetic de novo Design		107
6.1.	Introduction	107
6.2.	The RENATE Algorithm.....	108
6.2.1.	KNIME Implementation.....	112
6.3.	Top 200 Drugs 2017 Validation.....	113
6.3.1.	Introduction	113
6.3.2.	Data Selection	114
6.3.3.	Building Block and Reaction Vector Selection.....	115
6.3.4.	Method.....	116
6.3.5.	Results and Discussion	116
6.4.	PARP1 Inhibitor Design	121
6.4.1.	Introduction	121
6.4.2.	Target and Ligand Selection.....	121
6.4.3.	Design Strategy	123
6.4.4.	Setup Selection	125
6.4.5.	Method.....	133
6.4.6.	Results and Discussion	134
6.4.7.	Compound Selection.....	135
6.4.8.	Compound Synthesis	138
6.4.9.	Estimation of BBB Penetration.....	141
6.5.	Conclusions	142
Chapter 7: Reaction Classification		145
7.1.	Introduction	145
7.2.	50-class Model	146
7.2.1.	Introduction	146
7.2.2.	Data Selection	147
7.2.3.	Method.....	150
7.2.4.	Results and Discussion	151
7.3.	336-class Model	154

7.3.1.	Introduction	154
7.3.2.	Data Selection	154
7.3.3.	Methods	156
7.3.4.	Results and Discussion	162
7.3.5.	KNIME Implementation	176
7.4.	Applications	177
7.4.1.	Introduction	177
7.4.2.	Evotec ELN	177
7.4.3.	JMC 2008.....	188
7.5.	Conclusions	193
Chapter 8: Reaction Class Recommendation		195
8.1.	Introduction	195
8.2.	Theoretical Basis.....	196
8.3.	Proof of Concept Model	200
8.3.1.	Introduction	200
8.3.2.	Molecular Descriptors.....	200
8.3.3.	Data Selection	202
8.3.4.	Multi-label Approaches and Classifiers.....	205
8.3.5.	Method.....	206
8.3.6.	Results and Discussion	207
8.4.	Final Model.....	212
8.4.1.	Introduction	212
8.4.2.	Molecular Descriptors.....	213
8.4.3.	Data Selection	213
8.4.4.	Multi-label Approaches and Classifiers.....	216
8.4.5.	Methods	217
8.4.6.	Results and Discussion	217
8.4.7.	KNIME Implementation	222
8.5.	Validation of the Reaction Class Recommender	223
8.5.1.	JMC 2018 Class Prediction	223
8.5.2.	DSPL Single-step de novo Design	229
8.5.3.	Top 200 Drugs 2017 Recommended Validation	235
8.6.	Conclusions	239
Chapter 9: Conclusions and Future Work		241
9.1.	Conclusions	241
9.2.	Limitations and Future Work	243

Metrics and Properties	247
Classification Metrics	247
Regression Metrics.....	249
Pharmacokinetic (PK) Properties.....	250
Appendix A	251
Appendix B	285
Appendix C	293
Appendix D.....	301
Bibliography	319

List of Figures

Figure 1.1: Example of different notations for Menthol. The notations ‘@’ and ‘@@’ are used to describe chiral centres.	2
Figure 1.2: Example of connection table for Menthol.	3
Figure 1.3: Example of a molecular graphic representation (left) converted into a graph (right), in which atoms and bonds are identified by nodes and edges, respectively.	4
Figure 1.4: Examples of substructure searching where the query substructure (bolded) matches two drug structures, namely Nadolol and Heroin.	6
Figure 1.5: Example of bit string comparison prior to substructure search. The example shows that only three out of five query features are matched by the database molecule, hence the entry is discarded.	7
Figure 1.6: Example of distance computation between two vectors A and B using Euclidean, Manhattan, and Cosine metrics.	9
Figure 1.7: Reaction SMILES for an acid-catalysed dehydration.	11
Figure 1.8: SMIRKS notation for an amide formation reaction.	11
Figure 1.9: Examples of common substructures in a reaction. The MCS corresponds to the largest common substructure that is shared between reactant and product.	13
Figure 1.10: Example of a reaction query for an acid-catalysed dehydration on CASREACT. .	15
Figure 1.11: Vléduts, Fujita, and Hendrickson reaction representations for a six-atom reaction centre. Image readapted (Chen, 2003).	17
Figure 1.12: InfoChem’s CLASSIFY algorithm: Broad, medium, and narrow spheres describe increasing levels of inclusion of atoms in the reaction centre and its proximal environment. Image readapted (Kraut <i>et al.</i> , 2013).	21
Figure 2.1: Chemical space activity landscape. The axes lying on the plane describe molecular features while the vertical axis represents activity. Areas where the activity is lower and higher are also indicated by cold and hot colours, respectively.	24
Figure 2.2: Pseudoretrosynthetic design applied to the molecule Celecoxib. First, the query ligand bonds are (a) broken to yield a set of key fragments. Fragments are then used as references to retrieve similar blocks that are (b) recombined to yield novel compounds with properties similar to the original query.	33
Figure 3.1: Example of reaction vector generation using AP2 descriptors. Lost and gained atom pairs are highlighted in red and blue, respectively.	45

Figure 3.2: The AP2 reaction vector.	46
Figure 3.3: Example of structure generation using the algorithm introduced by Patel <i>et al.</i> (2009):	
(a) the starting material is processed by removing the negative AP2s in the vector. Multiple fragments with unsatisfied valence can be obtained after this procedure; (b) an abstract fragment is processed by adding the positive AP2s in the reaction vector; (c) as the growing continues, the new structures are checked on the presence of the positive AP3s in the reaction vector. Structures that generate AP3s that are not described in the reaction vector are considered as invalid.	48
Figure 3.4: Example of recombination path generation using the algorithm developed by Hristozov <i>et al.</i> (2011): (a) the reaction is used to produce its AP2+AP3 reaction vector; (b) the starting material and product are decomposed using the negative and positive atom pairs in the reaction vector, respectively. A series of fragments from both starting material and product are generated. The shared fragments are described in bold; (c) if any shared fragments can be found, the algorithm stores the recombination path which includes <i>base fragment</i> (unchanging substructure determined using MCS) and <i>reaction fragment</i> (changing substructure).	50
Figure 3.5: Example of structure generation using the algorithm developed by Hristozov <i>et al.</i> (2011): (a) a new starting material is checked against the database entries to find reaction vectors with matching negative AP2s. If a matching entry is found, the starting material is processed by removing its atoms and bonds according to the APs described in the entry, to yield an abstract fragment; (b) the abstract fragment is combined with the reaction fragment stored in the matching database entry to yield a new product.	52
Figure 3.6: Example of C-C bond formation using two reactants.	52
Figure 4.1: Classification problem types described on a two-dimensional space: (a) <i>binary</i> : entries can only belong to either class A (red) or B (blue); (b) <i>multi-class</i> : entries can only belong to class A (red), B (blue), or C (grey); <i>multi-label</i> : entries can belong to class A (red), B (blue), or both classes A and B (blue with red edges).	57
Figure 4.2: Examples of different classifier outputs on the same binary data, where the different coloured areas indicate how algorithms define the domain subspaces, which in turn determine how entries are classified. Entries are represented by coloured dots according to the class they belong to. Image adapted for the purpose of this work (Scikit-learn, 2007).	58

Figure 4.3: kNN classifications ($k = 3, 5$) on a binary dataset (A, B). The test entry is described as a star while training instances are points with their corresponding classes reported in brackets. Frequencies (P_A, P_B) are calculated as ratios between particular instances (e.g. A) on the total number of instances determined by k 59

Figure 4.4: Example of decision tree. The square and circles represent root and decision nodes, respectively, whilst the triangles represent terminal leaves. 60

Figure 4.5: SVM linear classification, where support vectors and remaining instances are represented as filled and unfilled points, respectively: (a) a number of separation hyperplanes between the points of two classes can be found; (b) however, the maximum margin of separation is produced only by one optimal hyperplane. 63

Figure 4.6: Example of kernel trick in binary classification. The left side describes a non-linear separation between points in a two-dimensional (\mathbb{R}^2) space. The right side shows that after the kernel trick, the points have been mapped into a three-dimensional space (\mathbb{R}^3), where an optimal hyperplane of separation can be found. 63

Figure 4.7: Example of a *p-value* calculation. The *p-value* is represented as the area under the curve beyond the observed data point. Image adapted from the literature (Jawlik, 2016). 65

Figure 4.8: Binary Relevance (BR) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. The original dataset (left) is split into a number of binary sets (right) equal to the number of labels to predict. 68

Figure 4.9: Classifier Chain (CC) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. The original dataset (left) is split into a number of binary sets (right) equal to the number of labels to predict. 69

Figure 4.10: Label Powerset (LP) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. Label columns in the original set (left) are merged together to form a single label column containing multiple classes (right). 70

Figure 4.11: ML-kNN classifications ($k = 3, 5$) on a binary dataset (A, B). The test entry is described as a star while training instances are points with their corresponding classes reported in brackets. Frequencies (P_A, P_B) are calculated as ratios between particular instances (e.g. A) on the total number of instances determined by k similarly to binary or multi-class problems, yet in multi-label classification, the same instances can contribute to increasing the frequencies of more than one label. 72

Figure 5.1: Unmapped compound removal.	79
Figure 5.2: Reaction balancing for missing fragments.	80
Figure 5.3: Reaction balancing for multi-product transformations.	80
Figure 5.4: Reaction indexing of three different examples from the same patent.	81
Figure 5.5: Examples of reactions associated with the same AP2 vectors and different AP2+AP3 vectors. AP2 reaction centres are coloured in blue and AP3 extensions are coloured in red.	82
Figure 5.6: Reaction standardisation KNIME workflow.	83
Figure 5.7: Dynamic fingerprint conversion algorithm. Reaction vectors represented as strings are converted to true vectors. The vector elements are integers with negative values indicating atom pairs that are lost from the reactants; positive values indicating atom pairs that are gained in the products; and zeros indicating atom pairs that are not present in the vector.	84
Figure 5.8: Reaction vector dataset adjustment: blue and red features are retained in the <i>adjusted</i> <i>slave</i> dataset, while grey features are discarded since they are not described in the <i>master</i> dataset.	85
Figure 5.9: Reactions per reaction vector ratios expressed in log10 scale for the USPD and USPDA datasets before the duplicate filtering. Vectors are sorted by descending order according to their numbers of examples.	89
Figure 5.10: Example of indistinguishable reaction classes after the duplicate filtering.	90
Figure 5.11: Reaction vector class distributions for the filtered USPD and USPDA sets. Classes are sorted by descending order according to their numbers of vectors.	91
Figure 5.12: “Only Classified Data” datasets intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.	92
Figure 5.13: “Balancing Tool” dataset intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.	93
Figure 5.14: “Duplicate Filtering” dataset intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.	94
Figure 5.15: Examples of validated (top) and rejected (bottom) reactions from the USPD dataset. The top reaction yields a structure that is identical to the original product, while the bottom reaction produces a ring closure that yields a different structure compared to that described in the original reaction.	96

Figure 5.16: USPD AP2+AP3 atom-pair frequency scatter plot (sorted by descending frequency count).	99
Figure 5.17: Example of two classes that are not distinguishable after the reaction standardisation workflow.	100
Figure 6.1: Fragment (highlighted in bold on the drug structure) generated by BRICS decomposition of the drug Celecoxib: The fragment has five heavy atoms and three connections, hence its size is equal to eight.	109
Figure 6.2: Scaffold and reagent identification procedure applied to the fragments derived from Celecoxib.	110
Figure 6.3: RENATE KNIME workflow.	113
Figure 6.4: Lovastatin and Simvastatin: Two cholesterol-lowering medications from the family of statins which differ by only a methyl group.	114
Figure 6.5: Lipinski's RO5 property distributions covered by the 73 drugs selected for the validation of RENATE.	115
Figure 6.6: Queries that failed the BRICS decomposition. Potential fragmentation bonds are highlighted in bold.	117
Figure 6.7: Examples of some best candidate-drug pairs according to RDKit-ECFP4 similarity generated from the USPD design pipeline.	119
Figure 6.8: DNA repairing mechanisms mediated by PARP1: a DNA strand break activates PARP1 which in turn activates a series of enzymes responsible for DNA repairing.	122
Figure 6.9: PARP1 inhibitors, for which crystallographic data is available in the PDB, showing similar interactions with several protein residues. Groups involved with hydrogen bonding as donors and acceptors are coloured in blue and red, respectively, while substructures involved with π - π stacking interactions are highlighted in bold.	123
Figure 6.10: PARP1 design <i>scoring</i> module. <i>Active</i> components drive the algorithm at each step of the design, while <i>passive</i> components are applied at the end of the process to refine the selection of the most promising candidates.	124
Figure 6.11: PARP1 catalytic domain (PDB ID: 4R6E) and its complexed ligands. The DNA binding pocket is coloured in salmon pink while the rest of the protein is in light purple.	129
Figure 6.12: PARP1 3D (left) and 2D (right) key residue interactions with Niraparib. Yellow and black dashed lines indicate hydrogen bonds in 3D and 2D representations, respectively.	

Green solid lines show hydrophobic interactions and green dashed lines show π - π stacking interactions in the 2D diagram.	130
Figure 6.13: Overlap between docked (green) and experimental (purple) poses of Niraparib. .	131
Figure 6.14: Overlap between docked (green) and experimental (purple) poses of the other inhibitors.	132
Figure 6.15: Overlap between candidates (e.g. Row26) (green) and reference drugs (e.g. Olaparib) (purple). Candidate IDs are reported in brackets. The residue Tyr907 is hidden to ease the view of the poses. Hydrogen bond interactions between protein and ligands are displayed in yellow.	136
Figure 6.16: Examples of valid and invalid candidates designed by RENATE using Talazoparib as a query (A, B, and C fragments are coloured in black, blue, and red, respectively). .	138
Figure 6.17: Compound synthesis summary scheme. The diagram describes on the left the names of the reference ligands (e.g. Olaparib), which are connected to their candidate structures (e.g. Row26). Candidate structures are associated with short descriptions on the right side of the chart, which describe the additional/alternative chemistry (e.g. Protection Chemistry) adopted to obtain the candidates.	139
Figure 7.1: Creation of a balanced collection of 50 reaction classes from the USPD AP2+AP3 fingerprint dataset: (a) 727 imbalanced classes; (b) selection of 50 most populated classes; (c) downsampling according to the minority class.	147
Figure 7.2: Reaction class coverage across fingerprint datasets. Represented classes are shown in blue and missing classes are in white.	148
Figure 7.3: 50-class AP2+AP3 USPDA dataset class distribution sorted by vector count by descending order.	150
Figure 7.4: Normalised confusion matrices of the internal validations of the 50-class AP2+AP3 models across different classifiers.	152
Figure 7.5: 30-example fingerprint dataset class distributions sorted by vector count by descending order.	155
Figure 7.6: RF performance-against-parameter trends in the RS.	165
Figure 7.7: SVM performance-against-parameter trends in the RS.	165
Figure 7.8: Correlation scatter plot of training data and performance metrics for internal (USPD test set - blue) and external (USPDA external set - red) validations of the RF classifier. The x-axes represent the number of examples in each class in the training data. Each dot represents one reaction class.	167

Figure 7.9: Examples of classes involving more (“C-C Bond Formation (Methylation)”) or less (“Synthesis (1-2-4-Triazole)”) variable reaction centres.	168
Figure 7.10: False positive trends for classifiers 3, 4, 5 from the internal validation.....	170
Figure 7.11: False positive trends for classifiers 3, 4, 5 from the external validation.	171
Figure 7.12: <i>Micro</i> (left) and <i>weighted</i> (right) F1-scores trends at increasing amounts of training data on the prediction of the external data set.	172
Figure 7.13: Absolute numbers (left) and ratios (right) of true and false predictions associated with each level of probability.	173
Figure 7.14: Absolute numbers (left) and ratios (right) of true and false predictions associated with each level of confidence (top) and credibility (bottom).	174
Figure 7.15: RF-CP classification KNIME workflow.	176
Figure 7.16: Confidence (left) and credibility (right) scores of the Evotec ELN data reaction classification.....	178
Figure 7.17: Level-1 classification of the Evotec ELN data.....	179
Figure 7.18: Absolute count time series of the Evotec ELN level-1 classes.	184
Figure 7.19: Normalised count time series of the Evotec ELN level-1 classes.	185
Figure 7.20: Heatmap that describes the lower triangular pairwise matrix of the Evotec ELN level-1 class correlation coefficients.	186
Figure 7.21: Yield time series of the Evotec ELN.....	187
Figure 7.22: Confidence (left) and credibility (right) scores of the JMC dataset reaction classification.....	189
Figure 7.23: Level-1 classification of the JMC 2008 dataset.	190
Figure 8.1: Examples of correct and incorrect grouping by molecular features.	198
Figure 8.2: Example of feature encoding and label <i>pivoting</i>	199
Figure 8.3: Example of mapping-based starting material extraction.....	200
Figure 8.4: Property distributions covered by the PoC USPD subset.....	203
Figure 8.5: Level-4 (left) and level-2 (right) PoC USPD subsets class distributions.	204
Figure 8.6: Model creation tree diagram: bolded nodes with directed edges represent an example of combinatorial enumeration, while dashed nodes with non-directed edges represent non-expanded paths.	207
Figure 8.7: PoC model: Level-4- and level-2 label dataset comparison.	208
Figure 8.8: PoC model: PT and AA approaches comparison using level-2 label datasets.	209

Figure 8.9: PoC model: Performance metrics comparison of PT approaches using RF and SVM and level-2 label datasets.	209
Figure 8.10: PoC model: Loss metrics comparison of PT approaches using RF and SVM and level-2 label datasets.	210
Figure 8.11: PoC model: Correlation plots between <i>micro</i> F1-score and 0/1 Loss (left) and Hamming Loss (right) for the level-2 label models.	211
Figure 8.12: PoC model: Classifier comparison of PT approaches using level-2 label datasets.	211
Figure 8.13: Final USPD subset level-1 class composition.	214
Figure 8.14: Property distribution covered by the Final USPD subset.	214
Figure 8.15: 319- (left) and 259-class (right) Final USPD subsets class distributions.	215
Figure 8.16: Final model: Level-3 and level-2 label dataset comparison.	218
Figure 8.17: Final model: BR, CC, RAKELo, and RAKELd approaches comparison using level-4 label datasets.	219
Figure 8.18: Final model: Classifier comparison for CC models using level-3 label datasets.	221
Figure 8.19: Recommended reaction vector-based design KNIME workflow.	222
Figure 8.20: Level-1 classification of the JMC 2018 dataset.	224
Figure 8.21: Property distribution of the starting materials extracted from the classified JMC 2018 test set.	225
Figure 8.22: Property distributions for the starting materials coloured by correct (blue), wrong (red), and no-recommendation (grey) following application of the recommenders.	226
Figure 8.23: Additional class recommendation test using the CC-RF MACCS model. The recommender did not suggest the class originally associated with the top molecule; however, the suggested transformation produces a new product for which the correct class is predicted.	228
Figure 8.24: Property separation between recommended and non-recommended starting materials.	230
Figure 8.25: RSynth and SAscore distributions per library.	233
Figure 8.26: Level-1 label class distributions across libraries.	234
Figure 8.27: Target hits distributions across libraries.	235
Figure 8.28: Comparison between best compounds generated without and with the use of the recommender. Designed compounds are also annotated with their similarity to their reference drugs using RDKit-ECFP4.	237

Figure 8.29: Example of best compound from the recommended design, associated with a similarity to its reference greater than that of the best compound from the non-recommended design. The reactions in the box describe the first step of the design, where the use of two different piperidines (highlighted in bold) result in the generation of different synthetic paths.....238

List of Tables

Table 1.1: A list of free organic reaction databases.	14
Table 1.2: A list of commercial organic reaction databases.	14
Table 5.1: USPD dataset description through the standardisation workflow.	88
Table 5.2: USPDA dataset description through the standardisation workflow.	88
Table 5.3: Reaction vectors per class statistics for the filtered USPD and USPDA sets.	91
Table 5.4: USPD and USPDA reaction validation and database encoding results.	97
Table 5.5: USPD and USPDA fingerprint dataset descriptions.	98
Table 5.6: NameRxn labelling of some cross-coupling reactions.	101
Table 5.7: NameRxn to SHREC label replacement for some cross-coupling reactions.	102
Table 5.8: JMC 2008 dataset description through the standardisation workflow.	104
Table 5.9: JMC 2018 dataset description through the standardisation workflow.	104
Table 5.10: Evotec ELN dataset description through the standardisation workflow.	104
Table 5.11: External dataset database encoding results.	105
Table 5.12: External fingerprint dataset descriptions.	105
Table 6.1: Top 200 Drugs 2017 design RENATE parameters.	116
Table 6.2: Statistics from the pair-wise similarities between queries and their corresponding best compounds from USPD and JMC 2018 designs.	117
Table 6.3: Comparison between virtual and real synthetic steps for the drug structures reproduced during the design.	120
Table 6.4: Selection of molecular descriptors for the <i>scoring</i> components.	125
Table 6.5: PARP1 QSAR model validation results.	126
Table 6.6: PARP1 QSAR Morgan (Radius 2) (Count) model best-cross validation and optimised model validation metrics.	126
Table 6.7: Pgp, BCRP, BBB classification dataset descriptions.	126
Table 6.8: Pgp, BCRP, and BBB model best-cross validation and optimised model validation metrics.	127
Table 6.9: Pgp, BCRP, and BBB model classification results for the PARP1 inhibitors selected for the design (true classes are reported in brackets).	128
Table 6.10: PARP1 crystallographic data description.	129
Table 6.11: General PARP1 docking parameters in GOLD.	131

Table 6.12: PARP1 docking validation scores and numbers of consistent poses per ligand. The original drugs are highlighted in grey.....	132
Table 6.13: PARP1 design RENATE parameters.	133
Table 6.14: PARP1 design original and enumerated candidates.....	134
Table 6.15: Summary of results of the PARP1 design synthesised compounds. The quantity and purity obtained are reported along with the proposed synthetic steps and the actual steps which were needed to obtain the compounds.	140
Table 6.16: PARP1 design selected candidates' pharmacokinetic properties.	142
Table 7.1: Pre-processed 50-class fingerprint datasets. The total number of unique reaction vectors is shown for the different types of fingerprints (number of rows in the datasets), along with the number of unique atom pairs (the number of columns in the datasets), and the number of unique reaction vectors in each class.	148
Table 7.2: Training and test internal validation datasets across fingerprint types.	149
Table 7.3: Classifier parameters.	150
Table 7.4: Macro averages of recall, precision, and F1-score metrics across fingerprint types in the 50-class model internal validation.	151
Table 7.5: <i>Micro</i> and <i>weighted</i> averages of Recall, Precision, and F1-score metrics in the 50-class model external validation.....	154
Table 7.6: Pre-processed 30-example fingerprint datasets.	155
Table 7.7: XGBoost (XGB) parameters.	156
Table 7.8: Random Search parameter distributions.....	158
Table 7.9: Evolutionary Optimisation parameter distributions.	158
Table 7.10: Weight settings.....	159
Table 7.11: Random Forests (RF) optimised parameters and weights.	160
Table 7.12: USPD <i>proper training</i> and <i>calibration</i> sets.	161
Table 7.13: <i>Weighted</i> F1-scores of the training data validation.....	162
Table 7.14: <i>Weighted</i> F1-scores of internal and external validations.	163
Table 7.15: 5-fold partition training times.	163
Table 7.16: RS cross-validation best parameters and scores.	164
Table 7.17: RS internal and external validation F1-scores.	164
Table 7.18: EO cross-validation best parameters and scores.	166
Table 7.19: EO internal and external validation F1-scores.....	166

Table 7.20: 10 classes with the highest number of false positives in the EO internal validation.	169
Table 7.21: 10 classes with the highest number of false positives in the EO external validation.	169
Table 7.22: Class-weighted internal and external validation F1-scores.	169
Table 7.23: Variations of performance (left) and percentage of filtered entries (right) associated with different probability cut-off levels.	173
Table 7.24: Variations of performance (left) and percentage of filtered entries (right) associated with different credibility cut-off levels.	175
Table 7.25: Credibility score threshold filtering tests applied on the Evotec ELN data.	178
Table 7.26: Top 15 reaction classes in the Evotec ELN data according to the level-2 labelling system.	181
Table 7.27: Top 15 reaction classes in the Evotec ELN data according to the level-4 labelling system.	182
Table 7.28: Reaction counts per year in the Evotec ELN.	184
Table 7.29: Credibility score threshold filtering levels applied on the JMC dataset.	189
Table 7.30: Top 15 reaction classes in the JMC 2008 according to the level-2 labelling system.	191
Table 7.31: Top 15 reaction classes in the JMC 2008 according to the level-4 labelling system.	192
Table 8.1: Selection of molecular descriptions investigated for the PoC model.	201
Table 8.2: PoC USPD subset description.	202
Table 8.3: Filtered PoC USPD subset descriptions.	203
Table 8.4: PoC model: Molecular description datasets generated after the <i>pivoting</i>	205
Table 8.5: Classifier parameters.	206
Table 8.6: PoC model: Performance metrics statistical analysis of PT approaches using RF and SVM and level-2 label datasets.	209
Table 8.7: PoC model: Loss metrics statistical analysis of PT approaches using RF and SVM and level-2 label datasets.	210
Table 8.8: PoC model: 15 best performing level-2 label PT models according to their <i>micro</i> F1- scores.	212
Table 8.9: Final USPD subset description.	213
Table 8.10: Final USPD Grants subset descriptions.	215

Table 8.11: Final model: Molecular description datasets generated after the <i>pivoting</i>	216
Table 8.12: RAKEL parameters.	217
Table 8.13: Final model: Maximum memory request per data and label type.	219
Table 8.14: Final model: Performance metrics statistical analysis of BR, CC, RAKELo, and RAKELd approaches using RF and SVM and level-3 label datasets.	219
Table 8.15: Final model: Maximum amounts of memory per data and approach type.	220
Table 8.16: Final model: level-3 label CC model performance metrics.	221
Table 8.17: Performance of the Avalon 1024-bit and MACCS recommenders on the JMC 2018 dataset expressed as percentages.....	225
Table 8.18: Ratios of wrongly predicted and non-recommended entries at level-1 of the hierarchy for the Avalon 1024-bit and MACCS recommenders on the JMC 2018 set.....	227
Table 8.19: The minimum, maximum, mean and median number of recommended classes per starting material for the different classification levels.	231
Table 8.20: Library statistics for recommended and control pipelines.	231
Table 8.21: Statistics from the pair-wise similarities between queries and their corresponding best compounds from the USPD design - without and with the use of the recommender.....	236
Table 8.22: Comparison between drugs regenerated without and with the use of the recommender. Each ligand is described in the number of steps required for its regeneration, number of products generated by the algorithm and enumeration times.	239

Table of Common Acronyms

AI: <i>Artificial Intelligence</i>	RC: <i>Reaction Centre</i>
AP: <i>Atom-Pair</i>	RF: <i>Random Forests</i>
BBB: <i>Blood-Brain Barrier</i>	RL: <i>Reinforcement Learning</i>
BCRP: <i>Breast Cancer Resistance Protein</i>	RS: <i>Random Search</i>
BR: <i>Binary Relevance</i>	RSC: <i>Royal Society of Chemistry</i>
CC: <i>Classifier Chain</i>	RV: <i>Reaction Vector</i>
CNS: <i>Central Nervous System</i>	RVSG: <i>Reaction Vector Structure Generator</i>
CP: <i>Conformal Prediction</i>	SA: <i>Synthetic Accessibility</i>
DL: <i>Deep Learning</i>	SHREC: <i>Sheffield Hierarchical REaction Classification</i>
DNN: <i>Deep Neural Network</i>	SM: <i>Starting Material</i>
ELN: <i>Electronic Laboratory Notebook</i>	SMARTS: <i>SMiles ARbitrary Target Specification</i>
EO: <i>Evolutionary Optimisation</i>	SMILES: <i>Simplified Molecular-Input Line-Entry System</i>
FF: <i>Force-Field</i>	SMIRKS: <i>Simple Molecular Input Reaction Kinetic String</i>
FP: <i>FingerPrint</i>	SRSG: <i>Superimposed Reaction Skeleton Graph</i>
GA: <i>Genetic Algorithm</i>	SVM: <i>Support Vector Machine</i>
GB: <i>Gradient Boosting</i>	TPSA: <i>Topological Polar Surface Area</i>
GUI: <i>Graphical User Interface</i>	USPD: <i>United States Patent Data</i>
IUPAC: <i>International Union of Pure and Applied Chemistry</i>	USPTO: <i>US Patent and Trademark Office</i>
LP: <i>Label Powerset</i>	
MAE: <i>Mean Absolute Error</i>	
MCC: <i>Matthews Correlation Coefficient</i>	
MCS: <i>Maximal Common Substructure</i>	
ML: <i>Machine Learning</i>	
MPO: <i>MultiParameter Optimization</i>	
MSE: <i>Mean Squared Error</i>	
NN: <i>Neural Network</i>	
NP: <i>Nondeterministic Polynomial Time</i>	
PDB: <i>Protein Data Bank</i>	
PGP: <i>P-GlycoProtein</i>	
QED: <i>Quantitative Estimation of Drug-likeness</i>	
QSAR: <i>Quantitative Structure-Activity relationship</i>	
QSPR: <i>Quantitative Structure-Property Relationship</i>	

Preface

The term “chemoinformatics” refers to ‘the application of informatics methods to solve chemical problems’ (Gasteiger, 2006). Although this term and its explicit definition appeared for the first time only in 1998 (Brown, 1998), this branch of computational approaches actually started gaining a role in the field of drug discovery in the late 1950s (Willett, 2011). This work began with the development of methods for the searching of chemical compounds in databases (Ray and Kirsch, 1957) and the prediction of quantitative structure-activity relationships (QSARs) between molecules and biological targets (Hansch *et al.*, 1962). Nevertheless, the computational costs of manipulating chemical information for modelling purposes were so expensive at that time, that they limited the effective growth of this discipline for several decades.

Nowadays, computational approaches (also referred to as *in-silico*) are routinely used in drug discovery with the aim of reducing the time and costs necessary for the identification of novel chemical entities (NCEs) with desired pharmacological properties. Molecular *de novo* (from Latin ‘on new’) design is a branch of chemoinformatics that is concerned with the creation of such structures ‘from scratch’. A major issue that is commonly faced with *de novo* design programs is that they generate large numbers of structures, which are often difficult to synthesise in reality (Gasteiger, 2007). More sophisticated methods attempt to account for these aspects by connecting molecular fragments together according to predefined rules, which are typically derived from collections of pharmaceutically relevant molecules or from the synthetic literature. Reaction-based *de novo* design represents a further advancement of these methods since it consists of generating molecular structures by simulating real synthetic pathways with the aim of incorporating as much chemical knowledge as possible during the design.

The Sheffield Chemoinformatics Research Group has developed a novel reaction-based *de novo* design approach (Patel *et al.*, 2009) which relies on the concept of the *reaction vector* formalised by Broughton *et al.* (2003). Reaction vectors can be created from examples of known reactions by subtracting the topological descriptions of the

reactants from those of the products to yield a set of changing features. Reaction vector-based *de novo* design involves the use of reaction vectors as templates for the generation of new molecular structures, with the aim of maximising their synthetic accessibility while preserving the chance of exploring novel chemical space. The method has been validated and optimised in several contexts in order to demonstrate its effectiveness, increase its efficiency, and extend its applications (Hristozov *et al.*, 2011) (Gillet, Bodkin and Hristozov, 2013) (Wallace, 2016).

The reaction vector method can be easily adapted to a variety of scenarios, for example, by enabling the use of in-house reactions (e.g. from laboratory notebooks) as references for the design of new compounds. However, the lack of protocols for reaction standardisation and validation, and the absence of guidelines on the practical use of reaction vectors in medicinal chemistry, constitute a major obstacle for their application in drug discovery. The current design framework also presents two limitations. The first relates to the fact that vectors cannot be promptly applied by reaction-type, for example, to produce products from a particular class of reactions (e.g. bromination), which would be useful to support the design of new compounds in the laboratory. The second regards the nature of the approach itself, which generates new structures only by accounting for the molecular features that are involved in the core of reactions (i.e., atoms that are directly involved in the transformations plus their proximal neighbours); hence, the method can fail to account for the presence of distant functionalities that can reduce the reactivity of compounds or compete in certain conditions. This thesis aims to overcome these issues and enhance the current method through the use of automation and machine learning.

The thesis begins with an introduction to the techniques involved in this work. Chapter 1 first describes the methods that are used for molecular and reaction representation, focussing on the principles of graph theory and the approaches that are used for searching in databases. The chapter also introduces the concepts and the evolution of algorithms for reaction mapping and classification. Chapter 2 discusses the main concepts on which molecular *de novo* design relies, and describes the evolution of

the components of design algorithms, including the recent introduction of artificial intelligence (AI) technologies for molecule generation. Chapter 3 introduces the concept of the reaction vector developed at Sheffield and describes its implementation in a structure generation algorithm. The chapter illustrates the functioning of the first structure generator implemented by Patel and colleagues (2009), then it grounds the motivations for the introduction of the current algorithm proposed by Hristozov and colleagues (2011). Chapter 4 discusses the main concepts of supervised machine learning (ML) by focussing on the application of algorithms for classification purposes with special attention on multi-label problems. The chapter also discusses the importance of confidence estimation in machine learning.

Following this introduction, the thesis reports a series of methods that are aimed at consolidating the reaction vector framework. Chapter 5 presents an ensemble of algorithms for reaction standardisation, validation, and fingerprinting. These algorithms are aimed at providing clean data that can be readily used with the reaction vector methods or for machine learning purposes. The algorithms are then applied to a number of reaction collections from the literature and industry. The chapter also provides the reasons for the creation of a tailored system for reaction classification. Chapter 6 describes the implementation of an automated design tool based on reaction vectors referred to as RENATE. The tool is first validated computationally, then adapted for the design of inhibitors with improved brain penetration for the biological target poly[ADP-ribose] polymerase 1 (PARP1). A number of selected compounds are finally synthesised using the synthetic routes proposed by the structure generator, then further evaluated on their PK properties.

The final chapters are concerned with enhancing reaction vector-based *de novo* design by means of machine learning. Chapter 7 describes the development of a model for reaction classification that can be applied effectively to noisy collections of reaction examples. A prototype model similar to that implemented by Schneider and colleagues (2015) is first validated, then extended and augmented to classify a much higher number of reactions and to output confidence estimations on the individual predictions. The

chapter illustrates the use of the model to analyse the content of reaction datasets. The introduction of reaction classification in *de novo* design aims at enabling the application of sets of transformations by reaction-type. Chapter 8 illustrates the exploitation of this new feature using a model (referred to as a *recommender*) that communicates which reaction classes should be applied to a given molecule during structure generation. The chapter describes the validation of a proof of concept model, then its scaling and adaptation to the reaction vector framework. The chapter also contains three additional validations of the model, with particular attention on its integration in *de novo* design, in order to highlight the benefits and limitations from its use for compound generation.

Finally, Chapter 9 summarises the experiments and results reported in the thesis, then it highlights the limitations of the methods developed in this work, while suggesting possible directions in which the reaction vector method could be improved further.

Sheffield, December 2019

Gian Marco Ghiandoni

Chapter 1: Chemical Representations

1.1. Introduction

For centuries, chemical observations have been communicated through the use of molecular modelling and representation methods. The development of more effective approaches has led to important advancements during the last two centuries, and the last sixty years, in particular, after the introduction of computer techniques for chemical data handling. The history of chemical information is critical to the future of chemistry and its related disciplines, such as drug discovery and material science, especially with the advent of the age of big data. The use of simple and effective methods for chemical information storage, retrieval and analysis, is essential when relationships between many molecular structures and their properties are investigated (Quadrelli, Bareggi and Spiga, 1978). The same principles can be applied when dealing with chemical transformations. This chapter presents an overview of the molecular and reaction representation techniques that have impacted in the field of chemoinformatics, as well as describing some algorithms for database searching, reaction mapping, and reaction classification.

1.2. Molecular Representation

Molecular representation methods generally fall into two categories: linear representations and connection tables (Holm, 1969).

Among several linear representations proposed in the middle of the twentieth century, the Wiswesser Line Notation (WLN) (Wiswesser, 1952) and the Dyson notation (Dyson, 1968) represented the methods that became most popular (Warr, 2011). Nowadays, these methods have been replaced by the SMILES (Simplified Molecular Input Line Entry System) (Weininger, 1988) and InChI (International Chemical Identifier) (Heller *et al.*, 2015), which permit the representation of structural data using standard ASCII (American Standard Code for Information Interchange) characters. Some examples of notations for molecular representation are reported in Figure 1.1.

SMILES strings are the most used format for molecular storage and interchange since they are easy to read or write and are encoded using limited grammar (O’Boyle, 2012). For example, capital letters represent single atoms, lower case letters indicate aromaticity, parentheses indicate different degrees of branching, numberings are used to open and close rings, and hydrogens are explicit only for chiral centres. The SMILES also has an extension called SMARTS (SMiles ARbitrary Target Specification) which allows substructural manipulation using logical operators and wildcard atoms (Warr, 2011). A limitation of SMILES strings is that they can be generated differently according to the method used, although some canonicalisation approaches (e.g. CANGEN (Weininger, Weininger and Weininger, 1989), Universal SMILES (O’Boyle, 2012)) have been proposed over time. A canonical representation corresponds to a unique atom ordering in a given molecule. The use of canonical structures has contributed significantly to increasing the efficiency of database search algorithms.

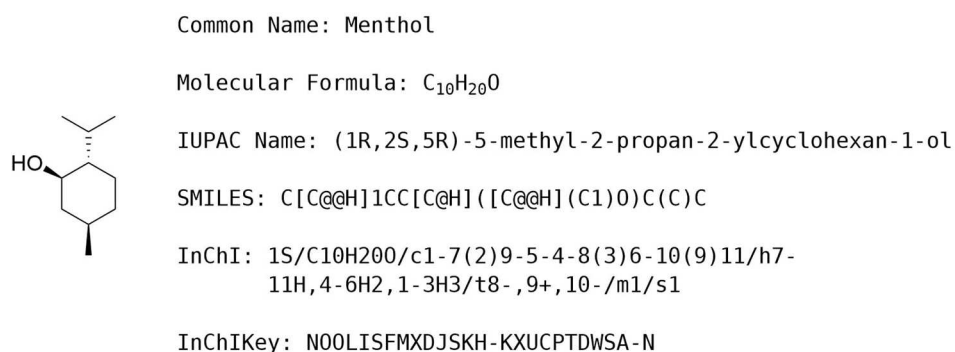


Figure 1.1: Example of different notations for Menthol. The notations ‘@’ and ‘@@’ are used to describe chiral centres.

The InChI, developed by IUPAC (International Union of Pure and Applied Chemistry), aims at providing a unique structural identifier using strings divided in layers that contain chemical metadata such as structure, charge, and stereochemistry. InChI strings are divided into six hierarchical layers representing different types of structural information, where each layer is separated by a slash (‘/’) character. Extra layers can also be added to represent extended molecular contexts such as polymers or reactions. The InChI also has a corresponding compact identifier (27-characters long) named InChIKey (Heller *et al.*, 2015). In addition to being shorter, the InChIKey can

also be used with interfaces, such as web services, that are cannot accept directly special characters. Both InChI and InChIKey are poorly readable by humans and they do not offer any extension to handle substructures.

Structures can be alternatively described using connection tables (Ctabs): The MDL Information Systems (Molecular Design Limited - currently BIOVIA) connection table is the current standard for exchanging chemical data (Dalby *et al.*, 1992). It works by separating atoms and bonds into two distinct blocks then describing their connectivity, coordinates, and properties. An example connection table is reported in Figure 1.2.

```

1  Menthol
2  .mol connection table
3
4  11 11 0 0 1 0 0 0 0 0 0999 V2000
5  -0.3572 0.2062 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6  -0.3572 -0.6187 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7  0.3572 -1.0313 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8  1.0717 -0.6187 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9  1.0717 0.2062 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0.3572 0.6187 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 0.3572 1.4437 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0.3572 -1.8563 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 -0.3572 1.8563 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 1.0717 1.8563 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15 -1.0717 0.6187 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16 1 2 1 0
17 2 3 1 0
18 3 4 1 0
19 4 5 1 0
20 5 6 1 0
21 6 1 1 0
22 6 7 1 6
23 3 8 1 1
24 7 9 1 0
25 7 10 1 0
26 1 11 1 1
27 M END

```

Figure 1.2: Example of connection table for Menthol.

Different versions of this template have been developed over time, leading to a number of file extensions (Warr, 2011): ‘mol’ (Molecule file) describes a single molecule; ‘rgfile’ (RGroup file) is used for a single molecular query with Markush structures; ‘rxnfile’ (Reaction file) describes a single reaction; ‘sdf’ (Structure-Data-file) can describe multiple structures and associated information; ‘rdf’ (Reaction-Data-file) format is similar to ‘sdf’ but can also contain reaction data; ‘xdf’ (XML-Data files) is also similar to ‘sdf’ yet it is based on Extensible Markup Language (XML), which describes a set of rules for encoding data in a format that is both human-readable and machine-readable.

A variation of XML named CML (Chemical Markup Language) has also been proposed as a specific rule schema for chemical information (Murray-Rust and Rzepa, 1999).

Several canonicalisation methods for connection tables, such as the Morgan algorithm (Morgan, 1965), have also been developed to increase the efficiency of search engines. These methods usually attempt the generation of unique atom numberings based upon features such as connectivity, aromaticity, stereochemistry or tautomerism. For example, the Morgan algorithm aims at differentiating atoms by iterative calculations on their connectivity values. The algorithm first assigns a connectivity value to each atom that is equal to its number of atomic connections. Second, each value is updated by summing the neighbours' connectivity values. The procedure continues until every atom has a unique value. At the end of the process, atoms are sorted by descending order based on their final connectivity values and some additional properties such as bond order and atom type.

1.2.1. Molecular Graph Theory

The principle on which molecular connection tables rely is explained by *graph theory*. This area of mathematics has found application in many disciplines, including chemistry, where it is necessary to describe connections between objects. Graphs are abstract structures represented as nodes connected by edges. In a molecular graph, nodes represent atoms with their properties, while edges represent bonds and their orders (Figure 1.3).

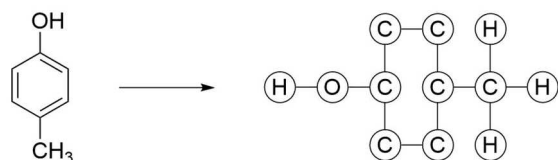


Figure 1.3: Example of a molecular graphic representation (left) converted into a graph (right), in which atoms and bonds are identified by nodes and edges, respectively.

Hydrogens are generally omitted (Leach and Gillet, 2007). However, molecular graphs are not equivalent to traditional graphical representations since they cannot encode effectively properties such as electronic delocalisation or isomeric bonds. Graphs

can also be searched by subgraphs, which are subsets of nodes and edges. This is particularly useful for substructure search in chemical databases.

1.2.2. Molecular Search Methods

Molecular search methods include *structure matching* (or *exact matching*), *substructure searching*, and *similarity search*.

1.2.2.1. Structure Matching

Structure matching is the simplest form of molecular search, where an exact query structure is searched in a database. However, this task can still result to be difficult since identical molecules can be encoded using different atom orderings, this can lead to issues such as duplicate storage or retrieval failure. A potential solution is to test every possible representation for each query and database entry, but this is computationally expensive as for a table of N atoms there are $N!$ combinations to consider. Canonicalisation algorithms narrow down the search using unique atom orderings, which avoid the evaluation of multiple equivalent structures. The search can occur by direct comparison of strings or connection tables, or it can be further speeded up by *data hashing*. Hashing consists of associating queries and entries with new alphanumerical strings according to a given algorithm, for example, the Freeland approach (Freeland *et al.*, 1979). These strings are indicated as *hash keys* and they are used to sort databases in a way that queries can be processed quicker. Nevertheless, sometimes different entries are associated with the same key, leading to a clash during the search. In these cases, dedicated algorithms are applied to resolve the clash by running a more accurate comparison between the query and clashing entries (Leach and Gillet, 2007).

1.2.2.2. Substructure Searching

Substructure searching consists of retrieving all the database structures containing a given substructure query (Figure 1.4). This is typically done by means of graph theory, for which substructure searching is formulated as a *subgraph isomorphism* problem, where database graphs are analysed to determine whether they contain subgraphs that

are isomorphic to the query graph. These problems are known as NP-complete (*nondeterministic polynomial*) which refers to the exponential relationship between problem size and time required to find the solution (Cook, 1971). NP-complete problems are addressed computationally using brute force approaches often combined with heuristics (Englert and Kovács, 2015).

The first computational approach for substructure searching was developed by Ray and Kirsch (1957). Their method consisted of testing the query against all the database graphs, hence resulting in long computational times when applied to real chemical databases (Barnard and Downs, 1992). For this reason, a preliminary screening was introduced to filter out the largest part (e.g. 99%) of database structures that do not have the structural features described in the query (Dittmar *et al.*, 1983).

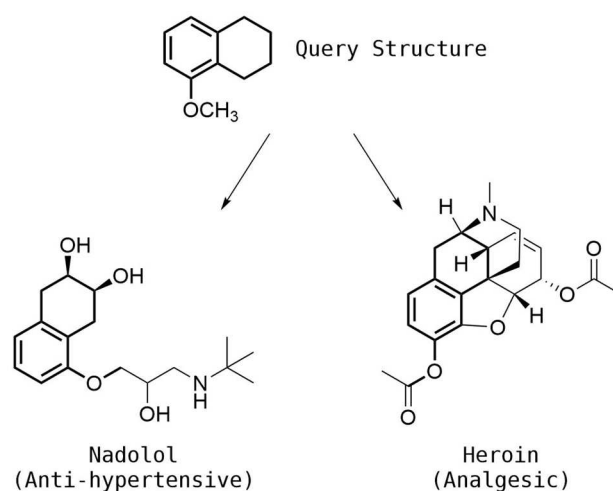


Figure 1.4: Examples of substructure searching where the query substructure (bolded) matches two drug structures, namely Nadolol and Heroin.

The screening relies on the fast comparison between query and database *bit strings* (also known as *bit vectors* or *fingerprints*) (Figure 1.5). An example of a bit string used in this process is a 2D fingerprint, which is a vector that stores bits ('0' and '1') representing substructural features, such as augmented atoms, linear sequences, branches, or rings. These features are often indicated as *structural keys*. Consequently, only a small percentage of structures that are retained after the screening are then searched using substructure search methods.

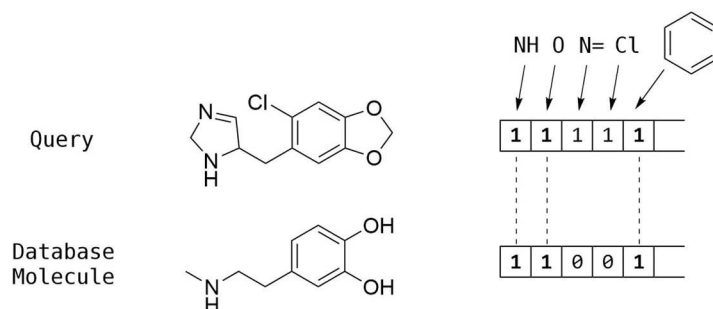


Figure 1.5: Example of bit string comparison prior to substructure search. The example shows that only three out of five query features are matched by the database molecule, hence the entry is discarded.

Note that an accurate selection of *structural keys* is necessary to perform an effective screening since frequent and correlated features tend not to be discriminative and redundant, respectively, whereas a tailored selection of motifs can improve remarkably the speed of search algorithms (Hodes, 1976). Alternatively, *hashed fingerprints* can be used to avoid the selection of a predefined structural dictionary. These fingerprints are defined by the enumeration of molecular linear paths, such as atom sequences or bond sequences, up to a specified number of atoms. *Hashed fingerprints* can be combined with *structural keys* to form hybrid systems, such as the UNITY system (Clark *et al.*, 2000).

1.2.2.3. Similarity Searching

Structure and substructure search methods can only be used when the user knows exactly what query to search for. Also, in substructure searching, if the query is not specific enough, the output can result in a vast number of structures. *Similarity searching* is a technique that provides a solution to these limitations. First, the query structure is used only as a reference for the retrieval of similar compounds, hence it is not necessary to consider a specific substructure. Second, results from a similarity search always come with a numerical score for each compound, hence structures can be sorted on their similarity to the query.

The *similarity property principle* by Johnson and Maggiora (1990), which states that similar structures have similar properties, is the driving force of similarity searching.

The search occurs by comparing compounds on their molecular fingerprints or descriptors. Fingerprints, as introduced previously, are vectors that encode sets of features, while descriptors are numerical values that are computed to characterise molecules. Fingerprints are roughly divided into *bit* and *count* fingerprints. The first encode only the presence of features, while the second also records the number of times each feature appears in a molecule. Many fingerprints have been developed for diverse purposes, for example, to search for compounds with similar structure or functional groups. Many descriptors have also been developed over time, ranging from simple counts such as molecular weight (MW), to complex descriptions derived from quantum mechanics (QM) calculations. Descriptors can be purely computed numbers (e.g. numbers of hydrogen-bond donors (HBD) or acceptors (HBA)) or predicted physicochemical features (e.g. solubility, logP, etc.).

A class of descriptors that is relevant to the purpose of this work is the *atom-pair* (AP), which describes a pair of atoms and their properties using a linear notation. Atom pairs were first introduced by Carhart *et. al.* (1985) according to the following form:

$$\text{ATOM}_1(\text{description})\text{-S-ATOM}_2(\text{description})$$

Equation 1.1: Atom-pair notation introduced by Carhart and colleagues.

Equation 1.1 describes a generic atom-pair where atoms are represented with their properties in brackets (e.g. element type, number of non-hydrogen bonds, number of bonding π electrons), separated by a term ('S') that indicates the length of the shortest path (i.e., number of atoms) between the two atoms (included). For example, the sequence "CX2-(3)-O·X1", describes a carbon (C) bonded to 2 non-hydrogen atoms (X2), connected through a path of 3 atoms (3) to an oxygen (O) bonded to 1 non-hydrogen atom (X1). The dot ('·') indicates the presence of 1 π -electron on the oxygen.

Bit strings can be compared on their similarity or distance. Examples of metrics used for bit string comparison are given in Equation 1.2 and Equation 1.3 for a two vectors A and B. In both equations, a and b are the numbers of set bits ('1') in the vectors A and B, respectively, while c is the number of common set bits between A and

B. Similarity metrics (e.g. Tanimoto) provide a direct measure of similarity, hence they are increased by the presence of common features, while distance metrics (e.g. Hamming) are computed based upon the absence of features (Willett, Barnard and Downs, 1998).

$$T_s = \frac{c}{a+b-c} \quad D_s = \frac{2c}{[a+b]} \quad C_s = \frac{c}{\sqrt{[ab]}}$$

Equation 1.2: Tanimoto (T_s) (or Jaccard) (left), Dice (D_s) (centre), and Cosine (C_s) (right) metrics for molecular similarity.

$$H_d = [a+b-2c] \quad E_d = \sqrt{[a+b-2c]} \quad S_d = 1 - \frac{c}{[a+b-c]}$$

Equation 1.3: Hamming (H_d) (left), Euclidean (E_d) (centre), and Soergel (S_d) (right) metrics for molecular distance.

Count fingerprints or molecular descriptors can also be compared using appropriate distance metrics, such as Euclidean, Manhattan, or Cosine distances (Figure 1.6), which differ from those used with bit strings. These metrics are often used by machine learning algorithms to measure the distance between vectors.

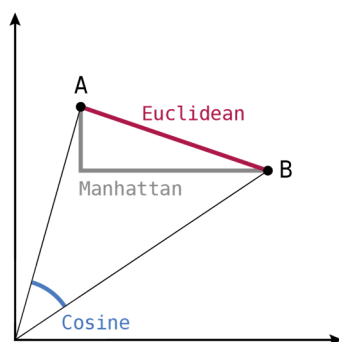


Figure 1.6: Example of distance computation between two vectors A and B using Euclidean, Manhattan, and Cosine metrics.

Euclidean distance represents the shortest distance between the vectors A and B; Manhattan distance is the distance between the projection of the points on the axes; Cosine distance is 1 minus Cosine similarity, which is the cosine of the angle between the points. Euclidean and Manhattan can cover any range of positive values and they both account for vector magnitude, whereas Cosine distance does not and ranges only from 0 to 2, which are derived from one minus the range of values covered by the Cosine

function (-1 to +1). Cosine distance accounts only for non-zero dimensions, thus it can be more appropriate for the computation of the distance between sparse vectors, and it does not represent a proper distance measure since it violates the triangle inequality property. The formulae for the calculation of these metrics are reported in Equation 1.4 for two vectors A and B:

$$d_{\text{Euclidean}} = \sqrt{\sum_{i=1}^n (A_i - B_i)^2} \quad d_{\text{Manhattan}} = \sum_{i=1}^n |A_i - B_i| \quad d_{\text{Cosine}} = 1 - \frac{A \cdot B}{\sqrt{A \cdot A} \sqrt{B \cdot B}}$$

Equation 1.4: Euclidean, Manhattan, and Cosine distance metrics for the computation of the distance between the vectors A and B.

1.3. Reaction Representation

Chemical transformations can also be represented using linear notations or connection tables by applying a few adjustments to cope with the presence of multiple components and roles (i.e., reactant, agent, product).

SMILES can be used for reaction representation by separating reactants, agents (e.g. catalysts), and products with ‘>’ signs; multiple components are separated with ‘.’ characters; atom mapping is expressed by enclosing each atom within parentheses and by assigning a colon ‘:’ with the corresponding numerical tag. An example of a reaction SMILES representation is reported in Figure 1.7.

SMILES strings do not allow the specification of substructure queries; however, reaction queries can be generated using an extension of the SMILES named SMIRKS (Simple Molecular Input Reaction Kinetic String), which is typically used in database searching (see Section 1.5). The SMIRKS relies on five rules for the correct generation of reaction queries: each mapped reactant atom must have its corresponding mapped product atom. Atom mapping is defined by :N, where N is the corresponding numerical tag; stoichiometry is assumed to be 1:1; explicit hydrogens must meet the same conditions on both sides; bond wild cards are not allowed, whereas atom wild cards are; SMARTS must be used to lock portions of structures that are expected not to change,

while SMILES must be used where changes occur. An example of SMIRKS query is reported in Figure 1.8.

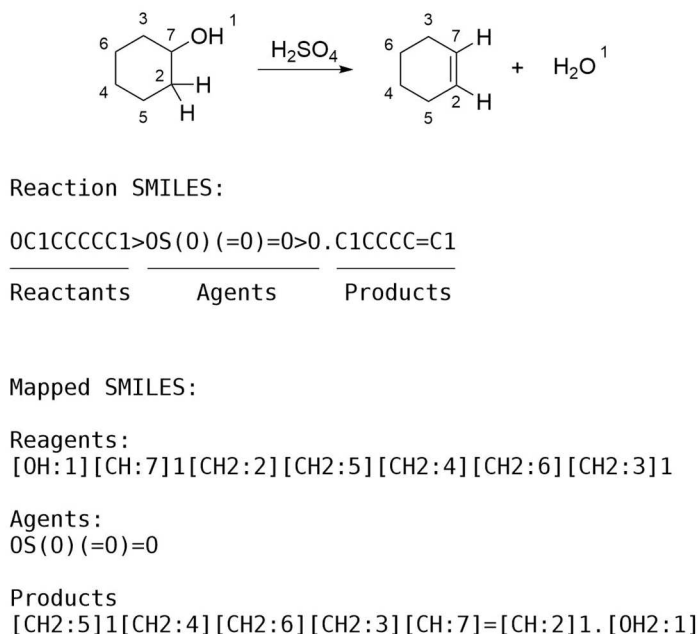


Figure 1.7: Reaction SMILES for an acid-catalysed dehydration.

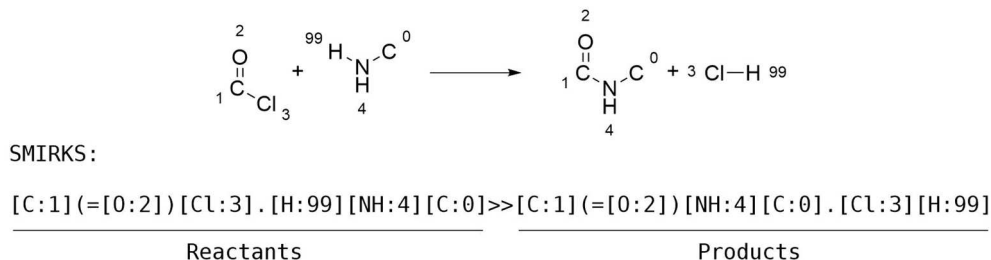


Figure 1.8: SMIRKS notation for an amide formation reaction.

Alternatively, reactions can also be represented using an extension of the InChI named RInChI (and RInChiKey), which is aimed at providing a unique, concise, machine-readable identification for chemical transformations (Grethe *et al.*, 2018).

1.3.1. Reaction Mapping

Reactions can be defined as transformations occurring between initial and final molecular states. The atoms and bonds that change in a given reaction are identified as the *reaction centre* (RC) and their mapping information can be used, for example, for classification purposes or reaction mechanism elucidation. Reaction mapping consists of

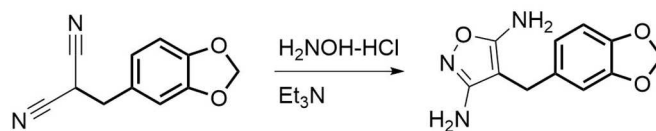
numbering which reactant atoms become product atoms. This information can be generated using *in silico* methods provided that the input describes balanced (i.e., stoichiometric 1:1) single-step reactions. New techniques that do not necessarily require stoichiometric conditions have also recently been proposed (Jaworski *et al.*, 2019). The current approaches for reaction mapping fall into two major categories: *common substructure-based* and *optimisation-based* approaches (Chen, Chen and Taylor, 2013).

The first automated reaction mapping algorithms consisted of comparing reactants and products written in Wiswesser Line Notation by fragmenting the components, eliminating the unchanging fragments, then reassembling, and comparing the retained structures (Harrison and Lynch, 1970) (Lynch and Willett, 1978). These methods were based on the concept of *extended connectivity* (EC) that is exploited, for example, by the Morgan algorithm, to assign a unique numbering to each atom according to its chemical neighbourhood.

More recent mapping algorithms typically involve *Maximum Common Substructure* (MCS) matching between the two sides (Vléduts, 1977) (McGregor and Willett, 1981) (Funatsu and Sasaki, 1988) (Arita, 2003) (Kumar and Maranas, 2014) (Rahman *et al.*, 2016). MCS-based methods rely on the molecular graph theory in order to determine the maximum common substructure between reactant and product sides. The concept of MCS is illustrated in Figure 1.9. Further developments of the MCS-based methods incorporate some chemical knowledge into the mapping process by weighting bonds according to the atoms they are associated with. For example, a weight of 1.5 is assigned for C-C σ -bonds, 0.48 for C-N_{amine}, C-O_{ester}, and C-S_{thioester} bonds, and 1 for all other bonds (Apostolakis *et al.*, 2008). These weights reflect the likelihood of a bond being broken in a transformation, with lower weights corresponding to easier breakage.

A different class of mapping algorithms involve the use of search strategies. These algorithms explore the problem domain in the effort of finding mappings that describe the shortest paths, in terms of number of the bonds broken and/or formed, between reactants and products (Akutsu, 2004) (Crabtree and Mehta, 2009) (Heinonen *et al.*, 2011) (First, Gounaris and Floudas, 2012) (Latendresse *et al.*, 2012) (Litsa *et al.*, 2019).

This mechanistic assumption relies on the so-called *principle of minimal chemical distance* (PMCD) (Jochum, Gasteiger and Ugi, 1980). Some of these algorithms have also shown better accuracy and performance than the MCS-based methods (Chen, Chen and Taylor, 2013).



Examples of common substructures:

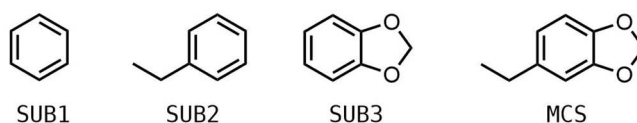


Figure 1.9: Examples of common substructures in a reaction. The MCS corresponds to the largest common substructure that is shared between reactant and product.

Methodologies that combine characteristics of both common substructure and optimisation approaches, as well as techniques that rely only on pure chemical heuristics, have also been proposed (Kraut *et al.*, 2013) (Fooshee, Andronico and Baldi, 2013) (Jaworski *et al.*, 2019).

1.4. Reaction Databases

Reaction databases are mainly divided into two content-based categories (Boiten, Ott and Noordik, 1995): Comprehensive literature contents within specific boundaries (e.g. CASREACT (Chemical Abstract Service, 1988), Reaxys (Elsevier, 2009)) and useful reaction collections without any claim of completeness (e.g. REACCS (Willett, 1986), ORAC (Miller *et al.*, 1994), SYNLIB (Chodosh *et al.*, 2010)). Databases can also be divided into *free* and *commercial*.

Database entries typically describe reactions with some additional information such as solvents, catalysts, experimental conditions, yields, and literature citations (Patel *et al.*, 2009). Entries are usually indexed to enable rapid retrieval (Hendrickson and Miller,

1990). Examples of some organic reaction databases are given in Table 1.1 and Table 1.2:

Database	Source	Reactions
Organic Syntheses (Organic Syntheses Inc., 1921)	Collection of repeated and validated procedures.	N.A.
SynArchive (Synarchive S.E.N.C., 2011)	Collection of indexed reactions and syntheses for known molecules.	~5,000
ChemSpider (Royal Society of Chemistry, 2008)	Comprehensive collection of compounds and reactions from ~500 data sources.	~275,000
WebReactions (openmolecules.org, 1999)	Partial content from ChemReact database.	~400,000

Table 1.1: A list of free organic reaction databases.

Database	Source	Reactions
e-EROS (John Wiley and Sons, 1999)	Submissions to editor	~70,000
ChemReact (InfoChem, 1996)	Unique reaction types taken from SPRESI data collection	~500,000
SPRESI (InfoChem, 1974)	Journals and patents	~4,500,000
Reaxys (Elsevier, 2009)	Journals and patents	~42,000,000
CASREACT (Chemical Abstract Service, 1988)	Journals and patents	~86,000,000

Table 1.2: A list of commercial organic reaction databases.

1.5. Reaction Search Methods

Reaction databases can be browsed using search methods for individual molecules, for example, by setting a specific reactant to start with or a specific product to obtain. However, these methods cannot be used to search for generic transformations or reactions that preserve particular substructures. This type of search can be done by specifying the reaction centre as a query using some auxiliary information, such as breaking/forming bonds and atom mapping. As introduced in Section 1.3, this can be achieved using the SMIRKS notation, or in some cases, database interfaces are customised to allow the specification of reaction centres. For example, CASREACT (Chemical Abstract Service, 1988) has a list of options and indicators to assign reaction

roles, map atoms in reactants and products, mark bonds to be broken or formed, and lock atoms/rings to protect them from being transformed (Figure 1.10).

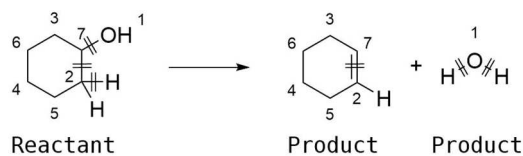


Figure 1.10: Example of a reaction query for an acid-catalysed dehydration on CASREACT.

1.6. Reaction Classification

Chemical compounds are unambiguously identified using the international systematic nomenclature and classification methods developed by IUPAC (Favre and Powell, 2014). However, such rules have not been extended for chemical transformations, hence traditional names or short descriptions are used instead. These schemes include the Merck Index (Stecher, 1960), the hierarchies of Carey and colleagues (2006), Roughley and Jordan (2011), and the formal RXNO ontology developed by the Royal Society of Chemistry (RSC).

More systematic methods have been proposed in the previous decades with the aim of improving searching and knowledge exploitation, for example, for efficient data retrieval or automatic identification of relationships between classes. These approaches can be mainly divided into *model-driven* and *data-driven* (Chen, 2003). The development of many reaction classification methods over time is due to the more complex nature of chemical transformations compared to individual molecules, which results in a greater number of ways in which reactions can be investigated.

1.6.1. Model-driven Methods

Model-driven methods classify reactions by means of pre-defined schemes that are usually hierarchical (Kraut *et al.*, 2013). Only a few general classes describe essential transformations, then multiple levels of subclasses are used to further discriminate the

entries. Most of these approaches rely on mapping information and reaction mechanism, hence they are capable to group together reactions that are mechanistically similar.

Theilheimer (Becker, 1961) proposed the earliest classification method which relied on the use of symbolic notations. He determined four types of fundamental reactions: addition (\Downarrow), elimination (\Uparrow), rearrangement (\curvearrowright), and exchange (\Updownarrow). The classification consists of describing bond formed, reaction type, and bond broken. For example, the notation $CC\Downarrow CX$ describes the formation of a carbon-carbon bond from a halide. Balaban (1967) proposed an early classification system for cyclic reactions by showing how six-atom pericyclic transformations could be described by the shift of six electrons on the atoms. Hendrickson (1974) extended Balaban's method to the four-atom pericyclics and introduced the concepts of homovalent and ambivalent reactions to describe pericyclics with an odd number of atoms.

Vléduts (1963) developed an advanced approach which relied on the identification of the reaction centre, by marking breaking and making bonds with strokes and arrowheads, respectively, while unchanging bonds were described using plain lines. He also introduced the concept of *superimposed reaction skeleton graph* (SRSG) to describe the reaction centre and its proximal environment: The $SRSG_0$ consists of a generic reaction graph which can be superimposed on the actual reaction centre, and its extensions include proximal unchanging atoms at different levels (e.g. $SRSG_1$, $SRSG_2$, etc.). A similar approach was proposed by Fujita (1986), who coined the concept of *imaginary transition state* (ITS) to embed all reaction features into a single representation by the superimposition of reactant and product states. Hendrickson (1997) eventually proposed a unification of the methods by Vléduts and Fujita (Figure 1.11) by simplifying the description of the reaction centre using solid and dashed lines for broken and forming bonds, respectively. Fujita also proposed a new hierarchy on three levels: the *basic reaction graph* (BRG) as the most abstract level (i.e., reaction template), the *reaction graph* (RG) with shell bonds, and the fully defined *reaction centre graph* (RCG) which also includes atom-types. Several years later, Varnek and colleagues (2005) applied the same concepts to develop a substructural fingerprint for

similarity searching and QSAR, and De Luca and colleagues (2012) formulated an analogue concept indicated as *Condensed Graphs of Reaction* (CGR), for similarity searching and classification.

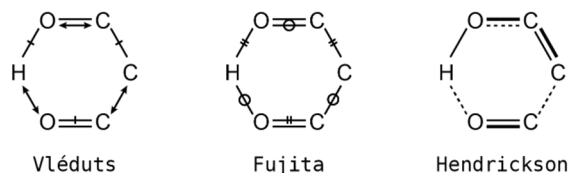


Figure 1.11: Vléduts, Fujita, and Hendrickson reaction representations for a six-atom reaction centre. Image readapted (Chen, 2003).

Zefirov (1980) (1987) also adopted an approach based on the concept of the reaction centre. In his method, reactions are described as complete reactions, then as *reduced systems*, by removing unchanging bonds and using abstract symbols to describe the *bond redistribution*. The procedure was applied to generate the so-called *symbolic equations* (SEQs) which represent basic reaction templates. Some years later, Zefirov and colleagues developed SYMBEQ (Zefirov, Baskin and Palyulin, 1994), a software for systematic classification and computer-assisted molecule design. SYMBEQ reflects the implementation of SEQs and related subconcepts such as reaction centres and bond redistribution. Zefirov (1998) also proposed a five-level classification scheme for organic interconversions.

Several years before Zefirov’s SEQs, Ugi and colleagues (1973) proposed a more abstract method based on the description of chemical reactions with matrices, which could also be manipulated by computers. In their approach, a given reaction is described by adding a reactant connectivity matrix (*B* as “beginning”) to a reaction matrix (*R* as “reaction”) which results in a product connectivity matrix (*E* as “end”). This process is described by the general equation: $B+R=E$. Ugi’s method was applied to several purposes, including synthesis design and reaction simulation by the EROS system (Elaboration of Reactions for Organic Synthesis) (Gasteiger *et al.*, 1987) (Gasteiger, Ihlenfeldt and Röse, 2010), reaction discovery (Bauer *et al.*, 1985) (Herges and Hooek, 1992) (Herges, 1994), and efficient reaction classification (Ram and Pal, 2012). Arens

(1979) independently developed another abstract method based on sequences of numbers and mathematical operators. In his approach, reactant atoms are described according to their bond multiplicity (e.g. an atom with two bonds is indicated as ‘2’), while transformations are described by sequences of $-$ and $+$ to indicate whether bonds are lost or gained, respectively. Arens’s operators are indicated as *reaction keys*, and their use was proposed for reaction classification and prediction.

A more recent computer-manipulable system was proposed by Hendrickson (1995), which was also integrated into the COGNOS software for database retrieval. Hendrickson’s approach consists of generalising all the atoms not included in reaction centres by their relative electronegativity to increase the efficiency of the algorithm.

Hendrickson (2010) also provided a unique, definitive, and universal, reaction signature, which joins and simplifies many of the representation methods proposed in the past. Nowadays, model-driven algorithms still often rely on reaction centres and mapping. For instance, the REACCS database search algorithm combines atom mapping with a substructure fingerprint approach, similarly to some database searching methods (Mooch *et al.*, 1988) (Grethe and Mooch, 1990).

1.6.2. Data-driven Methods

In data-driven methods, the nature of reaction datasets themselves determines which entries will be grouped together and which not. These methods are also indicated as “genuine knowledge discovery systems” (Chen, 2003). Many data-driven methods rely on the use of topological features, hence they cannot account for reactions that are mechanistically similar but structurally different (Kraut *et al.*, 2013). The number of data-driven approaches has significantly increased in the last years possibly due to the availability of freely accessible collections and more powerful computational tools.

Wilcox and Levinson (1986) made an early attempt to design systems capable of encoding structures and reactions with the aim of creating generalisations based on the input reaction set. These generalisations were then used to organise the data and to find solutions by considering two types of reaction network: the *minimum reaction concept*

(MXC), which represents only changing bonds; and the *complete reaction concept* (CXC), which describes an extension of the MXC by adding adjacent shell bonds.

Gelernter and colleagues developed SYNCHEM software (Gelernter, Rose and Chen, 1990), which adopts a conceptual clustering technique that deals with reaction centres similarly to the Wilcox’s MXC model. Their model represents reactants and products as a single *active concept*, whereas adjacent non-changing functionalities are indicated as the *reaction context*. In the same period, Blurock (1990) adopted a similar method that was integrated into the RETROSYN program, which is capable of extracting automatically an analogue of the reaction centre (*reaction pattern*), by summing reactant and product structural information. As a result, he identified distinct reaction classes with their related hierarchies and reaction centres. Unlike Wilcox’s and Gelernter’s methods, Blurock treated reactants and products separately.

A few years later, Rose and Gasteiger (1994) developed HORACE. Their methodology relies on both physicochemical and topological features in order to account for reactions that are structurally different but mechanistically similar. In the latest version of HORACE, physicochemical features such as charge distribution, inductive effect, and resonance effect are calculated for both reaction centre and proximal atoms using empirical methods (Gasteiger *et al.*, 1992). Features are first used to perform a physicochemical-based clustering, then topological features such as functional groups, are used iteratively to further discriminate the entries.

Chen and Gasteiger (1996) consequently developed an unsupervised classification technique based on the Kohonen neural networks, to enable the identification of the inter-relationships between different classes. These networks are used to reduce high-dimensional data into two-dimensional *self-organising-maps* (SOMs), which can be promptly visualised. In their study, Chen and Gasteiger represented each reaction as a vector containing calculated physicochemical features, which was mapped by the algorithm to the most suitable group, producing an effective clustering. Later, Satoh and colleagues (2000), and Zhang and Aires-de-Sousa (2005), reported works similar to the

SOMs presented by Chen and Gasteiger. Latino and Aires-de-Sousa (2006) also applied a similar approach for the classification of metabolic reactions.

Sello and Termini (1997) proposed a three-level similarity-based method for reaction classification using calculated property descriptors. The first level describes general classes such as additions or eliminations, according to a form of calculated chemical potential, while the second and third levels are determined from reacting atoms and atomic classes, respectively. Results showed that their method could be applied effectively for reaction prediction and synthesis planning, yet it required more generalisation for classification and database retrieval purposes. Their work was further continued by Sello (1998) who introduced new roles to account for both steric and electronic constraints.

A more recent structure-topology-based method is CLASSIFY, which is devised by InfoChem and used by several commercial databases (Eiblmaier *et al.*, 2002) (Kraut *et al.*, 2013). CLASSIFY is based on the InfoChem's *reaction centre perception* (RCP) algorithm, which analyses reaction centres at different levels (spheres) by including increasing neighbour-atom information (Figure 1.12). The algorithm works on three levels which are also hashed for rapid retrieval purposes: broad (reaction centre only); medium (adding σ proximal atoms); and narrow (adding σ and β proximal atoms). Generally, the more extended is the reaction centre, the more specific is the hashcode. Hashcodes take into account several atom properties such as atom type, valence state, total number of bonded hydrogens, aromaticity, number of π electrons, formal charge, and bond typology. Reaction hashcodes are generalised into *ClassCodes* for classification purposes. Two important drawbacks of this approach are the lack of stereochemistry and the large number of hashcodes generated.

The *reaction vector* (RV) approach was introduced by Broughton and colleagues (2003) as a structure-topology-based reaction classification method. The RV concept is central to the purpose of this study and it consists of determining a difference vector between products and reactants. This method is reviewed in Chapter 3, which focuses on its implementation for *de novo* design. Before Broughton and colleagues'

formalisation, the original idea of a reaction vector was conceived by Vléduts (1963), who relied on the assumption that in a given transformation, the reaction centre can be identified by tracking which atoms and bonds are subjected to changes, while all the other components remain the same (Willett, 1980). This assumption, under certain circumstances, enables the automatic extraction of reaction centres without the use of mapping information. For example, when atom-pair descriptors are used and reactions are balanced (see Chapter 3), reactants and products can be independently encoded, then their subtraction produces reaction vectors. Reaction vectors described using a structured format are often referred to as *reaction* or *difference fingerprints* (Daylight Chemical Information Systems Inc., 2019).

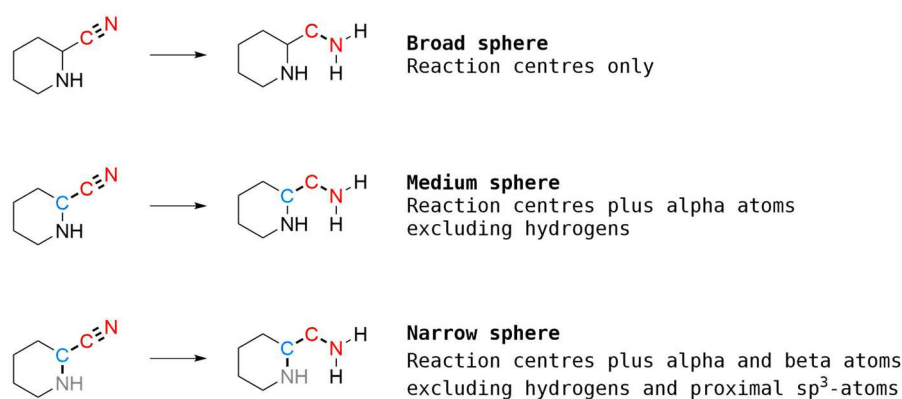


Figure 1.12: InfoChem’s CLASSIFY algorithm: Broad, medium, and narrow spheres describe increasing levels of inclusion of atoms in the reaction centre and its proximal environment. Image readapted (Kraut *et al.*, 2013).

In Broughton and colleagues’ work (2003), different descriptor types were investigated using the reaction vector method and their performance was compared in classification tasks. Ridder and Wagener (2008) described a method for metabolite prediction using a difference fingerprint based on Sybyl descriptors and augmented atom types, also by including some proximal atom information. Later, Hu and colleagues (2012) used difference fingerprints to assign enzymes and enzyme genes identifiers to biochemical reactions, and Schneider and colleagues (2015) developed a novel fingerprint for reaction classification, which also included information on components not described in the reaction centre, such as solvents, ions, or catalysts. Their method was tested on

a subset of 50 reaction classes extracted from the US pharmaceutical patent literature (NextMove Software, 2014).

More recent works describe the use of Deep Neural Networks (DNNs) to capture relationships between reactions. Schwaller and colleagues (2019) proposed a method where a much bigger dataset of reactions and classes from the US patents than that used by Schneider and colleagues (2015), was used to train an attention-based neural network solely on the base of their SMILES representations and annotated classes. The inspection of the model weights revealed that the algorithm identified the SMILES patterns responsible for describing the transformations. In the same period, Baylon and colleagues (2019) proposed the use of neural networks to cluster similar reactions together in order to support retrosynthesis prediction. In their approach, a given retrosynthetic task is solved by first identifying which reaction cluster is more likely to produce the query product, then, by ranking the entries in the cluster on their probability to identify specific reaction rules.

1.7. Conclusions

In this chapter, an introduction to the main approaches that have been used for the representation of molecules and reactions has been reported. The first part of the chapter has focused on the principles of graph theory, which have often been applied to convert molecule drawings into machine-readable representations, as well as describing some methods used for molecular searching, in particular when dealing with databases. The second part has discussed the importance and evolution of mapping algorithms and their role in reaction database searching and classification. Finally, the distinction between the main approaches used for reaction classification and some examples of their application has been given. The next chapter discusses the aim of molecular *de novo* design and the strategies that have been developed over time to enable an effective and efficient search of the chemical space.

Chapter 2: De novo Molecular Design

2.1. Introduction

Molecular *de novo* design aims at generating a limited number of compounds that meet specific criteria such as polypharmacological and safety profiles, synthetic accessibility, and novelty to secure intellectual property rights for commercial purposes (Hartenfeller and Schneider, 2011). The first *de novo* design programs appeared in the late 1980s with the rise of protein modelling, which allowed the analysis of the interactions between small-molecules and binding pockets. From there, more sophisticated programs have been proposed over time, with the aim of accounting explicitly for chemical knowledge and synthetic accessibility. However, due to the general complexity of drug design, the most recent techniques have been relying more on implicit modelling by means of data and machine learning to fulfil the criteria necessary for successful molecular design. This chapter introduces the basic principles of *de novo* design and reviews the components exploited by design algorithms, providing an overview of the evolution of this discipline.

2.2. The Molecular Design Route

The main goal of *de novo* design is to generate novel chemical series for biochemical testing while avoiding the systematic investigation of large numbers of compounds. From a more mathematical perspective, this discipline attempts to map molecular structures to physicochemical and biological properties for the rational design of compounds with desired characteristics (Schneider and Baringhaus, 2013). However, over the years this problem has turned out to be more challenging than expected for a number of reasons.

First, the chemical space is vast, with the number of potentially accessible organic compounds estimated somewhere between 10^{20} and 10^{60} (Bohacek, McMartin and Guida, 1996) (Ramström and Lehn, 2002) (Ertl, 2003) (Polishchuk, Madzhidov and Varnek, 2013). The interpretation of these values can be attempted by considering that the number of atoms in the Solar System has been estimated at 10^{54} (Mullard, 2017). Thus,

a systematic approach for the search of new small-molecule drugs is not an applicable strategy.

Second, the chemical space is regulated by numerous parameters that eventually determine the properties of compounds. Many of these parameters need to be evaluated and balanced accurately in order to obtain effective drugs that are also synthesisable in reality. In addition to this, alterations of molecular structures often do not produce a linear response, hence resulting in a difficult sampling and exploration of the search space. Due to this reason, the local optimisation of compounds can be performed effectively only within areas where a smooth response (or *Neighbourhood Behaviour*) to local changes is observed (Schneider and Baringhaus, 2013). These regions obey the *chemical similarity principle* coined by Johnson and Maggiora (1990), which states that molecules sharing similar structures, will also have similar properties. These areas of chemical space can also be visualised as smooth fitness landscapes where molecular features and properties (e.g. activity) are related to each other (Figure 2.1).

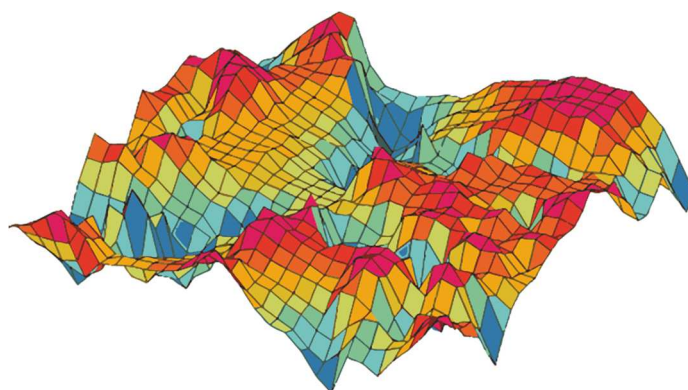


Figure 2.1: Chemical space activity landscape. The axes lying on the plane describe molecular features while the vertical axis represents activity. Areas where the activity is lower and higher are also indicated by cold and hot colours, respectively.

De novo design often relies on strategies for local compound optimisation rather than searching for global optima (Schneider and Baringhaus, 2007). This can be done effectively by means of scoring methods capable of modelling accurately the underlying fitness landscape. Due to the characteristics of the search space, *de novo* design programs usually consist of multiple components that work in synergy to form an iterative cycle

where molecules are generated, evaluated, then the results are used to drive the design of better compounds.

Virtual *de novo* design is also typically coupled with complementary approaches for drug discovery, such as virtual or physical compound screenings (Hartenfeller and Schneider, 2011). Nevertheless, these frameworks are far from being fully automated. Human decision making still plays an important role due to the general complexity of drug design problems and the lack of synthetic accessibility (SA) of compounds (Gasteiger, 2007). Due to these issues, the latest approaches attempt to incorporate implicitly as much chemical knowledge as possible into the design algorithms.

2.3. De novo Design Components

Molecular *de novo* design programs are built in order to generate a list of rationally designed structures for a given reference, which can usually be either a protein or a ligand. These programs exploit three main components that work in synergy to reduce computational times and increase the chance of finding valid candidates: *scoring* components, which are aimed at ranking structures according to one or multiple constraints; *assembly* modules, which determine how molecules are generated, for example, by combining atoms or fragments according to certain criteria to satisfy rules of chemical valence, or to account for synthetic accessibility; *search* strategies, which drive the compound searching and optimisation with the aim of exploring effectively and efficiently the chemical space.

2.4. Scoring Components

Scoring approaches can be mainly divided into *structure-based* and *ligand-based*. The first *de novo* design algorithms consisted of assessing the quality of the interaction of a compound with the receptor binding site (i.e., structure-based); hence, their application was limited to drug discovery problems for which three-dimensional target data was available. However, although tremendous progress has been made in the field of protein crystallography over the last forty years, many targets of pharmaceutical

interest still have limited or even no structural data from experiments to date (McPherson and Gavira, 2014). In addition, the computational demands of structure-based design software were often too high to be satisfied with the technologies of a few decades ago. Due to these reasons, ligand-based approaches were introduced to assess the similarity of designed structures to known active ligands. Nowadays, multiple scoring components are often deployed in *de novo* design with the aim of evaluating simultaneously several criteria such as activity, pharmacokinetics, and toxicity. The evaluation of multiple criteria during the generation of novel compounds is referred to as multi-objective *de novo* design (Schneider, 2002) (Nicolaou and Brown, 2013).

2.4.1. Structure-based Scoring

Early *de novo* design software, also referred to as *receptor-based* (e.g. HSITE/2D skeletons (Danziger and Dean, 1989), LUDI (Böhm, 1992a) (Böhm, 1992b), CONCEPTS (Pearlman and Murcko, 1993)), relied entirely on *structure-based* scoring. The aim of this approach is to maximise the complementarity between ligands and a binding site by taking into account steric and the electronic properties.

Scoring-based methods are closely related to virtual techniques for compound screening such as docking, which are used for the estimation of protein-ligand interactions. These techniques usually consider the protein structure as static. Three main scoring approaches have been established: *force-field*, *empirical*, and *knowledge-based* scoring functions. Some programs (e.g. X-Cscore (Wang, Lai and Wang, 2002)) use multiple scoring methods at the same time to produce a so-called *consensus scoring*.

Force-field (FF) functions rely on individual atomic interactions derived from quantum mechanical calculations and sometimes experimental data (Srinivas Reddy, Chen and Zhang, 2013). These functions sum up the contribution of several components such as electrostatic (i.e., Coulomb) and van der Waals interactions, hydrogen bonds, and conformational constraints, such as bond stretching or bending. Entropic and solvation terms are typically not accounted for these methods.

The first structure-based *de novo* design program using force fields was LEGEND (Nishibata and Itai, 1991), which implemented only a simple scoring component to select which ligand fragments to grow within the binding site. Later, other programs such as CONCERTS (Pearlman and Murcko, 1996), RASSE (Luo, Wang and Lai, 1996), GANDI (Dey and Caflisch, 2008), and Fragment Shuffling (Nisius and Rester, 2009) also implemented FF-based functions.

Empirical functions approximate binding energies by computing the weighted sum of uncorrelated terms such as interactions (e.g. Coulomb, hydrogen bonding, etc.), desolvation, entropy, and hydrophobicity (Chen, Yin and MacKerell, 2002) (Mackerell, 2004). Weights are derived, for example, by regression analysis of experimental binding energies of known protein-ligand complexes, then penalty terms, such as the number of rotatable bonds in ligands, are usually introduced as corrective factors. Empirical functions are typically much faster than force-field methods since they are based on simple energy terms; however, their domain of applicability is restricted due to their empirical formulation (Srinivas Reddy, Chen and Zhang, 2013). The first program implementing an empirical scoring function was LUDI (Böhm, 1992b), yet many others have appeared over time (e.g. CONCEPTS (Pearlman and Murcko, 1993), GrowMol (Bohacek *et al.*, 1999), PRO_SELECT (Murray *et al.*, 1997) (Liebeschuetz *et al.*, 2002), LigBuilder (Wang, Gao and Lai, 2000) (Yuan, Pei and Lai, 2011), FlexNovo (Degen and Rarey, 2006)).

Knowledge-based functions are modelled on the statistical differences between experimental and theoretical interactions of atom pairs (i.e., couple of interacting atoms) in protein-ligand complexes. These complexes, for which experimental data is available, are used as a training set for the derivation of potentials that are eventually transformed into interaction scores; hence, these functions also have a limited domain of applicability due to their experimental formulation. The basic principle of knowledge-based scoring relies on the probabilistic assumptions of the Boltzmann equation (Zhang, Golbraikh and Tropsha, 2006). As a result, atom pairs with high and low occurrences in experimental structures are associated with negative (attraction) and positive

(repulsion) scores, respectively, and the sum of all individual scores yields the global interaction score. Similar to empirical methods, knowledge-based functions attempt to capture the nature of binding avoiding explicit modelling. These approaches were introduced into drug discovery later than force-field and empirical methods. Examples of programs implementing knowledge-based functions are SMOG (DeWitte and Shakhnovich, 1996) (Ishchenko and Shakhnovich, 2002) and GeometryFit (Wang *et al.*, 2010), while some examples of knowledge-based functions are VALIDATE (Head *et al.*, 1996), BLEEP (Mitchell *et al.*, 1999), and DrugScore (Gohlke, Hendlich and Klebe, 2000).

Another way to exploit the receptor information is to create a pharmacophoric model from the spatial arrangement of interaction centres between the binding site and ligand. Interaction points are converted into *hotspots* that are used as constraints for the design of complementary ligands. Examples of programs implementing pharmacophore-based approaches are NEWLEAD (Tschinke and Cohen, 1993), SPLICE (Ho and Marshall, 1993), and SkelGen (Todorov and Dean, 1997) (Dean *et al.*, 2006). A more advanced example is PhDD (Huang, Li and Yang, 2010), which first enumerates a set of ligands that satisfy some specified pharmacophoric constraints, then refines the selection of compounds by performing a multi-objective scoring upon predicted activity, pharmacokinetic properties and synthetic accessibility.

2.4.2. Ligand-based Scoring

Ligand-based scoring programs (e.g. PRO-LIGAND (Clark *et al.*, 1995), LEA (Douguet, Thoreau and Grassy, 2000), TOPAS (Rasmussen and Williams, 2005), BREED (Pierce, Rao and Bemis, 2004), Flux (Fechner and Schneider, 2006) (Fechner and Schneider, 2007), SQUIRREL (Proschak *et al.*, 2009)) were introduced as complementary approaches and to overcome the limitations of structure-based methods. Ligand-based functions assess the similarity (or the distance) of designed candidates to known reference ligands. The principles used to compute similarity indexes are similar to those described in Section 1.2.2.3, which imply the selection of representative molecular descriptors and similarity metrics in order to perform an effective comparison

between molecules. Ligand-based scoring is also often applied for the validation of *de novo* design methods since reference molecules can be used to quantify directly the performance of algorithms (Stahl *et al.*, 2002) (Zaliani *et al.*, 2009). In addition, a great advantage of these scoring methods is that they typically work with two-dimensional structures, which permits their application when the active ligand conformation is not known as well as reducing drastically the computational costs.

As for the receptor information, active ligands can also be used to generate pharmacophores (Schneider and Fechner, 2005), which have found application in a number of studies, such as molecular field analysis (MFA) (Waszkowycz *et al.*, 1994), analysis of structural motifs (Schneider *et al.*, 2000), or pseudoreceptor modelling (Lloyd *et al.*, 2004). A pseudoreceptor is derived from one, or an ensemble of, ligands that are assumed to adopt bioactive conformations. An example of software that accepts pseudoreceptors as pharmacophoric constraints is Skelgen (Todorov and Dean, 1997).

An approach for exploiting ligand information for scoring purposes that has become a *modus operandi* in computer-aided drug discovery - due to the explosion of data sharing and more powerful computation - is by means of supervised machine learning algorithms (Roy, 2017) (Lo *et al.*, 2018). These methods are aimed at modelling the quantitative structure-activity relationship (QSAR) between ligands and targets, or the structure-property relationship (QSPR) of ligands against parameters such as physicochemical properties (e.g. solubility), pharmacokinetics, or toxicity. In *de novo* design, compounds are typically first generated, then scored using QSAR and/or QSPR to generate a ranking. Modern *de novo* design tools often combine multiple models for the prediction of polypharmacological and multi-property profiles (Besnard *et al.*, 2012) (Schneider, 2014).

From a different angle, QSAR can also be used to map properties backwards to the molecular descriptor space. These approaches are indicated as *inverse*-QSAR and they are aimed at identifying promising regions of the chemical space for a given problem (e.g. determining a set of molecular substructures that can lead to bioactivity), rather than producing a forward scoring on compounds generated by a design algorithm. The

major issue with inverse-QSAR is the selection of appropriate descriptors for effective modelling and molecular reconstruction. Some examples of *de novo* design methods based on inverse-QSAR have been proposed by Churchwell *et al.* (2004), Wong and Burkowski (2009), Miyao *et al.* (2010) (2016), Mishima *et al.* (2014), and Takeda *et al.* (2016). An advanced method that combines both forward and inverse-QSAR for the generation of SMILES strings with desired properties has been proposed by Ikebata *et al.* (2017).

2.5. Construction Components

Construction techniques can be mainly divided into *atom-based* and *fragment-based*. These methods can also be reclassified into *growing* and *linking* subcategories. Growing approaches consist of taking an atom or a fragment that is considered essential for the binding with the receptor, then adding fragments or atoms iteratively to increase the overall binding affinity. Linking methods involve placing a number of key fragments at different positions in the pocket, then connecting them together by means of linkers. Construction-based strategies are also typically concerned with accounting explicitly for synthetic accessibility.

2.5.1. Atom-based Construction

In *atom-based* methods, atoms are added one by one according to the binding site conformation (e.g. LEGEND (Nishibata and Itai, 1991), CONCEPTS (Pearlman and Murcko, 1993), GenStar (Rotstein and Murcko, 1993a), RASSE (Luo, Wang and Lai, 1996)) offering fine-grained molecule design with the possibility to access the entire chemical space. However, the simplicity of most atom-based approaches soon turned out to be ineffective for *de novo* design purposes due to the prohibitive number of solutions generated, and the abundance of chemically meaningless or inaccessible structures. Although this issue can be partially solved by applying substructure filters (e.g. the program PhDD (Huang, Li and Yang, 2010) implements a set of rules for drug-likeness and synthetic accessibility), the last true *atom-based* software, RASSE (Luo, Wang and Lai, 1996), was released more than twenty years ago.

More advanced atom-based approaches have gained some popularity in very recent years due to the emergence of deep learning generative models that rely on SMILES strings or molecular graphs (Olivecrona *et al.*, 2017) (Li, Zhang and Liu, 2018). These methods are also reviewed in Section 2.7.

2.5.2. Fragment-based Construction

In *fragment-based* methods, compounds are designed by means of fragment libraries and rules for the generation of virtual bonds (e.g. LUDI (Nishibata and Itai, 1991), NEWLEAD (Tschinke and Cohen, 1993), GroupBuild (Rotstein and Murcko, 1993b), MCSS (Caflisch, Miranker and Karplus, 1993), SPROUT (Gillet *et al.*, 1993) (Gillet *et al.*, 1995), HOOK (Eisen *et al.*, 1994), MCDNLG (Gehlhaar *et al.*, 1995), Chemical Genesis (Glen and Payne, 1995), PRO-LIGAND (Clark *et al.*, 1995), F-DycoBlock (Zhu *et al.*, 2001), LEA3D (Douguet *et al.*, 2005), Nikitin (Nikitin *et al.*, 2005), FlexNovo (Degen and Rarey, 2006), MED-Hybridize (Moriaud *et al.*, 2009), GeometryFit (Wang *et al.*, 2010), NovoFLAP (Damewood, Lemian and Masek, 2010), Contour (Ishchenko *et al.*, 2012)).

Fragment-based approaches have become the standard in *de novo* design from the late nineties/early two-thousands, although they offer a restricted chemical space search compared to atom-based methods. Fragments can be either small groups or large scaffolds, to account implicitly for drug-likeness and synthetic accessibility while generally reducing the computational cost. Larger building blocks are generally derived from known actives (Fechner and Schneider, 2006) in order to facilitate the design of synthetic routes for the candidates. However, although the strategic selection of fragments can offer a shortcut for a successful molecular synthesis, fragment linking is still performed virtually, thus designed compounds may still be synthetically inaccessible (Hartenfeller and Schneider, 2011).

A number of automated fragment-based techniques have been developed over time: For example, *alignment-based* (e.g. BREED (Pierce, Rao and Bemis, 2004)) and *fragment-shuffling* approaches (e.g. Fragment Shuffling (Nisius and Rester, 2009)) work

through the superimposition of ligands in the binding site, which are then fragmented on strategic bonds to generate their corresponding key fragments. Fragments are then exchanged to generate new candidates with similar pharmacophoric properties using a *linking* strategy. Due to their focus on molecular interactions, these methods work effectively only with active conformations of reference ligands (Schneider and Fechner, 2005). Other techniques rely, for example, on the use of force field-based scoring (e.g. GANDI (Dey and Caflisch, 2008), GeometryFit (Wang *et al.*, 2010), Contour (Ishchenko *et al.*, 2012)), or docking (e.g. FlexNovo (Degen and Rarey, 2006), Hecht and Fogel's approach (Hecht and Fogel, 2009), AutoGrow (Durrant, Amaro and McCammon, 2009) (Durrant, Lindert and McCammon, 2013)) to drive the generation of new ligands.

More advanced fragment-based approaches have also been proposed with the aim of accounting more explicitly for the issue of synthetic accessibility. The main approaches can be categorised as *pseudoretrosynthetic*, *Markov-chain-*, and *reaction-based* methods.

Pseudoretrosynthetic-based algorithms simulate the fragmentation of a reference ligand by means of retrosynthetic rules. These rules normally consist of a set of substructures (e.g. SMARTS) that identify cleavable bonds. Retrosynthetic algorithms use cleavage rules to disassemble compounds to generate hypotheses on their synthetic routes or to compute synthetic feasibility scores (e.g. CAESA algorithm applied in SPROUT (Gillet *et al.*, 1995)); whereas, pseudoretrosynthetic algorithms exploit the same principles for *de novo* design: ligands are first decomposed into key fragments, which are then used as queries for the retrieval of building blocks with similar features. The new building blocks are then recombined to design candidates with properties similar to the reference ligands. In addition, the recombination is usually performed only with building blocks describing attachment points similar to those in the key fragments to enhance the synthetic accessibility of the candidates. RECAP (Retrosynthetic Combinatorial Analysis Procedure) (Lewell *et al.*, 1998) is the most famous set of rules implemented in several fragment recombination programs (e.g. TOPAS (Schneider *et al.*, 2000), Flux (Fechner and Schneider, 2006) (Fechner and Schneider, 2007),

COLIBREE (Hartenfeller *et al.*, 2008)), which describes 11 cleavage bond types. An example of a pseudoretrosynthetic *de novo* design scheme is illustrated in Figure 2.2:

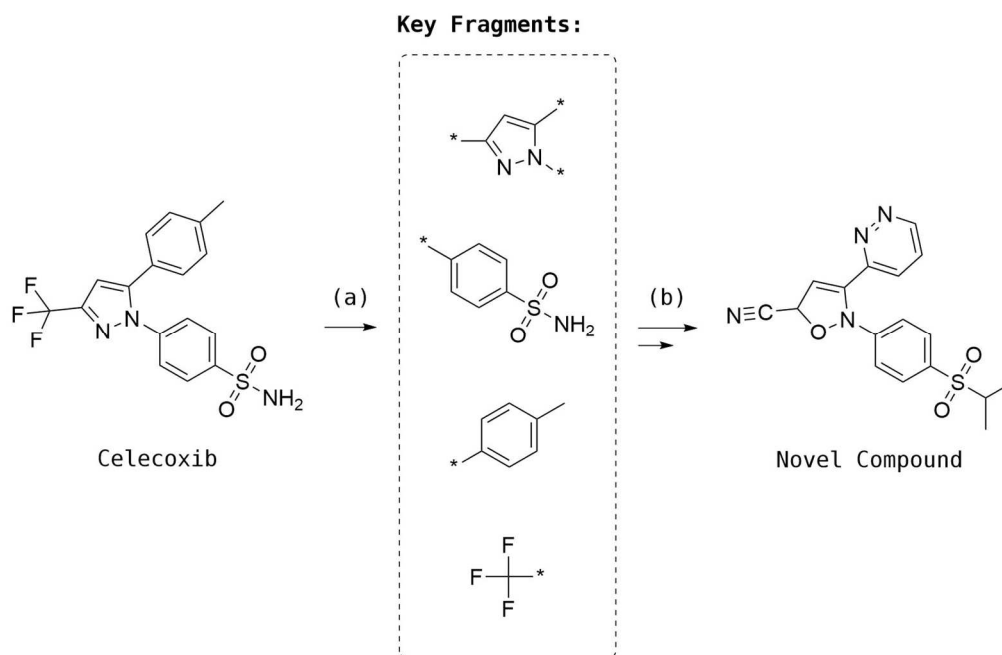


Figure 2.2: Pseudoretrosynthetic design applied to the molecule Celecoxib. First, the query ligand bonds are (a) broken to yield a set of key fragments. Fragments are then used as references to retrieve similar blocks that are (b) recombined to yield novel compounds with properties similar to the original query.

Markov-chain-based algorithms (e.g. FOG (Kutchukian, Lou and Shakhnovich, 2009) bias the sequential growth of compounds by connecting new fragments according to a reference set of connection rates (frequencies) extracted from a collection of pharmaceutically relevant molecules. For example, the FOG (Fragment Optimized Growth) algorithm exploits a first-order Markov chain model (Norris, 1998) to describe a sequence of possible bonds for each state (molecule), where the current state is not influenced by past states. The growing starts with the random selection of a fragment, then the first step of the expansion occurs by evaluating the states connected to that fragment according to the probabilities associated with them. States with higher probabilities will be evaluated first. The algorithm proceeds with exploring new solutions iteratively according to this criterion, until satisfactory solutions are found (Kutchukian *et al.*, 2013). Recent *de novo* design algorithms based on deep learning often exploit the

Markov-chain principles to generate chemically meaningful and accessible candidates (Olivecrona *et al.*, 2017) (Li, Zhang and Liu, 2018).

Reaction-based methods (e.g. SYNOPSIS (Vinkers *et al.*, 2003), the Reaction Vector Structure Generation (RVSG) (Patel *et al.*, 2009) algorithm, DOGS (Hartenfeller *et al.*, 2012), DINGOS (Button *et al.*, 2019)) attempt the construction of novel structures by mimicking existing synthetic routes on a set of building blocks on which transformation rules are applied. For example, SYNOPSIS, DOGS, and DINGOS use pre-defined sets of hard-coded reactions, while the RVSG method can extract new rules automatically from datasets of reference reactions (see Chapter 3). On the one hand, the advantage of these methods is that they also associate the designed structures with the virtual routes used to create them, hence providing suggestions for their synthesis. On the other hand, the number of solutions that can be explored by these algorithms is further restricted by the use of transformation rules (Hartenfeller, Renner and Jacoby, 2013).

2.6. Search Components

As mentioned above, the number of potentially accessible organic compounds has been estimated between 10^{20} and 10^{60} (Bohacek, McMartin and Guida, 1996) (Ramström and Lehn, 2002) (Ertl, 2003) (Polishchuk, Madzhidov and Varnek, 2013). Despite the large discrepancy between these estimations, all scientists agree on the fact that this region of chemical space is too vast to be explored systematically. For this reason, algorithms for compound search and optimisation in *de novo* design have been developed. Search algorithms can be mainly divided into *stochastic* and *deterministic*.

2.6.1. Stochastic Search

Stochastic search algorithms attempt the exploration of chemical space by means of sampling. For example, the neighbourhood of a given molecule would be characterised by randomly picking a few surrounding neighbours, which would be then used to predict the subspace structure in order to move towards the most promising directions. One important caveat is that these algorithms typically rely on heuristics, hence producing solutions that are not globally optimal. Second, since these algorithms incorporate a

component of randomness (e.g. for the sampling), multiple runs are likely to deliver different results; thus, the use of statistical methods is often necessary to obtain consistent results. Third, stochastic search can often lead to local trapping even when exploring subspaces that still adhere to suitable chemical landscape requirements.

Markov chains are an example of stochastic process used in *de novo* design (e.g. the search algorithm implemented in CONCEPTS (Pearlman and Murcko, 1993)), and are often integrated with mathematical conditions to avoid local trapping (e.g. the Metropolis criterion based on Boltzmann's probability (Metropolis *et al.*, 1953) (Altekar *et al.*, 2004)). These algorithms generally accept structural variations that lead to an increase in the objective score (e.g. predicted activity), while those resulting in worse scores may be accepted only if they meet some particular conditions. *Simulated annealing* works in a similar way, and reshapes the rejection criteria dynamically as the optimisation runs. For example, at the beginning of the optimisation, structural variations that produce negative scores are more likely to be accepted, to avoid local trapping and promote exploration, while at the end of the process the conditions that compounds have to fulfil are stricter, so that the algorithm can focus more on the exploitation of the local subspace. The work described by Guo and colleagues (2004) is an example of a simulated annealing-based optimisation algorithm.

Two more classes of optimisation algorithms that have gained significant popularity in drug design are the *genetic algorithms* (GAs) and *particle swarm optimisation* (PSO), which are inspired by natural processes (Hiss, Hartenfeller and Schneider, 2010).

Genetic algorithms refer to Darwin's Theory of Evolution (Darwin, 1859). In a *de novo* design context, a population of starting molecules is exposed to random mutation and crossover by particular operators to produce children which are then included in the population (Yang, 2014). The new population is then scored and a number of best solutions are retained. The process is repeated for a series of iterations moving from global (exploration) to local (exploitation) searches. Mutations can be roughly divided into *atom-based and fragment-based*. The same principles and limitations reported for atom- and fragment-based constructions apply here (e.g. atom-based mutations enable

better exploitation of the chemical space, yet they often generate unstable compounds). Some examples of programs implementing GAs for compound search and optimisation are LEA (Dougnet, Thoreau and Grassy, 2000), LigBuilder (Wang, Gao and Lai, 2000) (Yuan, Pei and Lai, 2011), SYNOPSIS (Vinkers *et al.*, 2003), CoG (Brown *et al.*, 2004), GANDI (Dey and Caflisch, 2008), and AutoGrow (Durrant, Amaro and McCammon, 2009) (Durrant, Lindert and McCammon, 2013)).

Particle swarm optimisation reflects the behaviour of real swarms of animals (e.g. birds searching for food). At the beginning of the optimisation, a number of particles (a swarm) which can communicate with each other, start the exploration of the solution space. Every time a promising solution is found, its score and position are shared with the rest of the swarm, influencing the behaviour of the group and, therefore, enabling the fine-grained exploration of promising areas while ignoring less attractive regions. An example of software implementing a PSO-based optimisation strategy is COLIBREE (Hartenfeller *et al.*, 2008). A similar strategy is implemented in the program MAntA (Molecular Ant Algorithm) (Reutlinger *et al.*, 2014), which mimics the behaviour of ant colonies when searching unexplored areas. In nature, when ants manage to find an interesting path, for example, leading to a food source, they mark the route with a substance called pheromone to attract more ants in that area. MAntA searches for optimal combinations of fragments according to their pseudo-probabilities (pheromones) which can be computed, for example, using QSAR. As the simulation runs, high-scoring combinations generate more intense pheromone trails, driving the search towards those regions of chemical space in order to find local optima.

2.6.2. Deterministic Search

Deterministic search algorithms are characterised by no random components, so that, for a given input, the search always produces the same output. These algorithms are less common in compound optimisation since certain parameters (e.g. directions of the chemical space to explore next) are hard to define *a priori*. Nevertheless, several deterministic strategies in *de novo* design have also been proposed. For example, a deterministic grow strategy is implemented in FlexNovo (Degen and Rarey, 2006) for

the prioritisation of fragment connections. First, each fragment is docked in the binding site to produce a single score. Second, the molecular growing is controlled by only evaluating fragments that can contribute positively to the global score of the extended molecule. A similar approach consists of ranking fragments individually by means of scoring functions, then summing up the best contributions according to the principle of fragment additivity to avoid the scoring of multiple combinations of fragments. Two examples of *de novo* design software using fragment additivity approaches are Nikitin (Nikitin *et al.*, 2005) and BI CLAIM (Lessel *et al.*, 2009).

2.7. Artificial Intelligence in de novo Design

The evolution of *de novo* design algorithms over time has involved the introduction of an increasing number of constraints and rules, for example, the use of fragment-based methods to account for drug-likeness and synthetic accessibility. However, the introduction of many constraints can bias the search space, leading to the exploration of subspaces already assessed or ignoring of promising regions (Gómez-Bombarelli *et al.*, 2018). Deep learning (DL) and artificial intelligence (AI) techniques attempt to provide a solution to these issues using big data collections to model implicitly the rules necessary for successful *de novo* design while maximising the accessibility to the full search space (Chen *et al.*, 2018). Although AI technologies have been applied to augment most components of *de novo* design algorithms, such as scoring functions and QSAR models, this subsection focuses only on those for molecule generation.

In AI approaches to *de novo* design, molecular construction often relies on the use of *generative models*, which are statistical models based on deep neural networks (DNNs) that aim to capture correlations and patterns in the training data and generate new data instances with some variations (Bishop, 2006). For example, generative models are trained with drug-like compound datasets to generate new structures that are also drug-like. SMILES strings are frequently used as input since many of the methods that are applied in *de novo* design have been demonstrated as being effective with sequential data with long-range dependencies (Segler *et al.*, 2017), and because SMILES strings can be

readily converted back into molecules (Gómez-Bombarelli *et al.*, 2018). Components of generative models include autoencoders (AEs) (e.g. adversarial (AAEs), variational (VAEs)), generative adversarial networks (GANs), and recurrent neural networks (RNNs), which are often combined with predictive models (e.g. QSAR/QSPR) that are capable of driving the generative component towards the design of structures with desired properties (Xue *et al.*, 2019). Generative models (*agents*) and predictive models (*interpreters*) are combined to form automated decision-making frameworks where the agent learns which actions are best according to the reward generated by the interpreter. These systems belong to the category of *reinforcement learning* (RL).

Autoencoders (AEs) are neural network-based (NN-based) architectures for unsupervised feature representation learning, which consists of three components: an encoder, a decoder, and a distance function. The encoder converts the input data (e.g. a SMILES string) into a representation with lower dimensionality (a continuous vector), then the decoder attempts the reconstruction of the original input from the low dimensional representation. The lower-dimensional space generated during the encoding/decoding process is referred to as *latent space* and the distance function is concerned with measuring the loss between the original (input) and the reconstructed (output) representations. Autoencoders are generally applied to *de novo* design by following a two-step process: first, the architecture is trained using a set of molecules with particular properties (e.g. drug-likeness) to minimise the loss between encoding and decoding; second, once the training is completed, the encoder is extracted and used for the generation of new representations by sampling the areas surrounding the chemical space covered by the training set; finally, these vectors are reconverted into SMILES strings by the decoder.

VAEs and AAEs are modifications of basic autoencoders that have been demonstrated to work effectively with SMILES strings (e.g. VAE (Gómez-Bombarelli *et al.*, 2018), Semi-supervised VAE (Kang and Cho, 2019), AAE (Blaschke *et al.*, 2018)), molecular fingerprints (e.g. AAE (Kadurin, Aliper, *et al.*, 2017)), and molecular graphs (e.g. GraphVAE (Simonovsky and Komodakis, 2018), the method by Li *et al.* (2018),

JT-VAE (Maziarka *et al.*, 2019)). Both VAEs and AAEs implement a constraint so that models can learn a generalised sampling function of the input data (molecules) which maps to a continuous representation of the latent space (*latent variable*). VAEs and AAEs treat the latent variable in a probabilistic manner to decompose and reconstruct high-dimensional representations, with the exception that AAEs also integrate an NN-based discriminator that is aimed at biasing the decoder to generate output data that follows a specific target distribution.

Gomez-Bombarelli and colleagues (2018) pioneered the implementation of AE-based generative models using SMILES strings, driven by QSPR models with the aim of optimising particular properties, in their specific case, logP, Quantitative Estimation of Drug-likeness (QED) (Bickerton *et al.*, 2012), and Synthetic Accessibility score (SAS) (Ertl and Schuffenhauer, 2009). Later, Blaschke and colleagues (2018) investigated the difference in performance between autoencoder architectures, where they demonstrated that AAEs are more effective and efficient than VAEs for generative tasks, at the cost of some loss in terms of chemical space coverage. Kadurin and colleagues (2017a) (2017b) introduced an AAE architecture for molecular fingerprint generation, which was applied to the design of anti-cancer drugs. Their model generated a series of fingerprints that were used as queries for the similarity screening of a molecule library from PubChem to identify potential candidates, yielding a final selection of 69 novel compounds of which, some already contained annotations as anti-cancer compounds against other targets. Other works implementing autoencoders have also been reported by Dai and colleagues (2018), Lim and colleagues (2018), and Polykovskiy and colleagues (2018).

GANs reflect some implementation aspects of autoencoders. These architectures have two NN-based components which are responsible for molecule generation and discrimination, respectively. The generative network learns the relationship between latent space and the data distribution so that it can generate new instances, while the discriminative network differentiates the instances fabricated by the generator from the original data distribution. GANs are applied to *de novo* design following a procedure similar to that used on autoencoders: first, the architecture is trained until the

discriminator cannot distinguish anymore real from fabricated data; second, the generator is used to output new instances, which are expected to have properties similar to those in the training data.

An example of more complex architecture has been proposed by Méndez-Lucio (2018) by stacking two GANs (i.e., two-step generation) with the aim of producing more refined data instances. In addition, ORGANs (objective-reinforced generative adversarial networks), which reflects the RL implementation of GANs, have been proposed for AI-augmented *de novo* design using either SMILES (e.g. ORGAN (Guimaraes *et al.*, 2017), ORGANIC (Sanchez-Lengeling *et al.*, 2017)) or graph representations (e.g. MolGAN (De Cao and Kipf, 2018)). However, although GANs have demonstrated better suitability for molecule generation than AEs, they can incur the risk of exploring less diverse chemical space due to the constraints generated between the adversarial processes of generator and discriminator (Xue *et al.*, 2019). More complex GAN architectures with improved convergence properties have also been proposed by Putin and colleagues (2018a) (2018b).

RNNs have been used extensively in natural language processing in the last decades (Irsoy and Cardie, 2014), and have also recently become the standard for molecule generation. These frameworks mainly consist of an internal state (*memory*) that is capable of processing and tracking sequences of inputs, for example, SMILES strings, in particular when augmented with micro-architectures such as long short-term memory (LSTM) cells (Goodfellow, Bengio and Courville, 2016). RNNs are applied to *de novo* design by first training them to predict sequences of SMILES characters (tokens) in a given set of molecules; then, once probability distributions are learnt, the trained networks can be used to sample new SMILES strings. Although RNNs require large amounts of training data before being able of outputting valid SMILES strings, they are easier to train compared to AEs and GANs, and they can deal with representations of variable length, whereas autoencoders and adversarial networks require fixed-length vectors.

Yuan and colleagues (2017) and Segler (2017) reported works using RNNs for the generation of molecule libraries with properties similar to those of the compounds used to train the networks. Arús-Pous and colleagues (2019) extended that task to regenerating a much bigger library composed of almost one billion compounds, GDB-13 (Blum and Raymond, 2009), by training an RNN using less than one per cent of the original input space. In addition, as for the other generative approaches, RNNs have also been coupled with predictive models, for example, Jaques and colleagues (2017) and Olivecrona and colleagues (2017) developed two *de novo* design RNN-based RL frameworks for the generation of structures with desirable properties. The work by Olivecrona and colleagues also describe the implementation of a policy-based constraint to penalise certain types of undesirable structures that are not interesting in reality. Popova and colleagues (2018) proposed a more complex RNN-based architecture called ReLeaSE (Reinforcement Learning for Structural Evolution) which was validated on the optimisation of melting point, hydrophobicity, and activity towards Janus kinase 2.

Generative models have also been deployed for *transfer learning* tasks. These approaches are aimed at transferring some knowledge acquired previously (e.g. implicit rules for drug-likeness learnt from a large dataset) to a new task (e.g. generation of relevant analogues of a known ligand). Gupta and colleagues (2018) and Awale and colleagues (2019) reported examples of RNNs applied to transfer learning. They both first trained an RNN model using large compound datasets, then they applied their models to the generation of focused libraries by feeding the models with small subsets of compounds with particular properties. Another interesting application is the one reported by Sattarov and colleagues (2019) where autoencoders were combined with a GTM (*generative topographic mapping*) module to generate focused molecular libraries of interest. GTM is a machine learning method for dimensionality reduction and data visualisation that has often been applied successfully to map the chemical space (Owen *et al.*, 2011) (Kireeva *et al.*, 2012) (Gaspar *et al.*, 2015). Ståhl *et al.* (2019) proposed one of the latest approaches using RNNs, which consists of a fragment-based RL framework

where structures are generated from an initial set of compounds with optimal properties, which are then tuned for a given problem by replacing the fragments on the compounds.

2.8. Conclusions

In this chapter, the principles and limitations of *de novo* design have first been introduced by discussing the concept of chemical space and its characteristics. According to these constraints, the main components of *de novo* design programs have been presented. Each component has been described according to its role and most relevant implementations, with the aim of providing an effective comparison between early and modern algorithms. The final section provided an introduction to recent applications of artificial intelligence to *de novo* design. The next chapter focuses on the concept of reaction vector and its implementation in algorithms for molecule generation.

Chapter 3: Reaction Vectors

3.1. Introduction

Reaction vectors (RVs) were originally conceived for reaction classification purposes, yet they have also found application in synthetically accessible *de novo* design. This is due to their ability to incorporate information on reaction centres, which can then be used to generate new synthetic paths virtually. The implementation and evolution of reaction vectors have been described in the works by Patel and colleagues (2009a) (2009b), Hristozov and colleagues (2011), Gillet and colleagues (2013), and Wallace (2016). The content of these works is reviewed to provide an understanding of how reaction vectors can be used to generate molecular structures.

3.2. The Concept of Reaction Vector

Reaction vectors (or *difference vectors*) encode the changes occurring between final and initial states of chemical transformations (Broughton, Hunt and MacKey, 2003) according to the general form described in Equation 3.1:

$$\text{Reaction Vector} = [\sum \text{Product Vectors}] - [\sum \text{Reactant Vectors}]$$

Equation 3.1: Generic definition of reaction vector.

The procedure consists of subtracting the reactants descriptions from the product descriptions to obtain a set of lost and gained features that are identified by negative and positive values, respectively. Reaction components are normally described by topological notations such as atom-pair descriptors. An atom-pair describes a pair of atoms and their properties, divided by a separator that indicates the length of the atom path between the two atoms (see Section 1.2.2.3). For example, an AP2 describes two atoms directly bonded (i.e., the value 2 indicates that only two atoms are contained in the atom-pair), while AP3 and AP4 describe atoms separated by one and two atoms, respectively. When atoms are directly bonded (i.e., AP2), the separator can also contain

information on the bond order. The current implementation of the reaction vector is based on a modified version of the atom-pair descriptor according to Equation 3.2:

$$X_1(h_1, p_1, r_1) - S(BO) - X_2(h_2, p_2, r_2)$$

Equation 3.2: Modified atom-pair notation adopted in the current implementation of reaction vectors.

X_1 and X_2 are the atom types; h_1 and h_2 represent the number of non-hydrogen bonds formed by the atom; p_1 and p_2 represent the number of π electrons shared by the atom. The ‘p’ property is calculated by evaluating all the bonds formed by the atom. For each double bond or aromatic bond, p is incremented by 1, while for each triple bond, p is incremented by 2; r_1 and r_2 represent the number of rings the atom is part of (Downs *et al.*, 1989); ‘S’ is the separator; ‘BO’ is the connection bond order. Single, double, triple, and aromatic bond orders correspond to 1, 2, 3, and 4, respectively. The current implementation of reaction vectors does not describe stereochemistry, chirality, and explicit hydrogens. An example of reaction vector generation is reported in Figure 3.1, where reaction components are first described using only AP2 descriptors, then descriptions are subtracted as indicated in Equation 3.1. Figure 3.1 shows that only the atom pairs that have changed during the reaction are identified, while the unchanged descriptions are not considered. The result of this procedure can be also represented as a list of lost and gained atom pairs associated with negative and positive integers, respectively, as shown in Figure 3.2.

Note that reaction vectors do not implement any sort of knowledge to recognise whether reactions describe valid chemistry, rather, they treat reactants and products as separate species that are then simply subtracted from each other. Hence, reaction vectors can also contain chemically incorrect or meaningless information, especially when encoded from reactions that are not curated. For this reason, before generating them, a reaction cleaning procedure is usually recommended.

Reaction vectors can be used to automatically extract reaction centres from datasets of known transformations to generate rules, which can subsequently be exploited in a

number of cheminformatics applications, including *de novo* design and reaction classification. These methods have also been referred to as *knowledge-based* or *data-driven* since they do not rely on pre-defined rules, yet they operate differently according to the data used with them.

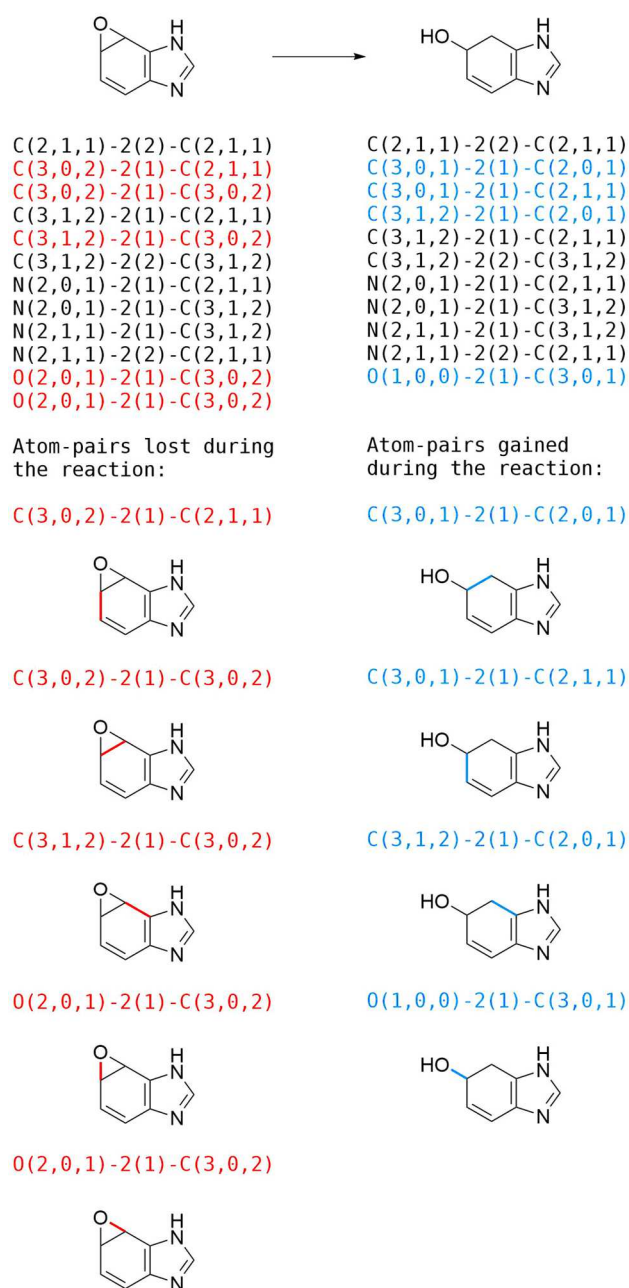


Figure 3.1: Example of reaction vector generation using AP2 descriptors. Lost and gained atom pairs are highlighted in red and blue, respectively.

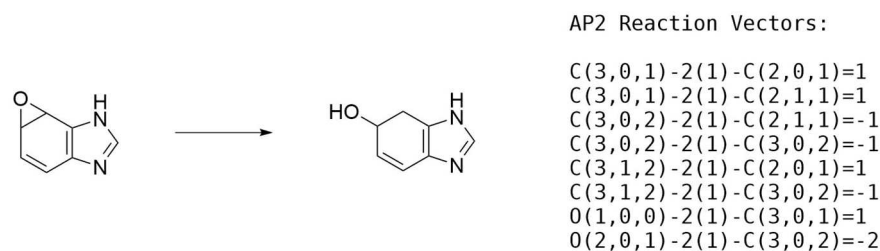


Figure 3.2: The AP2 reaction vector.

3.3. Structure Generation Using Reaction Vectors

Although the generation of reaction vectors can be seen as a relatively simple process, their implementation for molecule generation is considerably more sophisticated. Patel and colleagues (2009a) (2009b) developed the first molecular design framework based on reaction vectors. Their approach was first validated by applying the algorithm to the reproduction of known products using the corresponding reactant and reaction vector. The method was then validated by generating novel structures, which were assessed on their diversity against the products described in the reference reactions and on their relevancy in lead optimisation and library enumeration. Hristozov and colleagues (2011) followed Patel and colleagues' work by introducing a more powerful algorithm which was assessed on its applicability on a larger number of reaction classes. Both studies suggested that the combination of AP2 and AP3 vectors is the most effective for *de novo* design purposes since it provides a balance between generality and specificity, which are both required in order to generate novel and synthetically accessible structures, respectively. Finally, Gillet and colleagues (2013) described a more complex integration of reaction vectors for multi-objective optimisation.

The reaction vector-based design framework consists of three components: *starting materials*, which are molecules of interest, for example, obtained from screening or crystallisation data; *reagents* (optional), which usually correspond to sets of building blocks that are aimed at being combined with the starting materials; a *reaction vector database*, which is the source of reaction vectors. Database entries can also contain additional information such as synthetic methodologies (e.g. links to the original reaction references), reagents from reference reactions, reaction classes, time data, etc. This

design framework can be seen as an independent construction module which, given the necessary inputs, enumerates all the products that can be obtained according to the rules stored in the database. Due to this characteristic, the algorithm can be easily integrated in more sophisticated architectures and combined with other modules such as scoring functions or filters.

3.3.1. Original Algorithm

The original molecule generation algorithm implements a breadth-first search scheme which explores systematically all the atoms and bonds that can be processed on a starting material according to a given reaction vector, until all possibilities have been pursued (Patel *et al.*, 2009).

The algorithm's steps can be summarised as follows. First, the starting material is decomposed by removing its atoms and bonds according to the negative atom pairs (i.e., features that are lost during the reaction) described in the reaction vector. This operation results in the generation of one or more abstract fragments containing atoms with unsatisfied valences. Note that only one of these fragments reflects the correct decomposition described in the vector. Second, the fragments are extended by adding new atoms and bonds according to the positive atom pairs (i.e., features that are gained during the reaction) described in the reaction vector. At each step of the reconstruction, the atom with the highest unsatisfied valence is selected as a *seed atom* for the growing process, then the positive AP2s in the vector matching with the seed atom and its neighbourhood are used to extend the fragment. As the operation proceeds, the AP3s in the extended fragments are checked against the positive AP3s in the reaction vector to determine which solutions should be discarded and which should be pursued; hence, paths are validated through the analysis of AP3 vectors. The operation continues until all the positive APs in the vector are applied and all atom valences are satisfied. The application of the original algorithm is illustrated in Figure 3.3.

Patel and colleagues (2009) tested the algorithm on a variety of organic reaction types, including relatively complex oxidations, reductions, and rearrangements,

reporting 85% of the assessed reaction vectors (5,695 reaction vectors) being able to reproduce their corresponding reference reactions. Nevertheless, although this method was demonstrated to be generally effective, the use of reaction vectors from bigger and more complex reaction centres resulted in long generation times or failures due to timeout. As a consequence, a new version of the algorithm was introduced to permit more efficient management of computational resources.

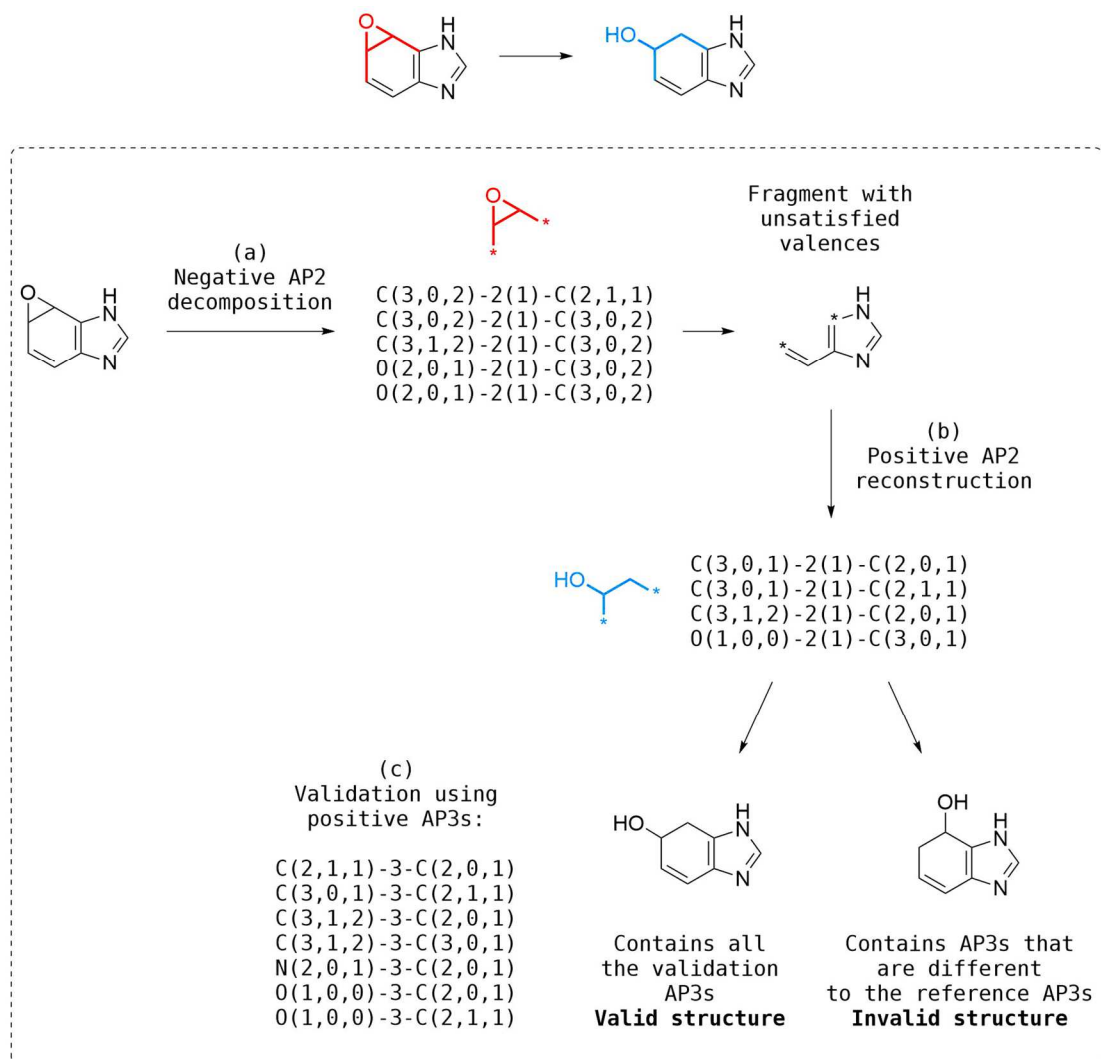


Figure 3.3: Example of structure generation using the algorithm introduced by Patel *et al.* (2009): (a) the starting material is processed by removing the negative AP2s in the vector. Multiple fragments with unsatisfied valence can be obtained after this procedure; (b) an abstract fragment is processed by adding the positive AP2s in the reaction vector; (c) as the growing continues, the new structures are checked on the presence of the positive AP3s in the reaction vector. Structures that generate AP3s that are not described in the reaction vector are considered as invalid.

3.3.2. Revised Algorithm

The current implementation of the reaction vector-based molecule generation algorithm is referred to in this work as the *reaction vector structure generator* (RVSG). The new algorithm was developed and tested by Hristozov and colleagues (2011) who replaced the breath-first search with a more efficient fragment-based approach. In order to achieve higher efficiency, the reaction vector encoding algorithm was modified to store additional information on fragments generated during the reactions. For a given reaction vector, the additional data describes an ordered list of fragments, also indicated as a *recombination path*, that is used to drive the algorithm when the reaction vector is applied to different starting materials.

The generation of recombination path data is summarised as follows. First, given a reaction, its corresponding reaction vector is computed, which in turn is applied to produce a structural decomposition on both starting material and product. This procedure is also indicated as *reverse fragmentation* since it occurs in both directions. More specifically, atoms and bonds are removed from the starting material and product according to the negative and positive APs in the vector, respectively, each one yielding a set of abstract fragments. These two fragment sets are compared to one another to identify shared fragments, which are then assumed to be the connection between starting material's and product's fragmentation paths. Once a connection is found, the remaining fragments necessary for the structure generation process are identified indirectly by excluding unchanging substructures between starting material and product. This operation is performed by an MCS algorithm (see Section 1.3.1).

Note that the more complex a reaction is (e.g. larger reaction centre, multiple components, etc.), the more difficult it is to find its recombination path. The recombination path is stored with the reaction vector in an indexed database for rapid information retrieval to further increase the speed of the process. An example of recombination path generation is illustrated in Figure 3.4.

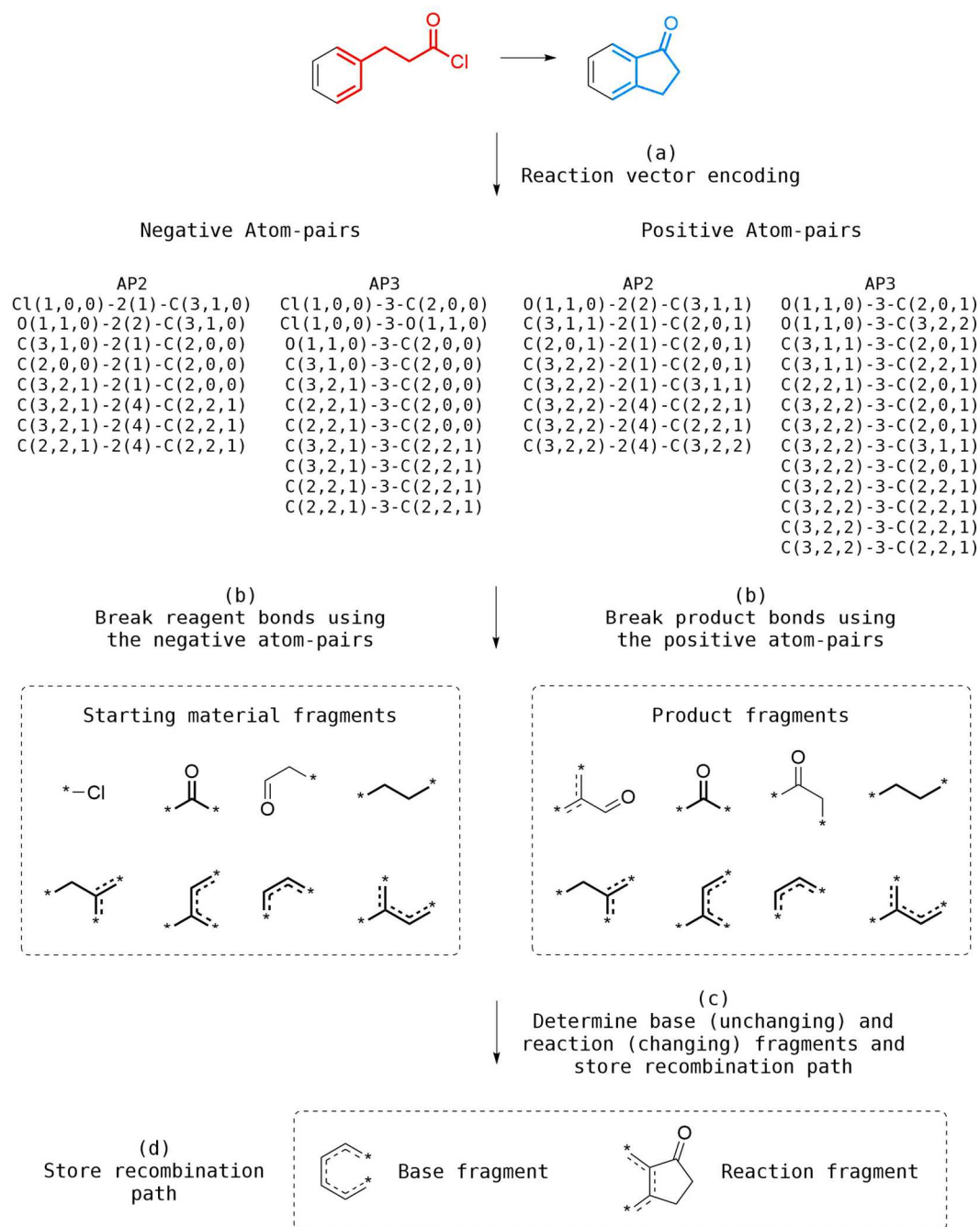


Figure 3.4: Example of recombination path generation using the algorithm developed by Hristozov *et al.* (2011): (a) the reaction is used to produce its AP2+AP3 reaction vector; (b) the starting material and product are decomposed using the negative and positive atom pairs in the reaction vector, respectively. A series of fragments from both starting material and product are generated. The shared fragments are described in bold; (c) if any shared fragments can be found, the algorithm stores the recombination path which includes *base fragment* (unchanging substructure determined using MCS) and *reaction fragment* (changing substructure).

The revised algorithm also implements an alternative approach to cope with cases where the reverse fragmentation strategy fails. In these cases, the breadth-first approach described in the original algorithm is applied to produce an ordered list of atom pairs that are stored in place of the recombination path. If the reaction cannot be processed using either of the two approaches, then its reaction vector is not stored in the database.

Hristozov and colleagues (2011) validated the revised algorithm on the same set of reactions used by Patel and colleagues (2009), reporting almost 90% of reactions successfully reproduced, although not all reaction types were validated with the same percentage of success. Results by Hristozov and colleagues (2011) also describe a remarkable increase in speed compared to the original algorithm reported in Patel's PhD thesis (Patel, 2009): An average run using the new algorithm took 0.015 seconds per reaction using a database of 5,695 vectors, versus 3.1 seconds for the old algorithm, necessary to only identify all applicable reaction vectors and reagents using a database of 6,016 vectors.

Once a database of reaction vectors has been created, it can be applied to new starting materials to generate novel structures. First, the atom pairs of a given starting material (and an optional set of reagents) are compared to the negative atom pairs for each reaction vector in the database to identify possible matching reaction vectors. Once an applicable vector is identified, the starting material's atom pairs are removed according to the negative APs in the database entry, and the fragments described in the recombination path are added, according to the order described in the database.

The application of the revised algorithm is illustrated in Figure 3.5. Following this operation, a new structure is generated and it is sanitised using the RDKit library (Landrum, 2016) to ensure its validity. The sanitisation process consists of cleaning valences, aromaticity, and hybridisation states. If the structure fails the sanitisation, the product is rejected. A given reaction vector can be applied to the same starting material at multiple reaction centres that match the atom pairs described in the database.

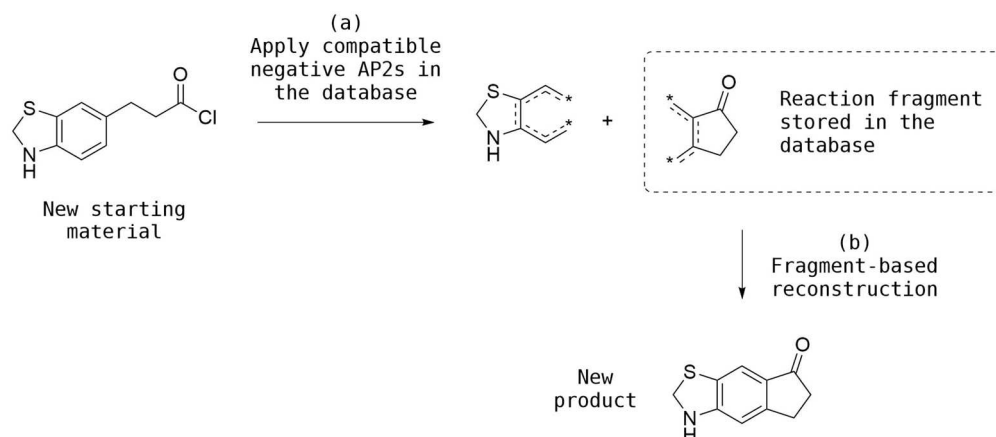


Figure 3.5: Example of structure generation using the algorithm developed by Hristozov *et al.* (2011): (a) a new starting material is checked against the database entries to find reaction vectors with matching negative AP2s. If a matching entry is found, the starting material is processed by removing its atoms and bonds according to the APs described in the entry, to yield an abstract fragment; (b) the abstract fragment is combined with the reaction fragment stored in the matching database entry to yield a new product.

3.3.3. Handling Multiple Reactants

The original and revised algorithms have been described using one-component reaction examples, which are more suitable for illustration purposes. Nevertheless, organic reactions typically contain more than one reactant and, as introduced previously, the reaction vector-based design framework also consists of a reagent component, which enables the use of external sources of reagents. An example of a multiple-reactant transformation is reported in Figure 3.6, which describes the formation of a carbon-carbon bond between two compounds.

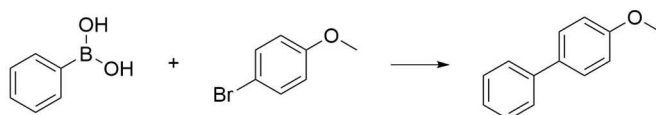


Figure 3.6: Example of C-C bond formation using two reactants.

The reaction vector encoding algorithm processes these cases by storing the negative APs of all reactants in the database. Subsequently, when a new starting material partially matches a reaction vector, the structure generation algorithm also performs a

search in a reagent pool to identify molecules that contain the remaining APs necessary to apply the reaction vector. Reagents can be found either in the reaction vector database, where they are stored as an additional field during the reaction vector encoding, or they can be fed to algorithm externally. In the first case, the reagent pool will contain all the reactants described in the reference reactions used to generate the vectors, while in the second case, it will contain compounds from the external input.

3.4. Conclusions

In this chapter, the general concept of the reaction vector and its computation using topological descriptors has been described. Subsequently, the application of reaction vectors for *de novo* design has been illustrated by focusing on the principles and limitations of the breadth-first approach introduced by Patel and colleagues (2009), which later motivated the development of a more efficient method by Hristozov and colleagues (2011). The chapter also discussed the use of multiple and external reagents, aimed at increasing the versatility of the approach for design purposes, also pointing out that reaction vectors can result to be corrupted when encoded from reaction examples that are not curated. The next chapter describes the main concepts and techniques used in supervised machine learning classification.

Chapter 4: Machine Learning

4.1. Introduction

Learning can be defined as the process that involves the acquisition of new, or the modification of existing, knowledge (Holt *et al.*, 2012). The problem of learning has been investigated in many fields over time, including psychology, statistics, mathematics, and computer science. Machine learning (ML) is an area of computer science that is concerned with the development of algorithms that are aimed at generating useful models using data instead of explicit programming (Alpaydin, 2010). Machine learning methods have been applied extensively in chemoinformatics and related fields, especially in the last decade with the explosive growth in the amount of accessible chemical data. This chapter introduces the main concepts of supervised learning for classification, with a particular focus on multi-class and multi-label problems. The chapter describes the algorithms and approaches that are used to address these tasks computationally, and reports some examples of multi-label classification in chemoinformatics and drug discovery. The chapter also introduces the principles of an approach for confidence estimation in machine learning referred to as Conformal Prediction.

4.2. Supervised Classification

Machine learning techniques can be mainly divided into *unsupervised* and *supervised*. *Unsupervised learning* attempts to model the underlying distribution in the input data to identify useful patterns and relationships that can be used to get a better understanding of the composition of the data. *Supervised learning*, on the other hand, tries to map the input data to an output variable to learn a function that can be applied to new data (Bishop, 2006). This task involves a procedure called *model training* which consists of fitting a function to a set of examples to yield a model that is capable of generalising to similar data. The generalisation towards unseen examples is achieved by means of a series of algorithmic assumptions that are referred to as *inductive bias* (or *learning bias*) (Mitchell, 1980). Supervised problems can be further divided into two

main categories, specifically *regression* and *classification*, which are concerned with predicting continuous (e.g. pIC₅₀ value, melting point, etc.) and discrete (e.g. reaction class) output variables, respectively.

In both regression and classification, algorithms require the data to be described using a particular input representation, which typically consists of a set of features that are representative of the problem in question. For example, in the prediction of melting point, molecular attributes (input data) such as size, symmetry indexes, energies of attraction (e.g. electrostatic, van der Waals, hydrogen bond, etc.), and percentage of impurities, can be considered as representative for the construction of a useful approximation (model) of the temperatures at which molecules melt (output variable). The use of features that are not relevant to the problem or the omission of important features might lead to *under* or *over* fitting of the data: *Under-fitting* occurs when models cannot produce valid predictions on both training and unseen examples, while *over-fitting* occurs when models can produce accurate predictions only on their training examples. These unfavourable conditions can be generally improved or avoided by selecting appropriate features, data composition, number of examples, algorithms and parameters.

Machine learning algorithms that deal with classification problems are indicated as *classifiers*. Classifiers can be tuned on their parameters to achieve a reasonable trade-off between model variance and bias. Variance represents the ability of the model to fit the training data, while bias relates to the generalisations made by the model in order to predict new instances. Models with high bias are prone to under-fitting, while those with high variance are susceptible to over-fitting.

The modulation of the classifier parameters is referred to as *hyper-parameter tuning* or *model optimisation* and it often involves the use of search algorithms, similar to those described in Section 2.6, which are aimed at determining configurations that result in accurate predictions on unseen data instances.

4.2.1. Classification Problems

In machine learning, classification is the problem of labelling unseen instances according to a set of labelled training examples. This can be performed in a number of different ways according to the selected classifier. Classification problems, as illustrated in Figure 4.1, can be divided into *binary*, *multi-class*, and *multi-label*. In binary and multi-class problems, entries are associated with only one label, while in multi-label problems they can be associated with multiple labels.

Binary classification consists of identifying whether new instances belong to a given class (e.g. positive) or not (e.g. negative). An example of a binary problem in chemoinformatics is determining whether or not a compound is an active inhibitor against a particular target.

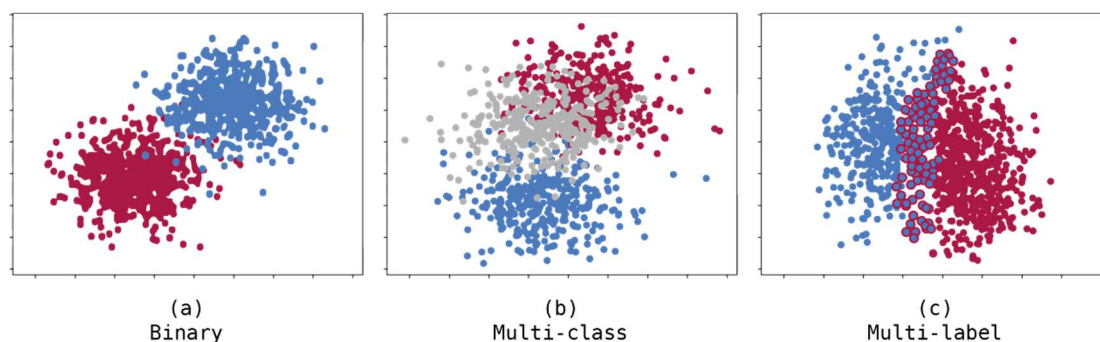


Figure 4.1: Classification problem types described on a two-dimensional space: (a) *binary*: entries can only belong to either class A (red) or B (blue); (b) *multi-class*: entries can only belong to class A (red), B (blue), or C (grey); *multi-label*: entries can belong to class A (red), B (blue), or both classes A and B (blue with red edges).

Multi-class classification consists of determining to which class a given instance belongs, among a set of labels described in the training data. An example of a multi-class problem in chemoinformatics is predicting whether a compound has low, medium, or high toxicity. Multi-label classification is similar to multi-class, yet it accepts more than one label as a prediction for a given instance. For example, the determination of the selectivity profile of a compound would be likely to generate an output describing multiple labels that describe the targets on which that compound is active.

4.3. Classification Algorithms

A number of classification techniques have been developed, ranging from simple to very complex classifiers. Each algorithm has a different inductive bias and constraints, and thus generates a different mapping function. Some algorithms are more prone to over-fitting or other undesirable conditions, whereas some are more robust and able to minimize these effects. Some illustrations of how different classifiers deal with the same binary input are described in Figure 4.2:

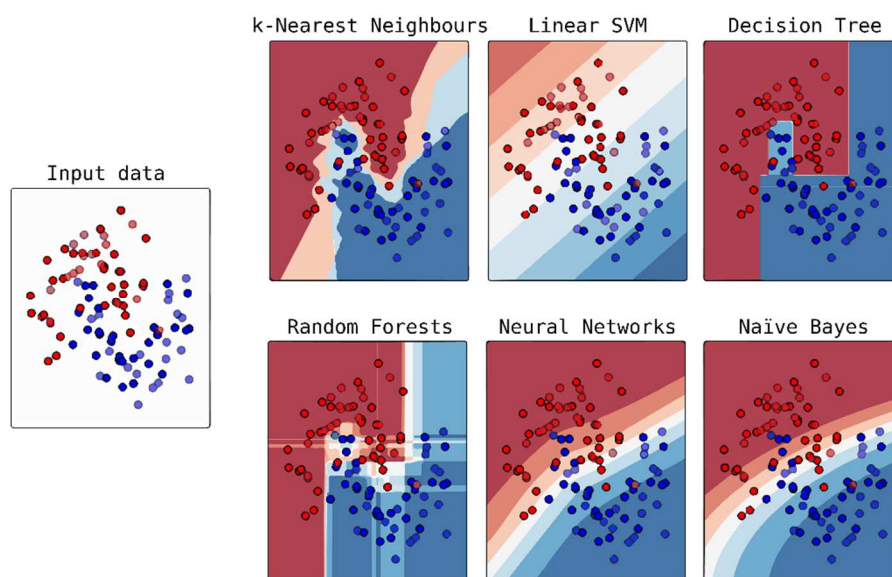


Figure 4.2: Examples of different classifier outputs on the same binary data, where the different coloured areas indicate how algorithms define the domain subspaces, which in turn determine how entries are classified. Entries are represented by coloured dots according to the class they belong to. Image adapted for the purpose of this work (Scikit-learn, 2007).

Classifiers can be divided into *parametric* and *non-parametric* classifiers. Parametric algorithms (e.g. Naïve Bayes) produce strong assumptions on the data in order to simplify the model function to a known form. For this reason, they are easier to understand, quicker to construct, and less demanding in terms of training data requirements; however, they often cannot deal effectively with problems of higher complexity due to the constraints they apply to match the underlying data distribution. Non-parametric algorithms (e.g. k-Nearest Neighbours, Decision Trees, Support Vector

Machine) attempt to construct the mapping function by dividing the input space into local regions on which no assumptions are made. Due to their nature, they are more powerful and flexible, yet they are slower, prone to over-fitting, and require more training examples. Non-parametric algorithms are generally more suitable when a large number of training examples is available and there is no prior knowledge on the data composition (Russell and Norvig, 2016). A number of algorithms of interest in this study are presented below, and are described on their principles applied for binary classification, yet their use can also be extended to multi-class problems.

4.3.1. k-Nearest Neighbours

k-Nearest Neighbours (kNN) is a non-parametric lazy learning technique (Altman, 1992). The term *lazy* indicates that the algorithm does not fit an actual function during the training, rather it postpones the computation until the classification phase, which is then performed based upon the similarity between features of training and test entries, as illustrated in Figure 4.3. The parameter k specifies the number of training examples that inform the classification of a new instance.

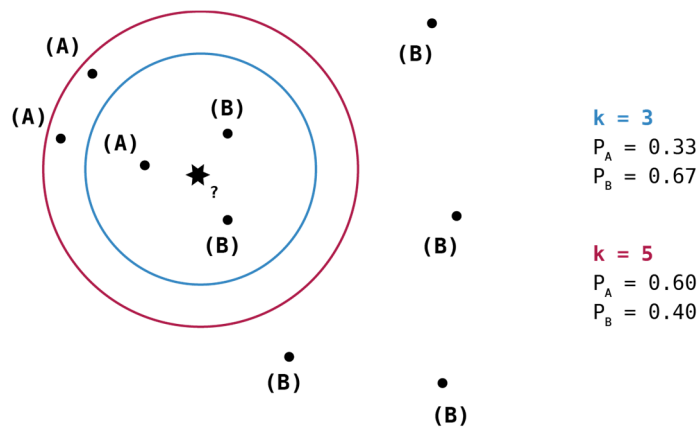


Figure 4.3: kNN classifications ($k = 3, 5$) on a binary dataset (A, B). The test entry is described as a star while training instances are points with their corresponding classes reported in brackets. Frequencies (P_A , P_B) are calculated as ratios between particular instances (e.g. A) on the total number of instances determined by k .

Figure 4.3 shows that for a given test instance, the algorithm searches for its k nearest training examples using a specified metric (e.g. Euclidean distance), then

computes the frequency of each class to finally assign the entry to the class with the highest frequency (i.e., method of majority vote). A higher k value generally results in a greater radius of inclusion of training instances; hence, this parameter needs to be tuned to determine the most effective configuration for a given problem. kNN is a simple, effective, and interpretable algorithm, yet it does not account for feature importance and performs poorly with high-dimensional data due to its lazy nature (Alpaydin, 2010). For this reason, it also requires all the training data to be stored in order to use it for the classification of new entries.

4.3.2. Decision Trees

A Decision Tree (DT) is a non-parametric hierarchical algorithm that uses training examples to derive a set of conditional rules to split recursively the data and define local regions (Quinlan, 1983). A tree is represented as a flowchart that begins at the root node and branches out to multiple decision nodes and terminal leaves, as described in Figure 4.4. Decision nodes have branches, while terminal leaves represent outputs (i.e., classes). Each node implements a local function.

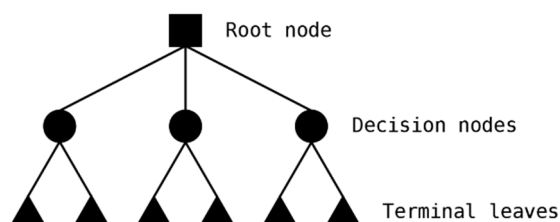


Figure 4.4: Example of decision tree. The square and circles represent root and decision nodes, respectively, whilst the triangles represent terminal leaves.

Decision trees are constructed using a top-down greedy search which explores multiple branches sequentially by scoring them according to a selected metric (e.g. Gini impurity (Buntine and Niblett, 1992), Shannon entropy (Shannon, 1948)). In this context, metrics are generally aimed at maximising the quality of the split, for example, by evaluating the homogeneity of examples according to their class composition; hence, metrics are used to determine the most effective rules to split the training data by class. The deeper the tree, the more complex the conditions will be. DTs are computationally

cheap, easy to understand, and they can handle both categorical and numerical data; however, they are prone to over-fitting (Alpaydin, 2010). For this reason, more robust approaches based on ensembles of trees, such as Random Forests and Gradient Boosted Trees, have been proposed.

4.3.2.1. Random Forests

Random Forest (RF) operates by constructing an ensemble of decision trees (*forest*) (e.g. 100) via *bootstrap aggregation* (bagging) (Ho, 1995) (Ho, 1998) (Breiman, 2001). First, decision trees are trained on random samples of training data (*instance bagging*) where instances are drawn with replacement (i.e., the same examples can be used more than once). Second, when trees are deployed to classify a new instance, their predictions are aggregated to determine the final class according to a particular method of voting. For example, the RF classifier implemented in scikit-learn (<https://scikit-learn.org/>) (Pedregosa *et al.*, 2011) infers class probabilities using the method of *soft-voting*, which consists of averaging the probabilities associated with each class in the trees of a forest, then selecting the class with highest mean probability as the most likely class. Other libraries adopt the method of *hard-voting* instead, where the predicted class is determined by the majority vote. Values obtained from the voting can also be used for the estimation of the confidence (see Section 4.4) of individual predictions, although the process behind the computation of such scores is not related to the genuine concept of class probability (Olson and Wyner, 2018). These values are also referred to as *built-in probability scores*.

Random Forest introduces further randomness during the training phase by evaluating subsets of features when splitting nodes. This process is referred to as *feature bagging*. The use of instance and feature bagging is aimed at reducing the correlation between trees to obtain a more robust ensemble of models. The bootstrap sampling is also used to validate the ensemble internally during the training phase. This is achieved by excluding around one-third of training instances from the tree construction, then using these unseen examples as an *out-of-bag* (OOB) validation set. RF is among the

best performing and robust supervised algorithms for classification, it is easy to train and capable of producing estimates of feature importance; however, due to its greater complexity, it is less interpretable and more computational demanding compared to a single decision tree (Alpaydin, 2010).

4.3.2.2. Gradient Boosted Trees

Boosted trees are ensembles of decision trees constructed by means of *gradient boosting*, which consists of fitting new trees on weighted versions of the original training set. The general purpose of gradient boosting is to convert weak learners into strong learners and its principles can be mostly explained by describing the AdaBoost (*adaptive boosting*) algorithm (Freund, Schapire and Abe, 1999): The algorithm begins the growth of the first decision tree by applying equal weights to all observations. The tree is then validated and observations that are hard to classify are assigned with higher weights. The next tree is trained using the weighted data, with the aim of improving the performance on those poorly predicted entries. A new validation is thus performed by combining the predictions of all subsequent trees. The process is repeated for a number of iterations with the aim of constructing an ensemble of trees capable of classifying all instances correctly. The main difference between AdaBoost and Gradient Boosting (GB) relies on the methods they use to determine which sections of decision trees need to be altered for the next iteration. GB is considered one of the most effective off-the-shelf algorithms for classification, yet it is computationally expensive, less interpretable compared to a single decision tree, and can result in over-fitting (Alpaydin, 2010).

4.3.3. Support Vector Machine

Support Vector Machine (SVM) (also known as *kernel machine*) is a non-parametric algorithm which attempts the discrimination between instances of two given classes by finding a hyperplane that maximises the separation between data points (Cortes and Vapnik, 1995). Points that determine the position and orientations of the hyperplane are referred to as *support vectors*, while the other instances are ignored by the algorithm. The dimensions of the hyperplane are determined by the number of features describing

the data (e.g. two features will result in two-dimensional hyperplanes). An example of linear classification in two dimensions using SVM is reported in Figure 4.5:

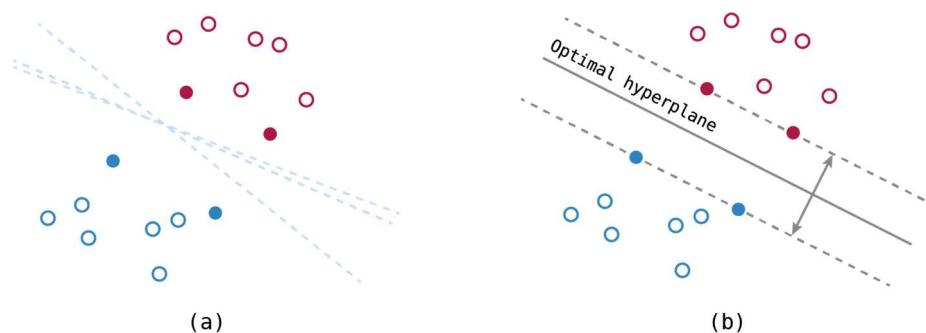


Figure 4.5: SVM linear classification, where support vectors and remaining instances are represented as filled and unfilled points, respectively: (a) a number of separation hyperplanes between the points of two classes can be found; (b) however, the maximum margin of separation is produced only by one optimal hyperplane.

Support vector machines are also particularly effective in performing non-linear classification using the so-called *kernel trick*. This transformation consists of mapping the input space into a higher-dimensional feature space where instances can be linearly separated. An example of the *kernel trick* is described in Figure 4.6. Several types of kernels can be used for this purpose, including linear, polynomial, and the radial basis function, and the selection is generally based on expert knowledge and/or validation.

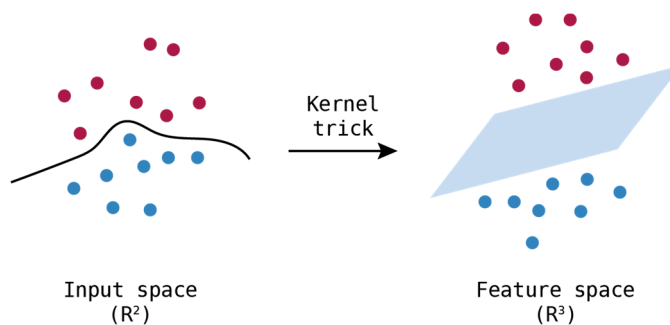


Figure 4.6: Example of kernel trick in binary classification. The left side describes a non-linear separation between points in a two-dimensional (\mathbb{R}^2) space. The right side shows that after the kernel trick, the points have been mapped into a three-dimensional space (\mathbb{R}^3), where an optimal hyperplane of separation can be found.

SVM is another effective supervised algorithm for classification, which can also be trained efficiently since its local functions are determined only by a limited number of

points (i.e., support vectors); however, it often results in long training times using larger datasets and lower performance when dealing with noisy data (Alpaydin, 2010).

4.4. Confidence Estimation using Conformal Prediction

Confidence estimation is a fundamental aspect of machine learning since predictions cannot be considered all reliable in the same way. The quantification of the individual likelihood of predictions enables a safer use of models, in particular in risk-sensitive applications including those concerned with drug discovery. Two established techniques for the statistical estimation of confidence levels in machine learning are Bayesian framework and the theory of Probably Approximated Correct learning (PAC theory). However, the first requires prior knowledge on the data distribution, and the second has been found to perform poorly with noisy datasets (Papadopoulos and Haralambous, 2010). An approach named Conformal Prediction (CP) has been proposed as a method that uses existing data to obtain valid prediction regions for new examples (Vovk, Gammerman and Shafer, 2005). Conformal predictors are built on top of machine learning algorithms (i.e., *underlying algorithms*) to complement predictions with confidence and credibility scores. The validity of such scores is guaranteed provided that the observed data is exchangeable. CP is compatible only with classifiers that are capable of inferring probability scores as well as predicted classes (classification) or values (regression), such as RF or SVM. Conformal predictors share the basic assumption that, in a given training set, described by attribute vectors (x_1, \dots, x_n) and class labels (y_1, \dots, y_n) , all the entries are independent and identically distributed. These methods use the probability scores generated by the *underlying algorithm* to compare the entries to each other and associate each of them with a *nonconformity measure* (*p-value* function) (Vovk, Gammerman and Shafer, 2005). These functions are then used to determine numerically how different the examples are to each other by assigning a nonconformity score (α) to each of them. Therefore, *nonconformity measures* associated with known examples can be used to construct a reference scale for the comparison of *nonconformity measures* of unseen examples. For an unseen example, CP uses all the nonconformity scores to compute a measure of likeliness (*p-value*) for each label contained in the training

data. The calculation of *p-values* is described in Equation 4.1 (Nouretdinov *et al.*, 2001), where ' $p(z_1, \dots, z_n)$ ' is the *p-value* associated with a given class for sequence of entries (z_1, \dots, z_n) , 'n' is the total number of entries, and ' α ' is referred to the nonconformity score associated with each entry:

$$p(z_1, \dots, z_n) = \frac{\#\{i = 1, \dots, n : \alpha_i \geq \alpha_n\}}{n}$$

Equation 4.1: *p-value* calculation in Conformal Prediction.

The concept of *p-value* in CP is practically equivalent to the one expressed in traditional statistics. To facilitate the interpretation of this measure, an example is plotted in Figure 4.7.

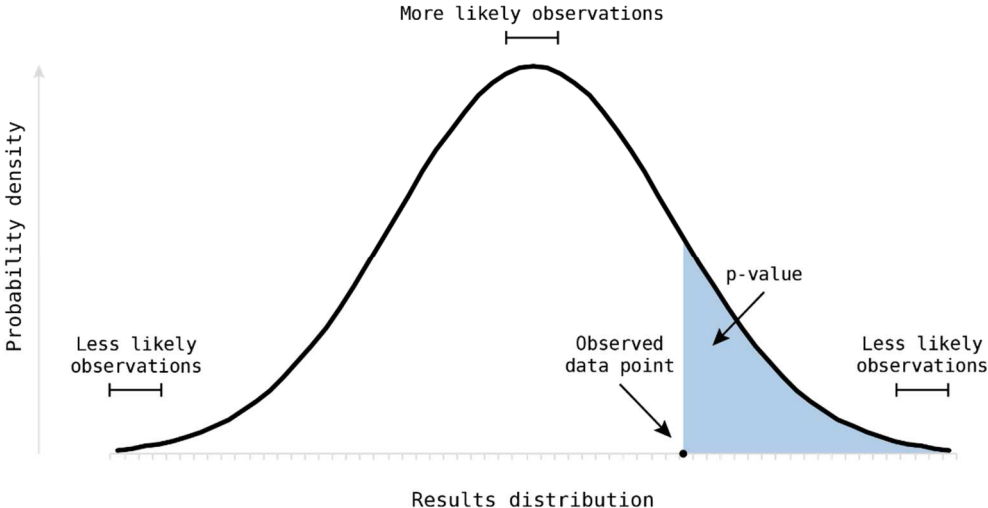


Figure 4.7: Example of a *p-value* calculation. The *p-value* is represented as the area under the curve beyond the observed data point. Image adapted from the literature (Jawlik, 2016).

A higher *p-value* means that an observed data point is closer to the typical observations, whereas a lower *p-value* corresponds to an unlikely observation. Conformal predictors typically output the predicted class as the class that is associated with the highest *p-value*, along with two numerical values that can be used to assess the reliability of the prediction (Vovk, Gammerman and Shafer, 2005): a *confidence* value, which corresponds to the highest *p-value*, and a *credibility* value, which corresponds to 1–2nd highest *p-value*. Therefore, the credibility score indicates the separation between the

class associated with the highest p -value and the class associated with the second highest p -value. The ideal case would be when, for a given instance, the resulting confidence value is high (i.e., the prediction is close to the likely observations) and the credibility score is high as well (i.e., $1 - 2^{\text{nd}}$ highest p -value tends to 1, hence the 2^{nd} highest p -value is very low; thus, the separation between the predicted class and the other classes is large). Another predictive approach is to define an arbitrary significance level that is used as a lower threshold to determine which labels are likely to be true. In this case, the prediction will be a set of labels which are associated with a p -value higher than the significance level. This approach has often been used in drug discovery when labels are limited to a small number or multiple output labels are useful in result evaluation, such as activity or toxicology profiling (Eklund *et al.*, 2015) (Norinder and Boyer, 2016) (Ahlberg *et al.*, 2017).

Conformal Prediction can be divided into Transductive Conformal Prediction (TCP) and Inductive Conformal Prediction (ICP). TCP applies exactly the algorithm as described above, where the entire training set is used directly to derive the prediction on each individual test example. Thus, for each unseen entry, the *underlying algorithm* is applied to the training set and unseen entry to determine all the possible nonconformity scores and the corresponding p -values for each label. In contrast, ICP splits the training data into a *proper training set* and a *calibration set*, then applies the *underlying algorithm* once only to determine a general rule from the *proper training set*. Hence, all the information from the training set is incorporated into this general rule and there is no further use of the training examples. The rule is applied to determine the nonconformity scores of the *calibration set* and the scores of each unseen entry for each label. The scores are then used to compute the p -value for each label. On the one hand, TCP has been demonstrated to have a higher validity compared to ICP due to its higher computational accuracy and because it uses the entire training set to compute the predictions. On the other hand, TCP is very inefficient compared to ICP due to the repeated use of the *underlying algorithm* for the prediction of each example. Several CP algorithms applied for drug discovery purposes have been recently compared by Carlsson

and colleagues (2017). More information about conformal prediction can be found in the literature (Vovk, Gammerman and Shafer, 2005) (Shafer and Vovk, 2007).

4.5. Multi-label Classification

As introduced previously, binary and multi-class classifications are concerned with the prediction of only one label per test instance, whilst multi-label classification provides for multiple labels. Multi-label approaches can also be applied for *label ranking* (LR) with the aim of quantifying the importance of labels in data collections. Due to these characteristics, these methods are established in many fields, including media categorisation (e.g. books, images, movies, music) and medical diagnosis (Tsoumakas and Katakis, 2007). Multi-label problems also differ in the methods used to address them computationally. This subsection focuses on the main approaches for multi-label classification and describes their application in chemoinformatics.

4.5.1. Multi-label Approaches

Multi-label problems can be generally addressed using two different methods: Problem Transformation (PT), where the multi-label problem is converted into a task that can be performed using traditional classifiers such as Random Forests (RF) or Support Vector Machine (SVM); or Algorithm Adaptation (AA), where classifiers are modified to cope with the multi-label nature of the problem. Hybrid combinations of multiple base models are generally indicated as Ensemble Methods (EM), and they are occasionally adopted to obtain better predictive performance compared to the use of the single methods alone (Rokach, 2010) (Rokach, Schclar and Itach, 2014).

4.5.1.1. Problem Transformation

Problem transformation approaches convert the multi-label problem into a framework compatible with traditional supervised classifiers (see Section 4.3). These approaches can be divided into Binary Relevance (BR), Classifier Chain (CC), and Label Powerset (LP). BR and CC share the basic principle by which a given multi-label problem is split into a series of binary problems. BR is the simplest approach since it

uses the training set features to train a separate binary classifier per label, then it merges the predictions for each label into a final output table (Brinker, Fürnkranz and Hüllermeier, 2006). An example of BR transformation is reported in Figure 4.8, which shows that BR decomposes a four-label dataset into four single-label datasets, which are used separately to train four binary classifiers. BR also supports multi-threading since classifiers can be trained and deployed in parallel. However, the simplicity of BR carries a significant drawback: since classes are treated independently, potential label relationships are not accounted for by this approach. For example, in an image classification task, landscapes with the label “sea” will often also contain “beach”; therefore, accounting for the correlation between these two labels can potentially increase the performance in the prediction of unseen landscapes. This relationship is generally indicated as *label dependence* and its implicit modelling can be useful when the selected features are not effective enough to produce good results for all binary problems.

X	Y ₁	Y ₂	Y ₃	Y ₄
x ₁	0	1	1	0
x ₂	1	0	0	0
x ₃	0	1	0	0
x ₄	1	0	0	1
x ₅	0	0	0	1

→

X	Y ₁
x ₁	0
x ₂	1
x ₃	0
x ₄	1
x ₅	0

X	Y ₂
x ₁	1
x ₂	0
x ₃	1
x ₄	0
x ₅	0

X	Y ₃
x ₁	1
x ₂	0
x ₃	0
x ₄	0
x ₅	0

X	Y ₄
x ₁	0
x ₂	0
x ₃	0
x ₄	1
x ₅	1

Figure 4.8: Binary Relevance (BR) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. The original dataset (left) is split into a number of binary sets (right) equal to the number of labels to predict.

CC attempts to account for *label dependence* by creating a directed sequence (‘chain’) of binary classification problems, where predictions are progressively converted into additional features that are used to train the subsequent classifiers. This method relies on the principle of the *Bayesian chain rule* (Read *et al.*, 2009). An example of CC transformation is reported in Figure 4.9, which describes the creation of a directed sequence of classifiers where the order is defined by the label appearance in the dataset. The Y₁ binary classifier is trained only on dataset features, while Y₂ is trained by adding Y₁ predictions as features, and so on. This concatenation results in an increasing bias as

the chain is constructed, hence sequences with different label ordering are expected to produce models with different performances. For this reason, several CC architectures and label-order estimation methods have been developed over time. However, although CC introduces some form of accounting for *label dependence*, this does not guarantee that the model performance will necessarily be better compared to BR since the concatenation can also propagate errors along the chain.

X	Y ₁	Y ₂	Y ₃	Y ₄
x ₁	0	1	1	0
x ₂	1	0	0	0
x ₃	0	1	0	0
x ₄	1	0	0	1
x ₅	0	0	0	1

X	Y ₁
x ₁	0
x ₂	1
x ₃	0
x ₄	1
x ₅	0

X	Y ₁	Y ₂
x ₁	0	1
x ₂	1	0
x ₃	0	1
x ₄	1	0
x ₅	0	0

X	Y ₁	Y ₂	Y ₃
x ₁	0	1	1
x ₂	1	0	0
x ₃	0	1	0
x ₄	1	0	0
x ₅	0	0	0

X	Y ₁	Y ₂	Y ₃	Y ₄
x ₁	0	1	1	0
x ₂	1	0	0	0
x ₃	0	1	0	0
x ₄	1	0	0	1
x ₅	0	0	0	1

Figure 4.9: Classifier Chain (CC) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. The original dataset (left) is split into a number of binary sets (right) equal to the number of labels to predict.

LP deals with the problem by adopting an unconventional strategy. The multi-label problem is converted into multi-class by concatenating individual labels together to form label-sets; hence, the new number of classes in the problem is determined by the unique label-sets in the training data (Boutell *et al.*, 2004). An example of LP transformation is given in Figure 4.10, which shows that the individual labels are condensed to form a single multi-class column that can be used to train a single classifier. Consequently, predictions on new data are generated as label-sets (single labels), which are eventually reconverted into individual labels. On the one hand, this algorithm often performs better than others at modelling *label dependence* since the label merging is particularly effective for incorporating the correlation between classes. On the other hand, this process results in the generation of a much greater number of labels, often yielding highly imbalanced datasets where some minor classes (i.e., infrequent label-sets) are associated with only a few training examples. Hence, although this algorithm can successfully detect patterns across labels, it often leads to data imbalance or overfitting since it cannot predict label-sets that are not present in the training data.

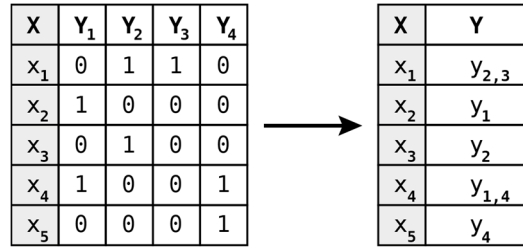


Figure 4.10: Label Powerset (LP) transformation applied to a multi-label dataset, where grey and white columns describe features and labels, respectively. Label columns in the original set (left) are merged together to form a single label column containing multiple classes (right).

Another problem related to the huge number of label-sets that are created during the conversion of the problem, is that they can lead to memory issues. For example, a 50-class multi-label problem can potentially yield a maximum number of 2^{50} label-sets, which corresponds to 1,125,899,906,842,624 classes. In reality, this number is always much smaller but models can still be memory inefficient. For this reason, some modifications of LP have been proposed over time, mainly involving the construction of ensembles of smaller LP models using label subsampling techniques.

4.5.1.2. Algorithm Adaptation

Algorithm adaptation aims at modifying the architectures of traditional algorithms to use them for multi-label classification. These algorithms have often been applied successfully in the past, in particular for text categorisation purposes. McCallum (1999) defined a probabilistic generative model where labels are associated with sets of words, hence text documents are considered as distributions of labels. Schapire and Singer (2000) proposed two multi-label extensions of the algorithm AdaBoost (Schapire and Singer, 1999) (see Section 4.3.2.2), which combines multiple weak models to obtain a robust ensemble. Clare and King (2001) proposed an adaptation of the algorithm C4.5 (Quinlan, 1993), which exploits the concept of *information entropy* for building decision trees. Elisseff and Weston (2001) developed a ranking algorithm using linear SVMs driven by ranking loss as a cost function. However, their algorithm was only capable of producing label rankings rather than an actual classification.

Godbole and Sarawagi (2004) adapted SVM for multi-label classification by running the training in two cycles: the first iteration consists of training individual SVM binary classifiers then inferring predictions on the training data; the second training cycle includes all binary predictions as extra feature columns; hence a new set of binary SVM classifiers are trained using n original features plus l label features. This way, the second generation of binary classifiers are trained by including some information on how labels are correlated. The classification occurs in a similar fashion to the training: the test data is classified using the classifiers from the first training cycle, then predictions are introduced as extra features, and the classification is repeated using the second generation of binary SVMs.

Thabtah, Cowling, and Peng (2004) developed MMAC (multi-class multi-label associative classification), an algorithm based on association rule mining for modelling classification rule sets. The algorithm learns a new association rule, removes the items associated with it, then repeats the operation iteratively until no items are left. Similar rules associated with different labels are merged together as multi-label rules.

Zhang and Zhou (2005) developed a modified version of the kNN lazy learning algorithm for multi-label data, usually referred to as ML-kNN. Similar works based on kNN lazy learning have also been proposed over time (Luo and Zincir-Heywood, 2005) (Wieczorkowska, Synak and Raś, 2006) (Zhang and Zhou, 2007) (Spyromitros, Tsoumakas and Vlahavas, 2008) (Lin and Chen, 2010) (Brinker and Neubauer, 2010). These methods generally apply kNN independently for each label as follows. For a given test entry, the algorithm determines k neighbours based upon pair-wise Euclidean distances, then one label at the time is evaluated using probabilities similarly to binary classification. An ML-kNN adaptation of the example in Figure 4.3 is described in Figure 4.11, which illustrates two examples of multi-label classification using $k=3$ and $k=5$, respectively. Lazy learning has also been integrated within associative learning algorithms. For example, Veloso and colleagues (2007) proposed a hybrid approach where association rules are learned by the algorithm, then lazy learning is applied in the classification stage to improve the classification ability of the model.

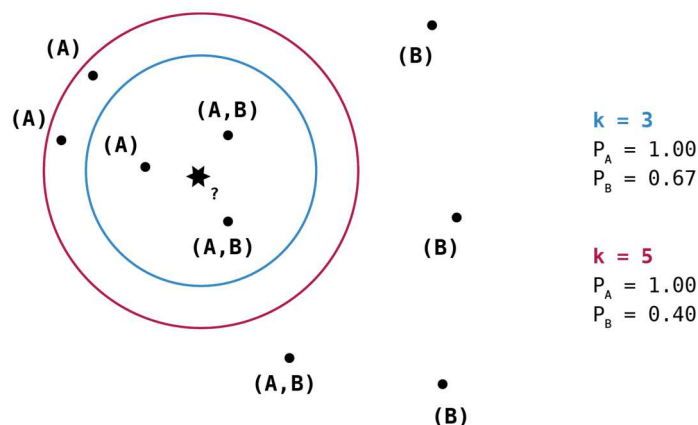


Figure 4.11: ML-kNN classifications ($k = 3, 5$) on a binary dataset (A, B). The test entry is described as a star while training instances are points with their corresponding classes reported in brackets. Frequencies (P_A, P_B) are calculated as ratios between particular instances (e.g. A) on the total number of instances determined by k similarly to binary or multi-class problems, yet in multi-label classification, the same instances can contribute to increasing the frequencies of more than one label.

4.5.1.3. Ensemble Methods

Ensemble methods are derived from the combination of multiple base approaches in order to increase the performance of the model. For example, Random Forests (RF) is a famous example of an ensemble approach for binary or multi-class classification where multiple decision trees are combined to reduce the risk of overfitting and increase the reliability of predictions.

In multi-label learning, two well-established methods are Ensembles of Pruned Sets (EPS) (Read, Pfahringer and Holmes, 2008) and Random k-Labelsets (RAkEL) (Tsoumakas, Katakis and Vlahavas, 2011). Both methods are extensions of LP, where combinations of labels (label-sets) are merged to form a set of single labels that can be predicted using any multi-class classifier. As reported previously, LP brings some disadvantages such as memory inefficiency and overfitting; hence, EPS and RAkEL have been proposed to overcome the issues related to the LP approach.

EPS decomposes infrequent label-sets to obtain subsets that are likely to be more frequent. For example, the label-set {"Beach", "Sea", "Boat"} would be decomposed

into {"Beach", "Sea"} and {"Sea", "Boat"}. Consequently, an ensemble of multi-class classifiers is trained using different subsets of the training data in order to obtain a more robust model. Therefore, EPS focuses only on obtaining more frequent label-sets in order to improve the model accuracy while reducing the problem complexity. RAKEL works similarly to EPS but fixes the label-set decomposition to a specified number of labels. Consequently, an ensemble of LP classifiers is constructed using random subsets of k labels from the original dataset. In both EPS and RAKEL, the training is computationally less demanding than LP and the label-set example distribution is less skewed. Consequently, the classification of unseen data is performed by averaging the predictions from the trees of the ensemble.

4.5.2. Multi-label Classification in Chemoinformatics

Multi-label classification has been applied successfully for drug discovery purposes, and its chemoinformatics applications involve the same principles adopted in other fields. The input data (e.g. molecular structures) has to be associated with multiple outputs (e.g. activities). For this reason, most of the methods proposed so far have been concerned with the pharmacological (e.g. biological targets) profiling of compounds.

Kawai and colleagues (2008) developed an activity profiling approach based on the topological fragment spectra (TFS) method proposed by Takahashi (1998). Molecules were first described using substructure-count fingerprints, then their profiles were determined on 100 drug activities (e.g. antihypertensive) using an ensemble of SVM classifiers. Kawai and Takahashi (2009) also reported a specific application of the same method for the identification of dual action antihypertensive drugs. A few years later, Zhang and colleagues (2015) developed a multi-label approach for the prediction of drug side effects based on similar principles, yet relying on the use of ML-kNN.

Michielan and colleagues (2009a) proposed an alternative method based on specific multi-target affinity prediction instead of activity labelling. Their method involved the use of auto-correlated molecular electrostatic potential (autoMEP) descriptors combined with an ensemble of SVMs trained on four Human Adenosine Receptor (hAR) subtypes.

Later, Afzal and colleagues (2015) presented two much wider approaches for ligand promiscuity identification across more than 300 biological targets using binary and multi-label Naïve Bayes classification.

Michielan and colleagues (2009b) also reported a comparison between single- and multi-label classification for selectivity prediction on five and seven members of the Cytochrome P450 (CYP) family. Zhang and colleagues (2012) described another application of multi-label learning on four CYP450 isoforms. Their approach consisted of integrating DTs with genetic algorithms for automated feature selection to yield a framework extendible to more CYP isoforms and larger training sets.

Montanari and colleagues (2016) and Aniceto and colleagues (2016) independently presented two methods for efflux prediction by ATP-binding cassette (ABC) transporters. Montanari and colleagues' approach focused on BCRP1 and Pgp/MDR1 transporters, while Aniceto and colleagues extended the ensemble capabilities to BCRP1, Pgp/MDR1, MRP1 and MRP2.

Finally, multi-label classification has also been applied to chemoinformatics data for purposes beyond pharmacology profiling. For example, Hristozov and colleagues (2008) presented two multi-label approaches for the identification of possible plant sources for an important class of natural products, showing that models were capable of assigning compounds to their corresponding sources according to skeletal types and substitutional patterns of structures. More applications have been described using bioinformatics data, for example, for gene (Barutcuoglu, Schapire and Troyanskaya, 2006) and protein function prediction (Alves, Delgado and Freitas, 2008) (Otero, Freitas and Johnson, 2010) (Yu *et al.*, 2012), and for the identification of protein subcellular locations (Zhu, Yang and Shen, 2009) (Chou, Wu and Xiao, 2011).

4.6. Conclusions

In this chapter, the main concepts of supervised machine learning have been described by focusing on the application of algorithms for classification purposes. Classification problems have been distinguished on their nature, then some traditional

algorithms have been described on their functioning and inductive bias. The chapter has also discussed the importance of confidence estimation and the principles of Conformal Prediction. Following this, the chapter has illustrated the approaches used to address multi-label problems computationally, also reporting some examples of their application in drug discovery. The next chapter describes the implementation of some algorithms for reaction data standardisation and their application for generating a series of datasets used in this work.

Chapter 5: Reaction Data

5.1. Introduction

The importance of gathering bigger and more representative chemical and biological data collections for drug discovery purposes, has been increasingly recognised in the last years (Wassermann *et al.*, 2015) (Agatonovic-Kustrin and Morton, 2016) (Lo *et al.*, 2018) (Brown *et al.*, 2018). Reaction data also represents a potential source of information that can be used to drive decision making in this field. Reaction sets can be, for example, processed for their use in machine learning for drug design or for the development of new methodologies based on the concept of the reaction vector. Nevertheless, most of the accessible reaction data is not curated enough for these purposes, hence the implementation of techniques for reaction standardisation constitutes a requirement to increase and level the quality of datasets.

In this chapter, a collection of methods for reaction standardisation, fingerprinting, and reaction vector database encoding are first introduced along with their implementation in a graphical-user-interface (GUI). The methods are then applied to several datasets obtained from different sources and which are used in the experimental chapters of the thesis. The US pharmaceutical patents datasets represent the main source of reaction data used in this work. Three additional collections derived from journals and industrial data are also introduced as external datasets. The aim of this chapter is to highlight the importance of reaction standardisation, to provide preliminary results on the composition of the selected collections, and to produce a series of corresponding fingerprint datasets and reaction vector databases for data learning and reaction-based *de novo* design, respectively. Special attention is also given to the reaction class composition during the pre-processing of these collections in order to provide some background information for the following chapters of this thesis, which are concerned with the development of machine learning methods for reaction classification and reaction class recommendation. In this regard, a tailored reaction class labelling system is also introduced and discussed.

5.2. Reaction Standardisation

5.2.1. Introduction

Chemical reactions are generally encoded using the SMILES notation where reagents, agents, and products appear on left, middle, and right, respectively, separated by a “>” symbol (see Section 1.3). The term *agent* refers to those structures that do not actually take part in the transformations, such as catalysts or solvents. These structures are ignored by the reaction vector algorithms developed at Sheffield, which only account for atoms and bonds that change in reactions. In addition to this, since reaction data is rarely curated, transformations are often imbalanced because chemists tend to draw them without considering the reaction stoichiometry; hence, sub-products are often omitted or multiple products are written as results of the same transformation instead of being reported in two separate entries. Due to these inconsistencies, a reaction standardisation workflow is developed with particular attention on producing clean, balanced, and indexed entries, which can be subsequently checked for duplicates according to their reaction centres in order to obtain sets of ‘unique’ reaction vectors. This last operation consists of filtering out entries by only accounting for the structural parts that are involved in the transformations, hence removing all the redundant reaction centres to reduce the size of reaction datasets.

5.2.2. Methods

The reaction standardisation procedure can be split into four main steps: *unmapped compound removal*, *balancing*, *indexing*, and *duplicate filtering*. These processes are described in detail below.

The *unmapped compound removal* consists of processing the entries using a reaction mapping algorithm (Chen, Chen and Taylor, 2013) to retain only the structures that have mapped atoms. The mapping algorithm used is the Indigo Reaction Automapper node in the Indigo Toolkit in the KNIME Analytics Platform (EPAM, 2017). This way, compounds such as solvents or catalysts, which have no involvement in the reaction centre, are filtered out. An example of unmapped compound removal is reported for a

reaction taken from the US patent data (USPD) in Figure 5.1, which describes the removal of solvents, ions, and catalysts by filtering out the structures that have not been detected by the mapping algorithm. On the one hand, this operation leads to a partial loss of information since these structures could be used to further describe the transformation. On the other hand, agents have no function in the *de novo* design algorithms developed at Sheffield, thus they only would require extra memory for their storage as well as potentially compromise the correct generation of reaction vectors.

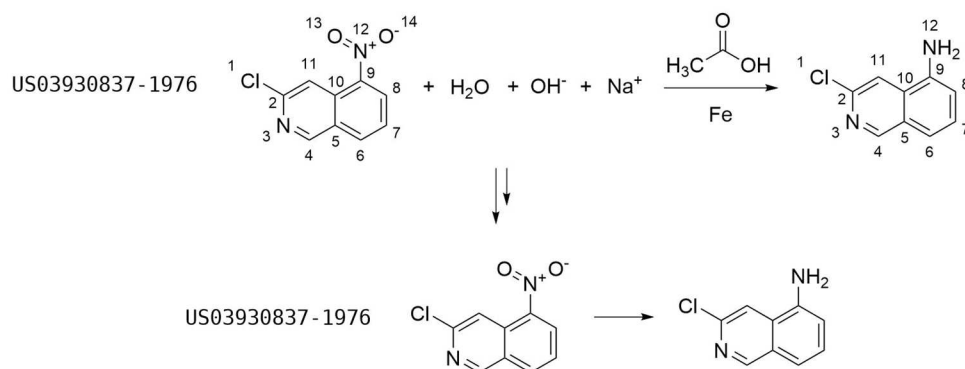


Figure 5.1: Unmapped compound removal.

The *balancing* step consists of applying the Reaction Balancing Tool (Patel, 2009) (Wallace, 2016) implemented in the Sheffield Chemoinformatics package in the KNIME Analytics Platform. The importance of obtaining balanced reactions relates to the concept of the reaction vector, which encodes the difference between products and reactants as a set of atom pairs associated with negative or positive integers, depending on if they are lost or gained during the reaction, respectively. Structures or fragments that do not change must always be completely reported on both sides of the reaction, otherwise their features would be encoded in the reaction vector, thus producing an incorrect description of the transformation.

The balancing tool operates so that missing substructures are generated from the comparison between reactant and product species in order to obtain the same number of carbon atoms on both sides of the reaction (Patel *et al.*, 2009). Entries which do not achieve the carbon balance after the application of the tool are simply filtered out. An

example of reaction balancing for missing fragments is reported for a USPD reaction in Figure 5.2, where the missing substructure is highlighted in bold:

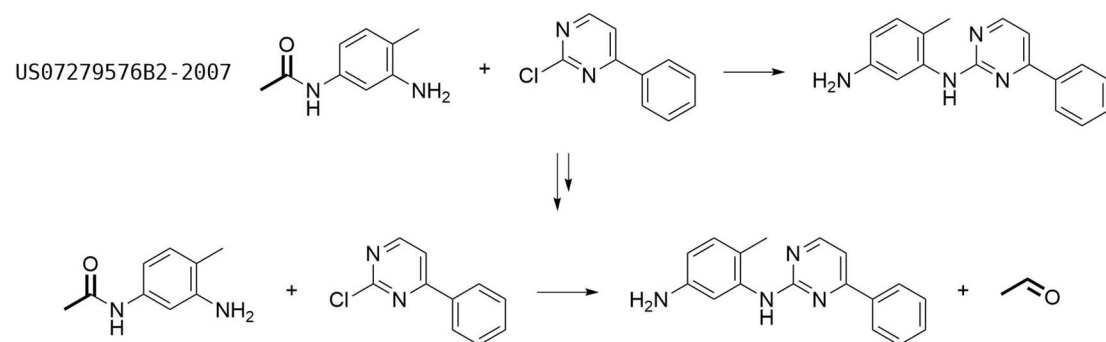


Figure 5.2: Reaction balancing for missing fragments.

A similar issue can also be caused by reactions that generate multiple products such as isomers, where the products are generally reported together instead of separated into two distinct single-product reactions, thus resulting in an incorrect representation. An example of balancing for multi-product transformations is reported in Figure 5.3:

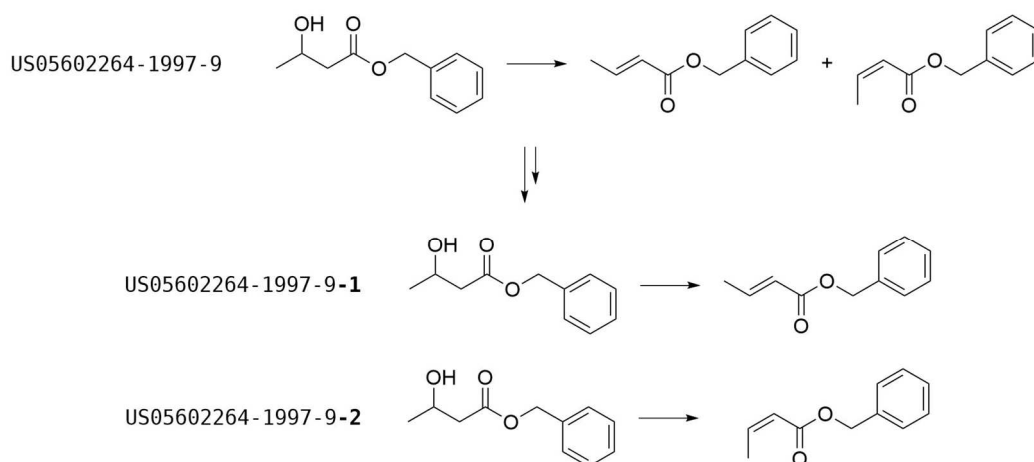


Figure 5.3: Reaction balancing for multi-product transformations.

Figure 5.3 also shows that the original entry index is regenerated after the balancing. This is particularly crucial for dataset analysis or information retrieval for compound synthesis in *de novo* design. Another example of reaction indexing is reported in Figure 5.4, which describes the indexing of three reactions that are all different although they are associated with the same patent reference, thus resulting in an ambiguous identification. The indexing tool adds a sequential numbering after the patent

number to ensure that the new identifiers are related only to one reaction example, while preserving the patent information.

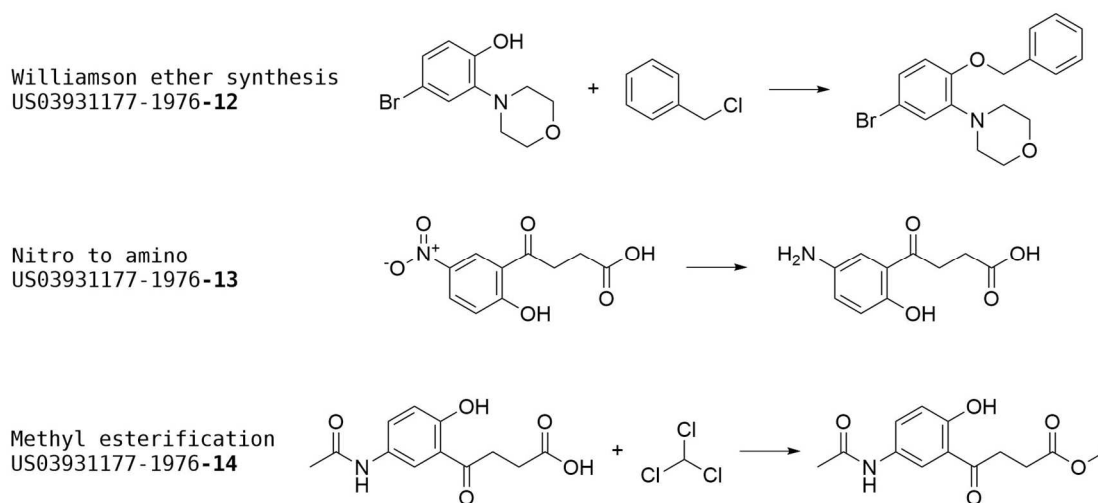


Figure 5.4: Reaction indexing of three different examples from the same patent.

The *duplicate filtering* relies on the generation of reaction vector strings, which describe canonically the topological features that change during transformations using a modified atom-pair notation developed at Sheffield (Patel *et al.*, 2009), and the consequent removal of the duplicate entries that are represented by the same vectors. This process can be modulated by specifying on which atom-pair level (e.g. AP2, AP3, AP2+AP3, etc.) the duplicates are filtered: The less reaction environment is included, the more duplicates will be likely to be filtered out. An example of reactions that are associated with the same AP2 level and different AP2+AP3 levels is reported in Figure 5.5. Therefore, the selection of different atom-pair levels enables the creation of reaction datasets of different sizes and content.

Note that the reaction vector-based structure generation algorithm uses AP2+AP3 vectors as references for the design process (see Section 3.3), thus encoding multiple reactions that describe the same AP2+AP3 level does not bring any advantage in terms of content in the reaction database, except for the inclusion of multiple synthetic references. For this reason, AP2+AP3 was selected as a default setting for the duplicate filtering process.

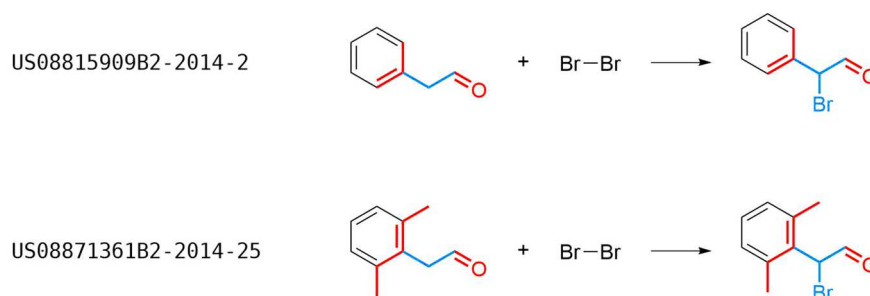


Figure 5.5: Examples of reactions associated with the same AP2 vectors and different AP2+AP3 vectors. AP2 reaction centres are coloured in blue and AP3 extensions are coloured in red.

5.2.3. KNIME Implementation

Chemoinformatics algorithms are developed on a daily basis, yet most of them are often not readily usable by users that do not have a solid background in computer science, thus they cannot benefit most of the people working in drug discovery. For this reason, the implementation of the algorithms described in this thesis in a user-friendly environment is central to the purpose of this project.

KNIME Analytics Platform is open-source software (<https://www.knime.com/>) where a graphical-user-interface is built on the top of a Java-based programming environment. Functions are implemented within ‘nodes’, which can be linked to each other using ‘connectors’ in order to generate data flows. This way, series of nodes can be configured in order to create automated workflows that can be reused with different data sources. Furthermore, KNIME offers a stable Python integration; therefore, adapting external scripts within a workflow is straightforward. In this subsection, the implementation of a reaction standardisation algorithm in KNIME is reported. A description of the workflow is shown in Figure 5.6.

The workflow is divided into four main blocks as described in Section 5.2.2. Each node implements a pipeline segment that is capable of catching a given reaction SMILES input, process it, then merge it back with the rest of the information. The “Duplicate Filtering” node can also be configured in order to filter out reaction vector duplicates

according to a specified level of atom pairs (e.g. AP2, AP2+AP3, AP4, etc.) as explained in Section 5.2.2.

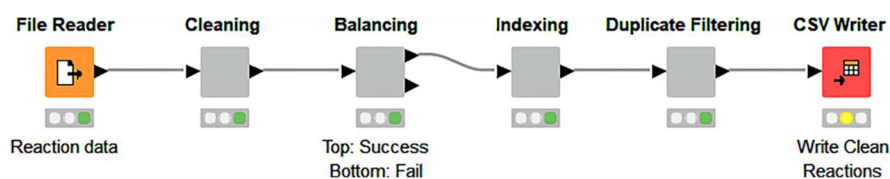


Figure 5.6: Reaction standardisation KNIME workflow.

5.3. Dynamic Reaction Fingerprints

5.3.1. Introduction

Machine learning requires the input data to be in a structured format (e.g. molecular fingerprints). As shown in the top left of Figure 5.7, reactions vectors are normally generated by the Sheffield algorithms as strings, hence they are not compatible with data learning algorithms. For this reason, the implementation of a string-to-fingerprint conversion workflow is described in this section.

5.3.2. Method

The algorithm works by first separating atom-pair types and values, then by rearranging atom pairs into columns and filling cells with their corresponding values. Missing values are replaced by zeros since they describe non-changing atom pairs, and columns are sorted alphabetically for canonicalisation purposes. A simplified scheme of the fingerprint conversion procedure is reported in Figure 5.7. The reaction fingerprint is referred to as *dynamic* since the number of columns is determined by the particular dataset and represents the minimum number of atom pairs necessary to fully describe the reactions within the dataset. Thus, the conversion of different datasets will return different numbers and types of atom pairs. On the one hand, this technique enables the direct conversion of string reaction vectors without losing or simplifying any atom-pair, while also minimising the amount of memory allocation necessary to store the fingerprint dataset. On the other hand, datasets described by different atom pairs will not be

directly comparable, thus they have to be adapted to each other in order to generate tables with the same number, type and order of columns.

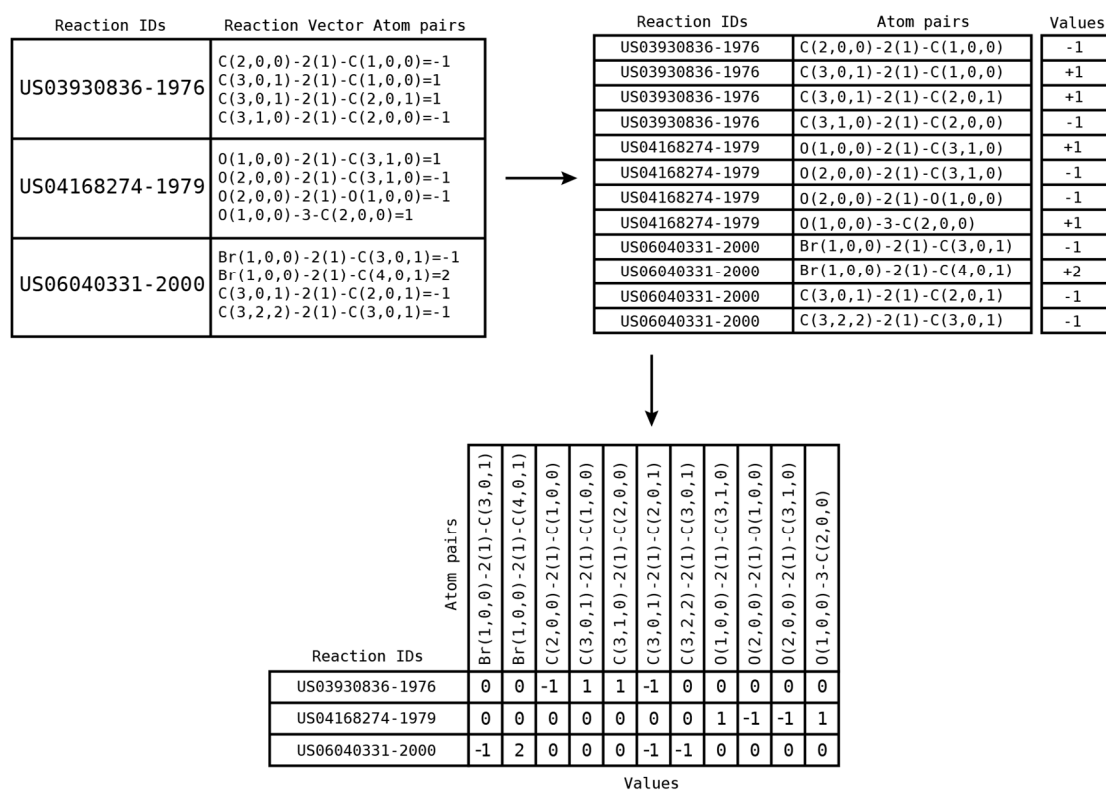


Figure 5.7: Dynamic fingerprint conversion algorithm. Reaction vectors represented as strings are converted to true vectors. The vector elements are integers with negative values indicating atom pairs that are lost from the reactants; positive values indicating atom pairs that are gained in the products; and zeros indicating atom pairs that are not present in the vector.

For supervised machine learning, the dataset adjustment can be carried out using the training data atom pairs as references and making adjustments to the test data. Test data atom pairs not included in the training data are filtered out because they are not accounted by the model; and training data atom pairs not included in the test data are simply added to the test data with cells filled with zeros since they represent non-changing features. An example of reaction vector dataset adjustment is given in Figure 5.8, where training and test datasets are indicated as *master* and *slave*, respectively. The KNIME implementation of the dynamic reaction fingerprint conversion and adaptation algorithms is described in Chapter 7.

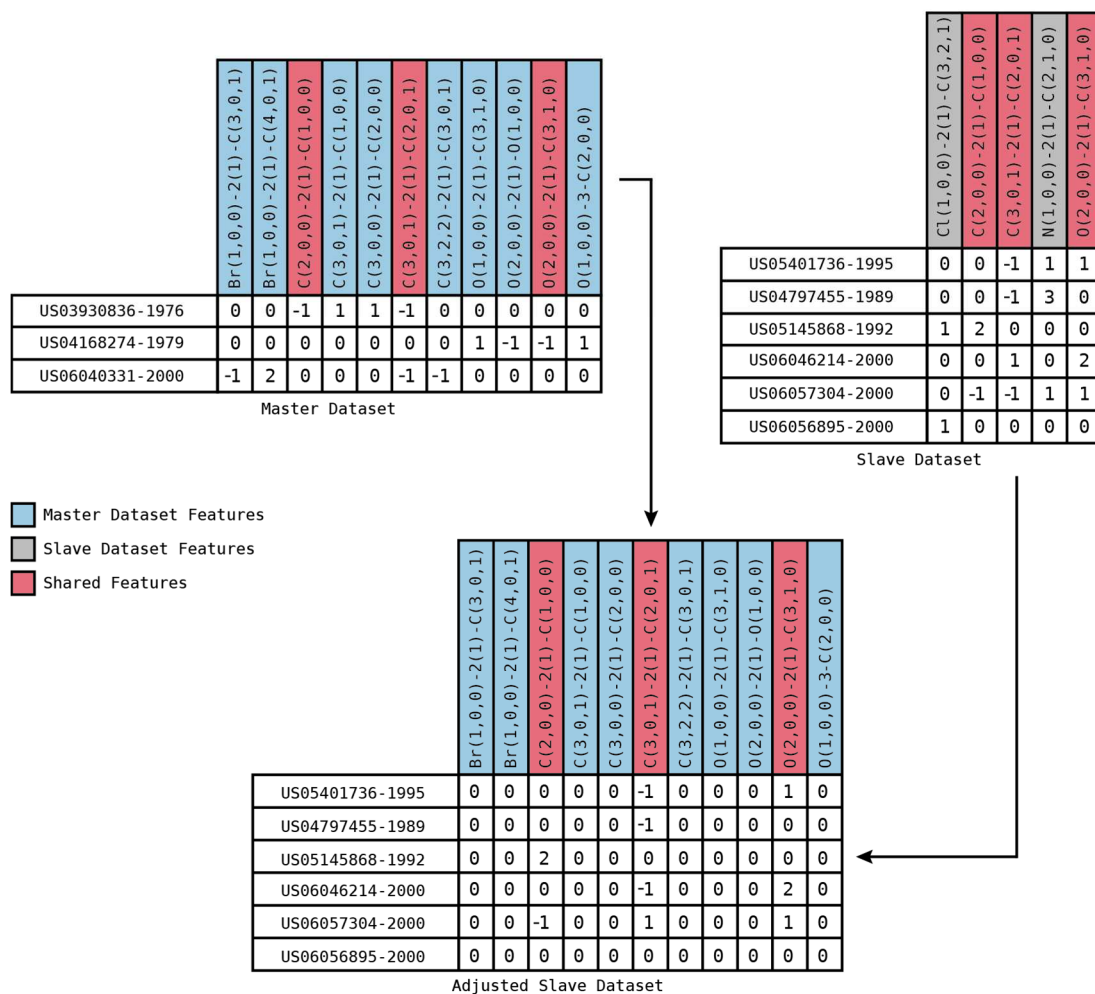


Figure 5.8: Reaction vector dataset adjustment: blue and red features are retained in the *adjusted slave* dataset, while grey features are discarded since they are not described in the *master* dataset.

5.4. US Pharmaceutical Patents

5.4.1. Introduction

Although commercial reaction databases, such as CASREACT and Reaxys (see Section 1.4) contain millions of organic reaction examples, the lack of publicly available reaction data sources has possibly slowed down the development of methods based on reaction data. The recent publication of several datasets of reaction examples text-mined from the United States pharmaceutical patents (NextMove Software, 2014) (Lowe, 2017) has remarkably contributed to several publications in the field of chemoinformatics.

Schneider and colleagues (2015) first proposed a novel reaction fingerprint that could also account for the presence of agents, which was validated for reaction classification purposes using traditional machine learning on both patent data and in-house reactions. Schneider and colleagues (2016a) (2016b) then analysed the patent content and proposed a data-driven method for assigning roles to reaction components (e.g. ‘reagent’), which was eventually validated on a large subset of the patent data.

Later, more sophisticated machine learning architectures have been trained with the patent data for different purposes: Nam and Kim (2016) and Coley and colleagues (2017) developed two forward prediction models for reaction outcome estimation based on RNNs (*sequence-to-sequence*) and fully connected networks, respectively, while Liu and colleagues (2017) validated an RNN-based retrosynthetic reaction prediction model. Schwaller and colleagues (2018a) (2018b) followed with two distinct forward prediction frameworks using sequence-to-sequence and transformer architectures, respectively. In addition, as reported in Section 1.6.2, Schwaller and colleagues (2019) has recently proposed a reaction classification model using on attention-based neural networks.

The US patent datasets represent the largest publicly available source of reaction data and they are used extensively throughout this thesis. Among several text-mined US patent data (USPD) collections released publicly in the last years, two datasets were selected here because reaction class information was available only for these:

- USPD Grants 1976-2016 (also referred to as USPD);
- USPD Applications 2001-2016 (also referred to as USPDA).

Patents are mainly divided into two typologies: (1) *applications*, which are requests pending at the US Patent and Trademark Office (USPTO) (USPTO, 1994) for the grants for the inventions, and (2) *grants*, which are successful prosecutions of patent applications. Applications are identified by a number format USYYYY/XXXXXX AX, where Y corresponds to the year and X to a serial number, whereas grants are identified by the format USXX/XXXXXX, where X is a serial number. In this section, the standardisation, encoding, and validation for *de novo* design of the selected patent

collections are described. In addition, a new reaction class labelling system is introduced. The aim of these operations is to produce data compatible with the algorithms that have been developed at Sheffield, also with a particular focus on obtaining classified datasets for the next experiments described in this work.

5.4.2. Standardisation

The standardisation of the USPD and USPDA sets is reported as follows. Datasets were first filtered by retaining only the entries for which classification data was available. Classification data was originally generated by NextMove Software using NameRxn (version 2.0) (NextMove Software, 2017). NameRxn adopts a nomenclature that is inspired by the RXNO Ontology developed by the Royal Society of Chemistry (RSC) (Royal Society of Chemistry, 2017) and earlier classification system proposals (Carey *et al.*, 2006) (Roughley and Jordan, 2011), whereby reactions are named using three levels of descriptions: major classes, subclasses (also indicated as Categories), and reaction types (also indicated as Classes). Major classes were not considered in this study. Reactions containing more than six reactants and/or products were filtered out since the Indigo Reaction Automapper is likely to time out on these examples, and also the algorithms developed at Sheffield do not support reactions with more than six components per side of reaction; the remaining entries were processed through the reaction standardisation workflow described in Section 5.2.2. Reaction vector duplicates were filtered by their AP2+AP3 centres.

Results for the USPD and USPDA datasets are reported in Table 5.1 and Table 5.2, respectively, which describe similar trends. A remarkable reduction in the number of entries is observed by retaining only the entries classified by NameRxn. More specifically, ~33% and 30% of the total entries were filtered out from the USPD and USPDA datasets, respectively. A similar result (i.e., 36% of unclassified entries) was reported by Schneider and colleagues (2016) in the classification of the US pharmaceutical patents published between the years 1976 and 2015. A smaller number of entries was rejected by the six reactants/products filter. In particular, 66,143 and 75,485 entries were filtered out from USPD and USPDA datasets, respectively. In both

cases, more than 99% of the entries were rejected because they contained more than six reactants. The manual inspection of the filtered reactions revealed that many of these contained agents on the reactant side, instead of reporting them on the top of the arrows. The six reactants/products filtering also reduced the total number of reaction classes in each set. The inspection of the results revealed that the removed classes were populated by low numbers of examples. Consequently, the balancing tool rejected 31,936 and 38,880 entries from USPD and USPDA datasets, respectively. These numbers cannot be directly calculated from Table 5.1 and Table 5.2 since the balancing tool also increased the total number of entries by splitting the multi-product reactions into multiple single-step reactions. The inspection of the entries filtered by the balancing tool did not reveal any particular class susceptible to rejection, although a high presence of reactions involving symmetric heterocycles was found among the examples filtered out. To conclude, a large reduction in size was found for both datasets after the duplicate filtering at the AP2+AP3 level: USPD and USPDA were reduced by 90% and 91%, respectively.

USPD	Reactions	Categories	Classes
Original Data	1,808,937	64	753
Only Classified Data	1,215,355	64	753
Six Reactants/Products Filtering	1,149,212	64	751
Balancing Tool	1,114,953	64	735
Duplicate Filtering	115,602	64	727

Table 5.1: USPD dataset description through the standardisation workflow.

USPDA	Reactions	Categories	Classes
Original Data	1,939,253	65	749
Only Classified Data	1,374,294	64	748
Six Reactants/Products Filtering	1,298,809	64	745
Balancing Tool	1,263,602	64	727
Duplicate Filtering	110,802	64	718

Table 5.2: USPDA dataset description through the standardisation workflow.

These results indicate that both collections originally contained a very high redundancy in reaction centres, although the inclusion of the AP3 level should have promoted the discrimination between entries. The USPDA set yielded a filtered dataset of smaller size compared to the filtered USPD collection, although USPDA originally contained several thousand more entries. The high redundancy in reaction centres can

be traced back to the nature of pharmaceutical patents. Patents are aimed at covering exhaustively certain regions of the chemical space, often by combining similar molecules with similar reagents; therefore, they are expected to produce a high redundancy in the reaction space as well. This reduction can be better visualised by plotting the ratios of redundant examples per reaction vector in the two datasets prior to their duplicate filtering. Results are reported in Figure 5.9, which describes very imbalanced distributions of examples per reaction vector for the two datasets. Very small numbers of reaction vectors are associated with thousands of reaction examples, whereas the rest of the populations are associated with fewer than 10 examples. In particular, $\sim 70\text{-}75\%$ of the entries are associated with 5 or fewer reaction examples in both datasets, and $\sim 40\%$ are associated with only 1 example. Therefore, the filtering process removed all redundant examples associated with the same reaction vectors since they did not contribute to increasing the diversity of the datasets.

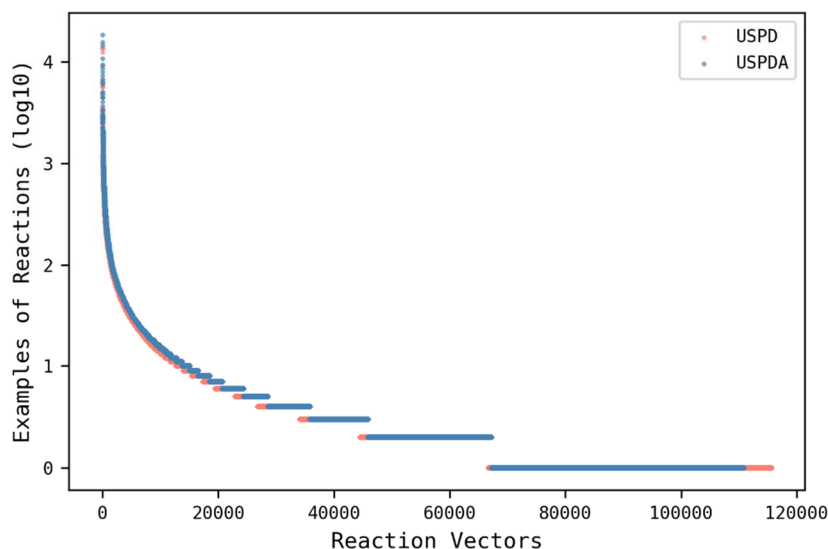


Figure 5.9: Reactions per reaction vector ratios expressed in log10 scale for the USPD and USPDA datasets before the duplicate filtering. Vectors are sorted by descending order according to their numbers of examples.

The duplicate filtering also reduced the total number of reaction classes in each collection: 8 and 9 reaction classes disappeared from USPD and USPDA datasets after the filtering, respectively. The manual inspection of the results revealed that the filtered classes represented examples of transformations that are distinguishable only on their

agents. As reported previously, agents are removed by the unmapped compound removal tool since they do not really take place in the transformations. These results suggest that the first step of the cleaning workflow possibly resulted in the creation of a number of different classes sharing identical vectors. An example is reported in Figure 5.10 for the “Zinin Reduction” and “Nitro to Amino Reduction” classes, which became indistinguishable after the unmapped compound removal.

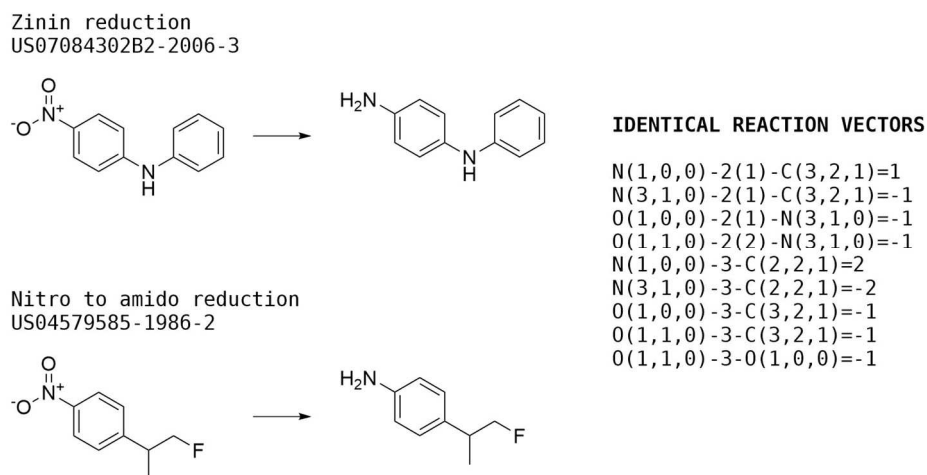


Figure 5.10: Example of indistinguishable reaction classes after the duplicate filtering.

This simplification is acceptable for the purposes of this work, which is not concerned with investigations on molecular reactivity or reaction mechanisms, hence do not require full information on the reactions. However, in addition to being formally incorrect, the presence of clashing classes is not ideal when preparing the data for use in supervised machine learning: identical data points associated with different classes are likely to cause difficulties to classifiers during the training phase since algorithms generally attempt to discriminate instances belonging to different classes. Mainly for this reason, a more suitable labelling system is introduced and discussed in Section 5.4.6.

Reaction classes were further investigated by determining statistics on the reaction vectors per class across the two filtered datasets. Results are reported in Table 5.3, which describes similar statistics for the two datasets where the difference between mean and median values indicates that only a few classes are associated with large numbers of examples, hence suggesting the presence of imbalanced classes.

Clean Dataset	Number of Vectors per Class			
	Min	Max	Mean	Median
USPD	1	4,335	159.0	35
USPDA	1	4,390	154.3	33

Table 5.3: Reaction vectors per class statistics for the filtered USPD and USPDA sets.

Distributions of vectors per class across the two filtered datasets are also plotted in Figure 5.11, which confirms the trends suggested by Table 5.3 where only a few classes are densely populated. Further investigations revealed that in both datasets less than 5% of the classes are associated with more than a thousand vectors, thus evidencing the presence of very imbalanced data. Imbalanced datasets have been extensively investigated in data learning in the last twenty years since they often lead to inaccurate results for the minority classes (Witten *et al.*, 2016). For this reason, class imbalance has to be taken into account for those particular uses. However, these sets can still be applied without any particular precaution for *de novo* design, where imbalanced data does not necessarily constitute an issue.

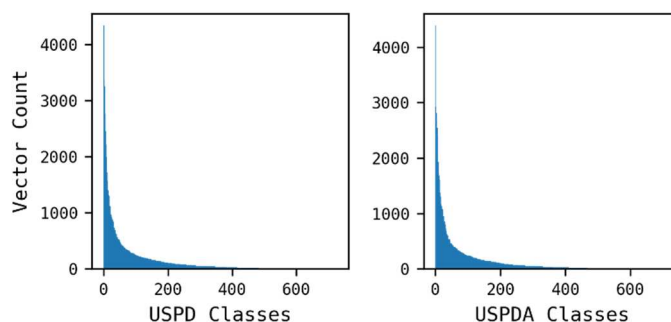


Figure 5.11: Reaction vector class distributions for the filtered USPD and USPDA sets.

Classes are sorted by descending order according to their numbers of vectors.

5.4.3. Dataset Intersections

As previously reported, chemical reactions are commonly stored in files as SMILES strings using some extra signs to separate the components and indicate reaction arrows. This representation system has no defined canonicalisation rules for reaction components, thus SMILES strings do not generally constitute a good representation for reaction search or comparison. For example, the SMILES for the generic reaction $A+B\rightarrow C$ is different from the SMILES $B+A\rightarrow C$, although they both represent the same

transformation. In addition to this, molecular SMILES are often generated using different canonicalisation methods, hence identical compounds represented using different SMILES encoding systems could be different from each other, further increasing the difficulty in the comparison between reactions. Reaction vectors provide a partial solution to these issues since they are generated canonically and they can be compared at different levels of extended reaction environments, provided that the entries to analyse are balanced. Therefore, the analysis of the intersections between USPD and USPDA datasets along the standardisation workflow was carried out to determine the relationship between the collections. The analysis was performed on some of the datasets reported in Table 5.1 and Table 5.2 as follows.

The “Only Classified Data” USPD and USPDA datasets were first compared according to their reaction SMILES and reaction vectors. These datasets were selected since they do not contain any unclassified data, thus their overlap can be compared with results from the data processed in the later stages of the standardisation pipeline. Two Venn diagrams describing SMILES and AP2+AP3 vectors intersections of the filtered USPD and USPDA datasets are reported in Figure 5.12.

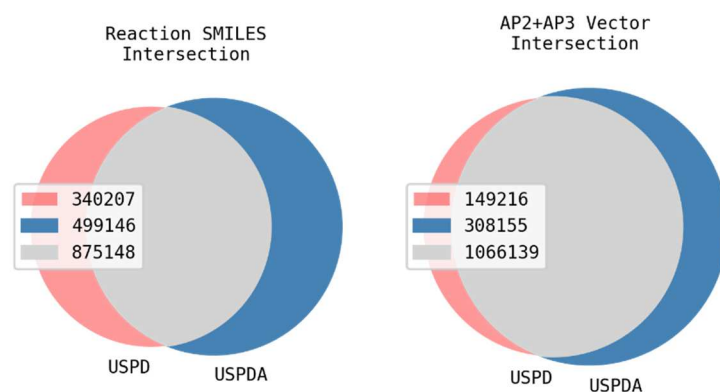


Figure 5.12: “Only Classified Data” datasets intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.

Both diagrams in Figure 5.12 show high percentages of intersection between the two datasets. In particular, USPD and USPDA reported 72% and 64% of reaction SMILES overlapping, then 88% and 78% of AP2+AP3 vector overlapping, respectively. The high intersection in reaction SMILES can be rationalised in two ways. First, the

two datasets are related to the same source (i.e., patents), thus they are expected to share very similar contents, although as commented previously, not all the patent applications always become grants and the USPDA dataset covers a limited range of time. Second, the collections were text-mined using the same software and version, hence they were encoded using the same rules, which is the equivalent of a relative canonicalisation. The comparison between reaction SMILES and vector intersections in Figure 5.12 also suggests higher suitability of reaction vectors for analytic purposes, since vectors reported ~10% more of overlapping entries compared to reaction SMILES. This increase can be explained by the canonical nature of reaction vectors and by the fact that they describe limited reaction environments instead of representing whole molecular structures unlike SMILES.

Successively, the “Balancing Tool” datasets were compared with each other using the same technique. These datasets represent cleaned and balanced versions of the patent collections, yet they still have to be filtered by reaction vector duplicates. Intersection diagrams are reported for these sets in Figure 5.13, which shows a similar reaction SMILES intersection and increasing overlap between AP2+AP3 vectors compared to the analysis of Figure 5.12.

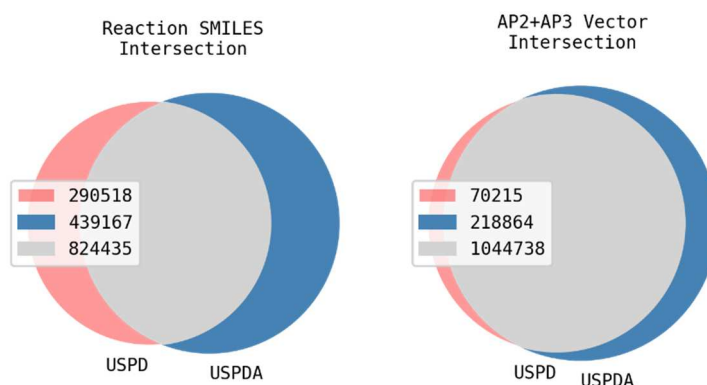


Figure 5.13: “Balancing Tool” dataset intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.

More specifically, USPD and USPDA reported 74% and 65% of SMILES overlapping, then 94% and 83% of AP2+AP3 vector overlapping, respectively. This is because the number of intersecting vectors in Figure 5.13 reported a lower reduction

compared to the other elements in the diagrams. These trends are related to the use of the balancing tool, which possibly filtered out the entries that either were not correctly text-mined or that could not be balanced, while preserving the reactions that generated correct vectors. The comparison between Figure 5.12 and Figure 5.13 points out the fundamental role of the standardisation workflow for the correct analysis of reaction datasets.

To conclude, the “Duplicate Filtering” datasets were also analysed on their intersections. These collections represent clean and balanced versions of the patent data filtered by AP2+AP3 vector duplicates. Diagrams for these datasets are reported in Figure 5.14, which shows the effect of the AP2+AP3 reaction vector filtering on the two collections.

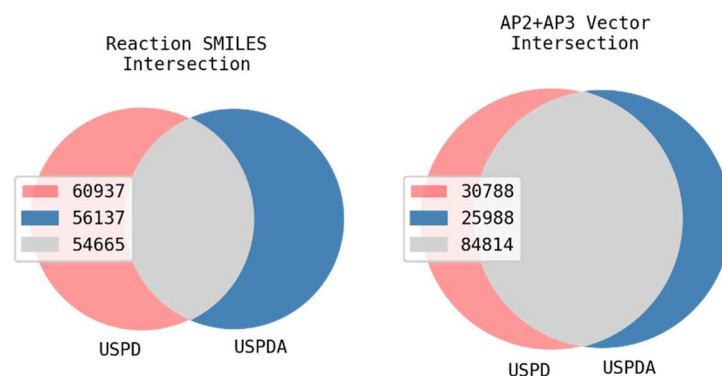


Figure 5.14: “Duplicate Filtering” dataset intersection diagrams. Reaction SMILES and AP2+AP3 vector intersections are reported on left and right charts, respectively.

The reaction SMILES intersection is visibly reduced due to the duplicate filtering process on the AP2+AP3 vectors. The vector intersection is also reduced compared to Figure 5.12 and Figure 5.13. More specifically, USPDA and USPDA reported 74% and 77% of filtered AP2+AP3 vector overlapping: the USPDA set shows a lower intersection compared to the USPDA set, in contrast with the vector intersections analysed previously. The comparison between Figure 5.13 and Figure 5.14 highlights the importance of the duplicate filtering in order to understand correctly the actual composition of reaction datasets.

5.4.4. Database Validation and Encoding

Stoichiometrically balanced reactions generally yield correct reaction vectors, yet they can still produce wrong results when applied in reaction vector-based *de novo* design. This is due to the simplicity of the reaction vector approach, which evaluates only limited extensions of reaction centres described by sets of atom pairs, which may be ambiguous when mapped onto molecular structures. For this reason, the structure generation algorithm can sometimes output incorrect products, for example, when it deals with symmetric reaction centres, or for particular cases that can generate incorrect recombination paths during the reaction vector encoding phase (see Section 3.3.2) (Wallace, 2016). Therefore, although the reaction vector database writer has already an internal algorithm for the validation of reactions, a more accurate reaction vector validation is presented and compared with the database writer algorithm. The aim of this validation is to enhance the quality of reaction vector databases as well as providing new examples of rejected reactions to allow enhancements to be made to the structure generation algorithm.

The validation workflow consists of a simulated structure generation where each reaction vector is tested for its integrity and compatibility with the structure generation algorithm. More specifically, each vector is applied to the reactant(s) from their original reaction example to check whether the product(s) generated by the structure generation algorithm coincides with the original product(s). During this process, every structure is converted according to a strict canonical protocol, where the original product's chirality is removed to ensure comparability with the data produced by the structure generator which ignores stereochemistry, and charges and tautomers are also taken into account in order to avoid false negatives (i.e., reactions that are rejected by mistake). More specifically, charges were neutralised using MOE (Chemical Computing Group ULC and ULC, 2019) and tautomers were canonicalised using MolVS 0.0.9 (Swain, 2017).

Examples of validated and rejected reactions are reported in Figure 5.15, which describes two simple transformations where starting materials are converted into products by applying their corresponding reaction vectors. The top reaction is simulated

correctly, however, the bottom reaction produces a wrong ring closure. The bottom reaction is a typical example where the symmetry of the reaction centre leads the algorithm towards producing an incorrect structure, although an extended level of atom pairs is evaluated. This result does not suggest that the second reaction vector has been produced incorrectly, rather that it is not informative enough for the current implementation of the structure generator to ensure the regeneration of the original products. Therefore, the non-validated (i.e., “Duplicate Filtering”) datasets can be still considered useful for other applications such as data analysis or learning.

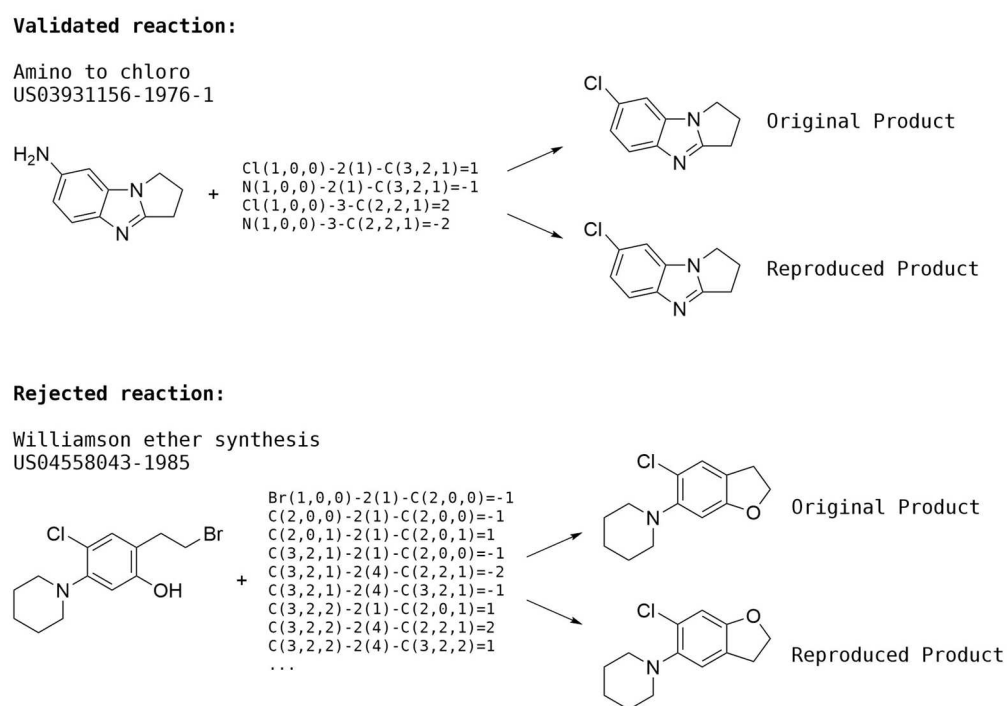


Figure 5.15: Examples of validated (top) and rejected (bottom) reactions from the USPD dataset. The top reaction yields a structure that is identical to the original product, while the bottom reaction produces a ring closure that yields a different structure compared to that described in the original reaction.

The “Duplicate Filtering” USPD and USPDA datasets described in Table 5.1 and Table 5.2, respectively, were validated using the workflow described above. The two datasets were also encoded into databases to determine the number of entries rejected by the reaction vector database writer to make a comparison with the new validation workflow. Results are reported in Table 5.4, which reports similar results for the two

patent sets: 20% and 22% of USPD and USPDA reactions were rejected by the validation algorithm, respectively. The manual inspection of the rejected entries confirmed that most of the examples described symmetric reaction centres, as well as particular cases of heterocyclic conversions, cleavages, and deprotections. This is possibly because the structure generation algorithm was designed with a bias towards structure growing rather than interconversion or cleavage.

Dataset	Validation		Database Writer	
	Successful	Rejected	Successful	Rejected
USPD	92,530	23,072	102,838	12,764
USPDA	86,280	24,522	97,155	13,647

Table 5.4: USPD and USPDA reaction validation and database encoding results.

Table 5.4 also reports the percentages of rejections by the reaction database writer. Only 11% and 12% of the entries from USPD and USPDA datasets failed the encoding process, respectively. Rejected reactions from both methods were also compared with each other to determine their intersections: database failures were fully contained within validation failures, indicating that the validation workflow was capable to detect all the entries that would have failed the database encoding as well as capturing new examples for further optimisation of the database writer. Besides, the validation workflow applies the database writer to obtain single-entry databases, hence resulting in a nested validation procedure. The large difference in percentages between validation and database encoding failures suggests the need of special attention when dealing with vector databases that have not been validated using the simulated structure generation workflow reported in this subsection.

5.4.5. Fingerprint Encoding

The “Balancing Tool” USPD dataset described in Table 5.1 was encoded into dynamic fingerprints, and then used to generate four unique fingerprint datasets: AP2, AP3, AP2+AP3, and AP4. The “Balancing Tool” USPDA dataset described in Table 5.2 was encoded only into unique AP2+AP3 fingerprints. The fingerprint datasets are described in Table 5.5. The different numbers of unique reaction vectors atom pairs, and classes reported in Table 5.5 provide preliminary evidence on the discriminative power

of each atom-pair type. The AP2 dataset describes a drastically smaller number of unique reactions and atom pairs compared to the other datasets, hence suggesting a very low discriminative power at the AP2 level. This is because AP2 fingerprints encode information on the reaction centre only, whereas other fingerprints also encode structural environments that are not directly involved in the transformations. The AP3 and AP4 datasets each result in a higher reduction in reactions, atom pairs, and classes, still suggesting worse discrimination compared to AP2+AP3.

Fingerprint Dataset	Reactions Vectors	Atom pairs	Classes
USPD (AP2)	41,726	1,592	715
USPD (AP3)	113,975	2,613	726
USPD (AP2+AP3)	115,602	4,205	727
USPD (AP4)	112,119	2,898	726
USPDA (AP2+AP3)	110,802	4,046	718

Table 5.5: USPD and USPDA fingerprint dataset descriptions.

The USPD AP2+AP3 was further analysed by converting it into a binary map by replacing all the non-zero values of the dataset with ones. The map is not reported due to its huge content in cells (~500 million). An inspection of the map revealed the presence of recurrent patterns across reaction classes. These patterns were mostly concentrated around the carbon, nitrogen, and oxygen atom-pair columns. These atom pairs are logically the most used to describe organic reactions. In addition, a high content of zeros was identified. A quantitative analysis of the map reported that the percentage of zeros was around 99.55%. Statistically, this means that the USPD AP2+AP3 set contains 1 non-zero value for every 222 zero values. Although dynamic fingerprints are prone to reduce the fingerprint length into the minimum number of atom pairs necessary to describe a given dataset, according to this result the dataset is very sparse.

Atom-pair frequencies were also visually inspected in the USPD AP2+AP3 set. Results are reported in Figure 5.16, which shows that only a very small number of atom pairs are very frequent, hence the dataset is highly skewed in terms of atom pairs as well. An analysis of the most frequent atom pairs revealed the highest abundances were expressed in terms of AP3 atom pairs.

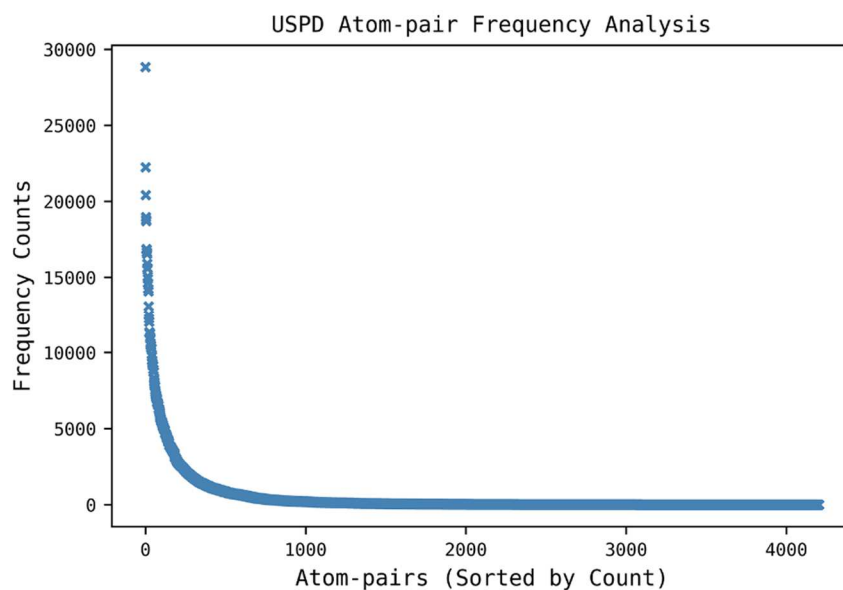


Figure 5.16: USPD AP2+AP3 atom-pair frequency scatter plot (sorted by descending frequency count).

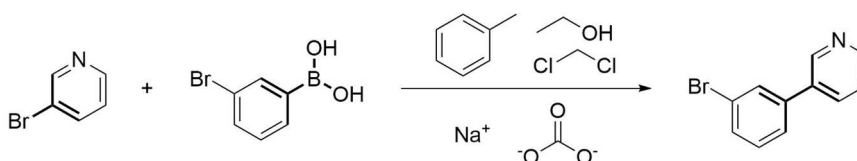
5.4.6. Label Optimisation

The presence of different classes associated with identical or very similar entries can cause problems when datasets are applied for machine learning purposes. For example, if in a classification task, two classes are indistinguishable, their corresponding test entries are likely to be randomly assigned to one of the two classes, thus causing a drop in performance. Several examples of indistinguishable classes were detected from the standardisation workflow following the filtering of some low populated classes described in Section 5.4.2. In particular, the unmapped compound removal tool generated ambiguities among those classes that are distinguishable only on their reagents or solvents. Another example of those classes is reported in Figure 5.17 for “Bromo Suzuki coupling” and “Bromo Suzuki-type coupling”, which shows that the two reactions are originally distinguishable from each other since the actual Suzuki coupling involves the use of Palladium as a catalyst, while the Suzuki-type coupling is described as a transition-metal free reaction. Nevertheless, these two classes become identical after the unmapped compound removal, generating ambiguities for learning algorithms. A potential solution to this issue is to merge such classes together and create a new set of labels. The main drawback of this procedure is that not all these classes can be detected

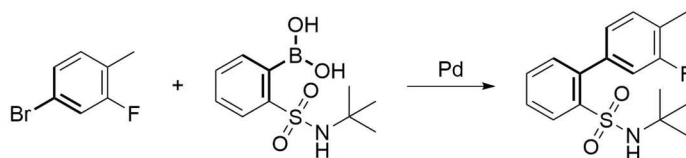
by computational analysis, hence they require to be manually inspected. In addition, the new labelling system has to be defined manually as well, which may result in a bias towards the content of the reference datasets used to create it, specifically the patent collections. However, the creation of a new set of labels would also offer the advantage of having a tailored nomenclature for *de novo* design.

Original data:

Bromo Suzuki-type Coupling
US05128335-1992-1

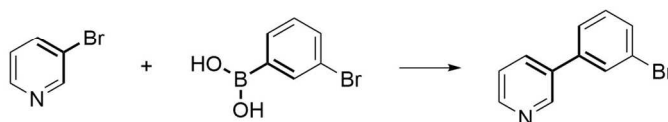


Bromo Suzuki Coupling
US05281614-1994-26



Cleaned data:

Bromo Suzuki-type Coupling
US05128335-1992-1



Bromo Suzuki Coupling
US05281614-1994-26

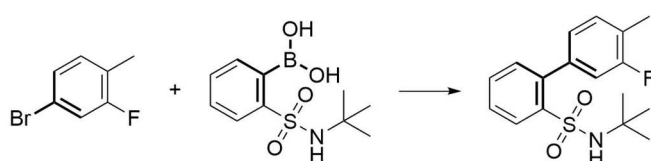


Figure 5.17: Example of two classes that are not distinguishable after the reaction standardisation workflow.

As reported in Section 5.4.2, NameRxn adopts a three-level classification system that is based on official nomenclatures developed in the past, which are accurate but not optimised for browsing purposes. This is because although NameRxn already adopts a hierarchical system, subclasses and reaction types are often described by the names of

the scientists who discovered such reactions, which are not promptly understandable by non-experts in organic chemistry. In addition, NameRxn labels are not optimally structured for alphabetical sorting since they often contain redundant information, thus searching for multiple reaction classes can be confusing sometimes. For example, Heck, Negishi, Sonogashira, Suzuki, and Stille couplings are all cross-coupling reactions that form a carbon-carbon bond between complex fragments characterised by typical functional groups, hence NameRxn includes all of them in the major class “C-C bond formation”. However, their subclasses and reaction types contain very similar information, also not organised in a convenient manner for browsing. These labels are reported in Table 5.6, which shows that subclasses and reaction types share redundant descriptions, as well as being very detailed compared to their major classes. An additional level between major classes and subclasses would be first required to group these reactions as couplings; consequently, subclasses and reaction types should be reorganised to describe specifically transformations and involved functionalities.

Major Class	Subclass	Reaction-type
C-C bond formation	Heck reaction	Bromo Heck reaction
C-C bond formation	Other Pd-catalyzed reactions	Negishi coupling
C-C bond formation	Stille reaction	Chloro Stille reaction
C-C bond formation	Sonogashira reaction	Iodo Sonogashira coupling
C-C bond formation	Suzuki coupling	Iodo Suzuki coupling

Table 5.6: NameRxn labelling of some cross-coupling reactions.

By accounting for this requirement and in the effort of merging together those classes that were not distinguishable after agent removal, a new hierarchical labelling system was developed manually following the inspection of each individual class in the USPD and USPDA datasets. For each class, multiple examples of reactions were evaluated to identify the general cores of transformations, and thus produce a new set of suitable labels. Labels describing transformations that cannot be processed using reaction vectors (e.g. stereochemistry inversions, resolutions, etc.) were not considered. The procedure condensed a total number of 695 NameRxn reaction types into 597 new labels; however, none of the datasets obtained from Sections 5.4.2 and 5.4.5 was processed using the new labelling system at this stage. A table containing NameRxn

labels (only subclasses and reaction types) and their corresponding replacements is reported in Appendix A.

In the new labelling system, the level of detail is distributed across 4 levels, ranging from general categories to increasingly more specific sub-classes and reactant descriptions. More specifically, the first level describes the general transformation according to some essential definitions (e.g. C-C Bond Formation, Functional Conversion, Protection, etc.). The second level describes the typology of the transformation (e.g. Coupling, Alcohol to alkene, etc.). The third and fourth levels contain additional information on substrates/products (e.g. Isocyanate + amine), reaction inventors (e.g. Suzuki) or involved functionalities (e.g. Bromo). This system will be conventionally indicated as SHREC (Sheffield Hierarchical REaction Classification). An example of label replacement is reported in Table 5.7, which shows that the NameRxn reaction types can be used as lookups for algorithmic label replacement with single hierarchical labels. The SHREC mainly focuses on reporting the same information ordered by increasing level of detail, thus enhancing the retrieval of groups of similar transformations by sorting them alphabetically. Labels can also be decomposed by removing the content within brackets to increase their generalisation. Nevertheless, the SHREC is not supposed to be accurate or exhaustive in terms of class coverage since its development was purely based on the patent datasets and NameRxn.

NameRxn Reaction-type	SHREC Replacement
Bromo Heck reaction	C-C Bond Formation (Coupling) (Heck) (Bromo)
Negishi coupling	C-C Bond Formation (Coupling) (Negishi)
Chloro Stille reaction	C-C Bond Formation (Coupling) (Stille) (Chloro)
Iodo Sonogashira coupling	C-C Bond Formation (Coupling) (Sonogashira) (Iodo)
Iodo Suzuki coupling	C-C Bond Formation (Coupling) (Suzuki) (Iodo)

Table 5.7: NameRxn to SHREC label replacement for some cross-coupling reactions.

5.5. External Data

5.5.1. Introduction

Additional data sources are crucial for the validation of computational methods. For example, external datasets can help with reducing the bias generated by particular

data compositions, they can be used to repeat a given experiment in the attempt to verify its results and conclusions, or to explore the potentials of new tools. As sources of unclassified external data, two collections of reactions from the years 2008 and 2018, were obtained from the Journal of Medicinal Chemistry (JMC), a well-established peer-reviewed journal in drug discovery, and one additional reaction set was mined from the Evotec Electronic Laboratory Notebook (Evotec ELN). The JMC 2008 collection describes a set of reactions published in the year 2008 which was originally created for testing the reaction vector-based design tool (Patel, 2009). The JMC 2018 collection contains a set of single step reactions published between January the 1st and September the 10th, 2018, obtained from Reaxys (Elsevier, 2009).

ELNs are scientific programs which offer remarkable advantages over traditional paper laboratory notebooks, such as fast information retrieval, backup, and sharing. Furthermore, they are widely used in industry to protect intellectual property (Baykoucheva, 2015). The Evotec ELN set describes some in-house reactions carried out between September the 9th, 2009 and February the 27th, 2018, recorded into the Evotec corporate ELN in the United Kingdom.

Although none of these datasets contains reaction class information, so that they cannot be used, for example, for quantitative assessment for reaction classification purposes, they can still be used to reduce the bias in *de novo* design simulations or to enable the comparison between different data sources, specifically patents, journals, and industrial data. In this section, the standardisation and encoding of the selected external datasets is described.

5.5.2. Standardisation

The external datasets were first processed through the reaction standardisation workflow described in Section 5.2.2. Entries containing more than six reactants/products were not pre-filtered at this stage. Duplicates were filtered by AP2+AP3 vector centres to yield unique datasets for reaction vector database encoding. Results for the JMC 2008

and 2018 collections, and the Evotec ELN dataset are reported in Table 5.8, Table 5.9, and Table 5.10, respectively:

JMC 2008	Reactions
Original Data	19,914
Balancing Tool	19,209
Duplicate Filtering	12,242

Table 5.8: JMC 2008 dataset description through the standardisation workflow.

JMC 2018	Reactions
Original Data	26,459
Balancing Tool	24,606
Duplicate Filtering	7,635

Table 5.9: JMC 2018 dataset description through the standardisation workflow.

Evotec ELN	Reactions
Original Data	168,375
Balancing Tool	144,014
Duplicate Filtering	29,105

Table 5.10: Evotec ELN dataset description through the standardisation workflow.

Table 5.8, Table 5.9, and Table 5.10 describe different trends for the processed datasets: the balancing tool reduced the JMC 2008 and 2018, and the ELN sets by 4%, 7%, and 14% in size, respectively. This result possibly indicates that the 2008 collection was encoded paying particular attention to the reaction stoichiometry, while the ELN set originally contained a much higher presence of mistakes and/or imbalanced entries. Consequently, the duplicate filtering further reduced the JMC 2008 and 2018, and the ELN sets by 36%, 69%, and 80%, respectively, suggesting a higher variety of AP2+AP3 centres in the JMC 2008 collection. In addition, the comparison with the duplicate filtering reduction on the USPD and USFDA (Section 5.4.2), which corresponded to 90% and 91%, respectively, suggests that journal data is likely to describe more diverse reaction centres compared to industrial and patent data.

5.5.3. Database and Fingerprint Encoding

The “Duplicate Filtering” datasets were encoded into reaction vector databases using the reaction vector database writer. The current versions of these datasets also contain classification data obtained from Chapter 7. Numbers of encoded and rejected

reactions for each database are reported in Table 5.11, which describes a trend in line with the results from Section 5.4.5: JMC 2008 and 2018, and ELN datasets reported 6%, 7%, and 13% of rejected reactions, respectively.

Dataset	Database Writer	
	Successful	Rejected
JMC 2008	11,545	697
JMC 2018	7,109	526
Evotec ELN	25,463	3,642

Table 5.11: External dataset database encoding results.

Fingerprint datasets were also generated from the “Balancing Tool” datasets, in order to preserve the original composition of their corresponding collections. Fingerprint datasets are described in Table 5.12, which provides additional evidence on the nature of the datasets. Although the JMC 2008 collection describes 22% and 87% less total entries compared to the JMC 2018 and the Evotec ELN sets, respectively, JMC 2008 is described by 2.2 fold the number of atom pairs compared to JMC 2018 and 1.6 fold the number of atom pairs compared to the ELN. These results confirm the presence of a high relative variety of AP2+AP3 centres in the JMC 2008 collection. The analysis of the ratios between atom pairs and number of entries confirms these trends: the JMC 2008 and 2018 collections, and the Evotec ELN set reports the ratios 0.43, 0.31, and 0.11, respectively. The qualitative comparison with the USPD and USPDA fingerprint datasets (see Section 5.4.5) further substantiates these conclusions.

Fingerprint Dataset	Reactions	Atom pairs
JMC 2008	19,209	5,331
JMC 2018	24,606	2,382
Evotec ELN	144,014	3,305

Table 5.12: External fingerprint dataset descriptions.

5.6. Conclusions

In this chapter, the issues related to the use of reaction data for analysis, learning, and drug design purposes have first been discussed. These issues are mainly related to the lack of protocols for reaction standardisation and validation. Consequently, a series of algorithms have been presented and applied to a selection of reaction datasets derived

from literature and industrial data. As a result of the pre-processing pipeline, a number of standardised datasets, also encoded into fingerprint format for their use in machine learning, and reaction vector databases, were produced. The new datasets generated in this study are meant to provide standardised data for the next experiments reported in this thesis. The next chapter presents a new reaction vector-based *de novo* design framework and its application for the experimental validation of reaction vectors.

Chapter 6: Pseudoretrosynthetic de novo Design

6.1. Introduction

Reaction vectors were first validated through the reproduction of known reactions of different types, then by attempting the generation of novel structures that were assessed on their relevancy in lead optimisation and their diversity against the products originally described in the reference reactions (Patel *et al.*, 2009) (Hristozov *et al.*, 2011). More recently, the method was tested using evolutionary algorithms in multi-objective *de novo* design (Gillet, Bodkin and Hristozov, 2013). However, the experimental validation of reaction vectors has never been reported in the literature; hence, no practical demonstration of their use in medicinal chemistry has been given.

In this chapter, the development and implementation of a workflow for ligand-based *de novo* design using reaction vectors is described, where a ligand is first fragmented, each fragment is used as a search query to identify new fragments, which are then combined combinatorially using reaction vectors. This approach is called RENATE and is similar to that of Flux mentioned in Section 2.5.2, with the significant difference that the fragments are combined using knowledge of real reactions. The algorithm is first validated retrospectively on a diverse set of marketed drugs using two reaction vector databases. The most promising setup is then tested in a real design experiment, where a selection of inhibitors for a specific target are used to generate novel candidates with improved properties. The designed structures are scored using predictive models developed using machine learning and docking methods. A subset of compounds is selected for synthesis by synthetic chemists at Evotec. A number of these are successfully synthesised and computationally evaluated on their BBB penetration in comparison with their reference drugs.

The aim of this study is to provide experimental evidence on the effectiveness of the reaction vector approach for *de novo* design purposes, in particular when combined with the US pharmaceutical patent data as a source of reaction vectors.

The chapter is organised as follows. The components and functioning of the reaction-based *de novo* design algorithm based on the concepts of pseudoretrosynthesis are described in Section 6.2. The use of reaction vectors in pseudoretrosynthetic *de novo* design is then validated computationally and experimentally. Section 6.3 describes a retrospective validation on a set of top prescribed drugs in the US to verify that the algorithm can explore effectively the chemical space, and to identify a promising setup for *de novo* drug design. Section 6.4 reports the application of the algorithm to a case study, where small-molecules with improved predicted brain penetration are designed from a set of known inhibitors. Candidates are scored computationally then inspected to yield a selection of compounds for synthesis and further evaluation.

6.2. The RENATE Algorithm

The general concepts of pseudoretrosynthetic *de novo* design are reviewed in Section 2.5.2. In this section, the integration of the structure generation algorithm within a pseudoretrosynthetic framework is presented. This tool is referred to as RENATE (pseudoRetrosynthEtic design using reAcTion vEctors). RENATE is composed of four modules: *ligand fragmentation*, *building block search*, *structure generation*, and *scoring*.

The *ligand fragmentation* module is the first component and is used to break a given query ligand into fragments from which the main scaffold is identified, with the remaining fragments identified as substituents. The fragmentation is performed using the BRICS module in RDKit (`rdkit.Chem.BRICS`) (Degen *et al.*, 2008). BRICS accepts a parameter called *minFragmentSize* that determines which bonds can be broken according to the size of their resulting fragments. *minFragmentSize* can be used to obtain bigger fragments when dealing, for example, with branched molecules or structures that contain linkers, rather than producing many small fragments. An additional parameter that was implemented in the ligand fragmentation module is *MinKeyFragSize* that determines which fragments are filtered out according to the sum of their heavy atom counts and number of connections. For example, the scaffold generated from the decomposition of the drug Celecoxib has a size equal to eight (Figure 6.1).

MinKeyFragSize can be used to filter out fragments that are too small in order to focus on key fragments, hence reducing the number of design iterations on a given query.

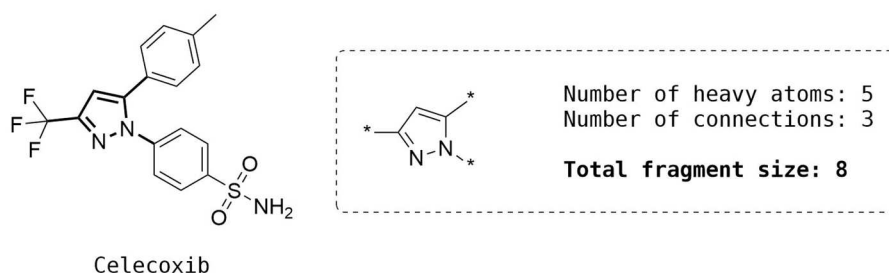


Figure 6.1: Fragment (highlighted in bold on the drug structure) generated by BRICS decomposition of the drug Celecoxib: The fragment has five heavy atoms and three connections, hence its size is equal to eight.

Once key fragments are determined, they are sorted first by descending number of connections and then by the number of heavy atoms. The fragment at the top of the ranked list is identified as the scaffold (referred to as the starting material hereon), while the remaining fragments are considered as substituents or reagents. This heuristic gives priority to highly connected fragments in order to build up candidates from ‘the inside to the outside’ across the design cycles. An example of this procedure is reported for Celecoxib in Figure 6.2, where the pyrazole (diazole heterocycle) is identified as the starting material, even though it has a smaller number of heavy atoms compared to the benzenesulfonamide (i.e., five versus ten heavy atom counts, respectively). The main difference between the ligand fragmentation modules in RENATE and in other pseudoretrosynthetic design programs, such as Flux (Fechner and Schneider, 2006), is the definitions used to break molecules into fragments and the post-processing methods applied on them. For example, Flux uses the 11 bond-cleavage types implemented in RECAP (Lewell *et al.*, 1998) with some exceptions to avoid the generation of building blocks that cannot be used during the design. RENATE uses BRICS, which can be considered as an extension of the RECAP approach since it applied similar principles yet implements up to 16 bond-cleavage types. Building blocks that are not considered useful for the design are then filtered out by RENATE afterwards according to the parameter *MinKeyFragSize*.

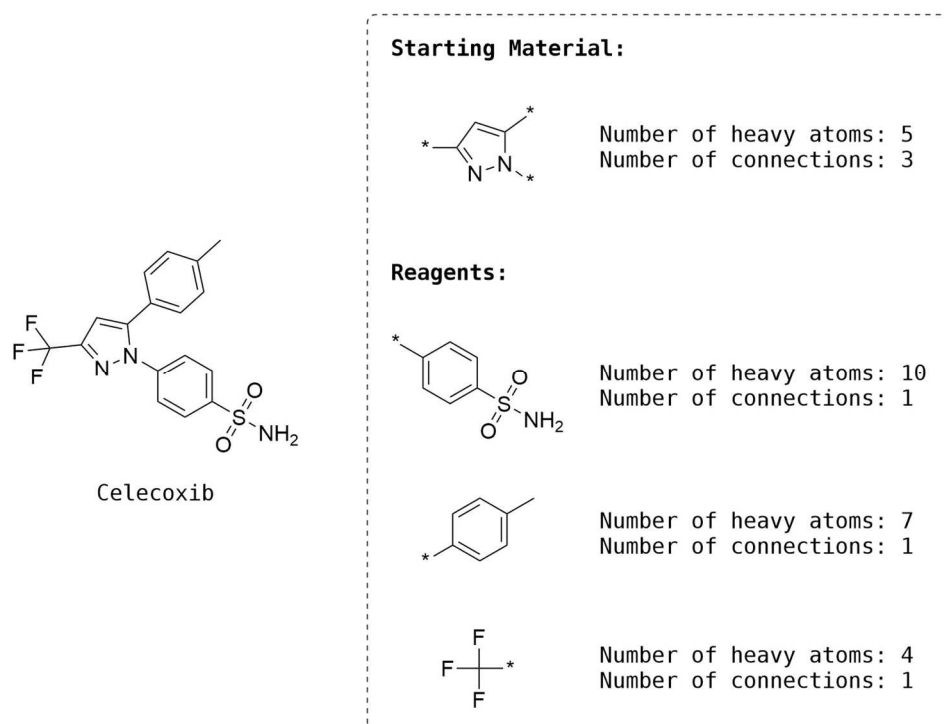


Figure 6.2: Scaffold and reagent identification procedure applied to the fragments derived from Celecoxib.

Next, the *building block search* module takes the key fragments (starting material and substituents) and performs a search on an external source of compounds, for example, a commercial catalogue of reagents, in order to retrieve similar fragments according to the selected scoring method. The current implementation of RENATE uses fingerprint similarity. A set of compounds is returned sorted by their similarity to the corresponding key fragment. For example, given a starting material ‘A’ and a reagent ‘B’, the algorithm returns two sets of compounds (e.g. $\{a_1, a_2, \dots, a_x\}$, $\{b_1, b_2, \dots, b_y\}$) scored by similarity on the selected key fragments. The size of the set is determined by the parameters *MaxStartingMaterials* and *MaxReagents*, which control the maximum numbers of compounds in the starting material and reagent sets, respectively. These two parameters can be configured using similar values (e.g. 750 and 1,000), or, for example, with imbalanced values (e.g. 25 and 30,000) to retrieve a higher number of reagents. The building block search module implemented in RENATE operates in a simpler way compared to algorithms such as those in Flux or COLIBREE (Hartenfeller *et al.*, 2008). These two programs implement strategies for the selection of blocks by accounting for

the bond-cleavage type origin of each key fragment in order to increase the synthetic feasibility of products. RENATE does not implement such strategies since synthetic accessibility is maximised using reaction vectors in the molecule generation phase.

The *structure generation* module was constructed in order to operate in two ways. In the first design cycle, the algorithm combines the compounds retrieved for the starting material with those retrieved for the first reagent. For example, the sets $\{a_1, a_2, \dots, a_x\}$ and $\{b_1, b_2, \dots, b_y\}$ are combined combinatorially to form products. For each pair of fragments (one from the *a* set and one from the *b* set), the set of reaction vectors is searched and for each applicable reaction vector a product is generated. Hence, a set of product molecules is produced (e.g. $\{a_1-b_3, b_2-a_5, a_9-b_{11}\}$), which are then scored by the *scoring* module, and the top scoring products form a new set of starting materials. These are then input to the structure generator to be combined with the next reagent set (e.g. $\{c_1, c_2, \dots, c_z\}$). The size of the starting material set after every design iteration is controlled by the parameter *MaxStartingMaterials*. The algorithm iterates through each key fragment until every reagent set has been used. The structure generation module, therefore, uses a reaction vector database as a source of reactions (see Section 3.3.2).

The structure generation module combines input fragments with no regards to their properties. At the end of each cycle, products are filtered according to two more parameters implemented in the tool. *QueryHeavyAtomsAddThreshold* is a threshold that is used to filter out products that are too big compared to the query. For example, if *QueryHeavyAtomsAddThreshold* is equal to 0.25, all the products exceeding 25% the heavy atom count of the query structure are filtered out. Celecoxib has 26 heavy atoms, hence products with more than 33 heavy atoms will be filtered out. *NumProductsCycle* determines how many products are retained at the end of each cycle for the final scoring. This operation allows a number of intermediates to be retained until the end of all design cycles to enable the scoring of a larger number of products when selecting a final population of candidates.

This can be particularly useful, for example, when products generated during the last step of the design are worse than those generated in the previous cycles. In these

cases, intermediates will be scored higher than the last generation of products, hence the latter will not be selected as final candidates. The structure generation in RENATE is more sophisticated compared to other pseudoretrosynthetic design programs since it occurs by means of the reaction vector structure generator, whereas algorithms such as Flux or COLIBREE simply evaluate the nature of the attachment points on fragments for their recombination.

The *scoring* module is defined by the user. In the simplest implementation of RENATE, it is configured as a similarity-based scoring method which selects the best molecules based upon their similarity to the query. The scoring module first drives the design by selecting the best products at the end of each cycle (*active* scoring), then finally sorts the entire population of intermediates and final products to yield a set of candidates (*passive* scoring). The total number of candidates produced by the algorithm for a given query is controlled by the parameter *NumFinalProducts*. Other examples of ligand-based tools also use similarity techniques for product scoring, yet different and/or multiple scoring modules can be optionally implemented in RENATE if desired.

6.2.1. KNIME Implementation

An illustration of the workflow is shown in Figure 6.3, which describes the combination of the modules of RENATE and some technical implications necessary for their functioning.

Query molecules are first fragmented and used to find sets of scaffolds (starting materials) and substituents (reagents). Consequently, scaffolds are written in a temporary table, which in turn is read as a starting population set by the *structure generation* module. Once the scaffolds are combined with the first set of substituents, the new population is scored (*active scoring*) and written in the temporary table. The algorithm iterates through each key fragment set while reading and overwriting the temporary table until the process is over. The final population is then rescored (*passive scoring*) and written out.

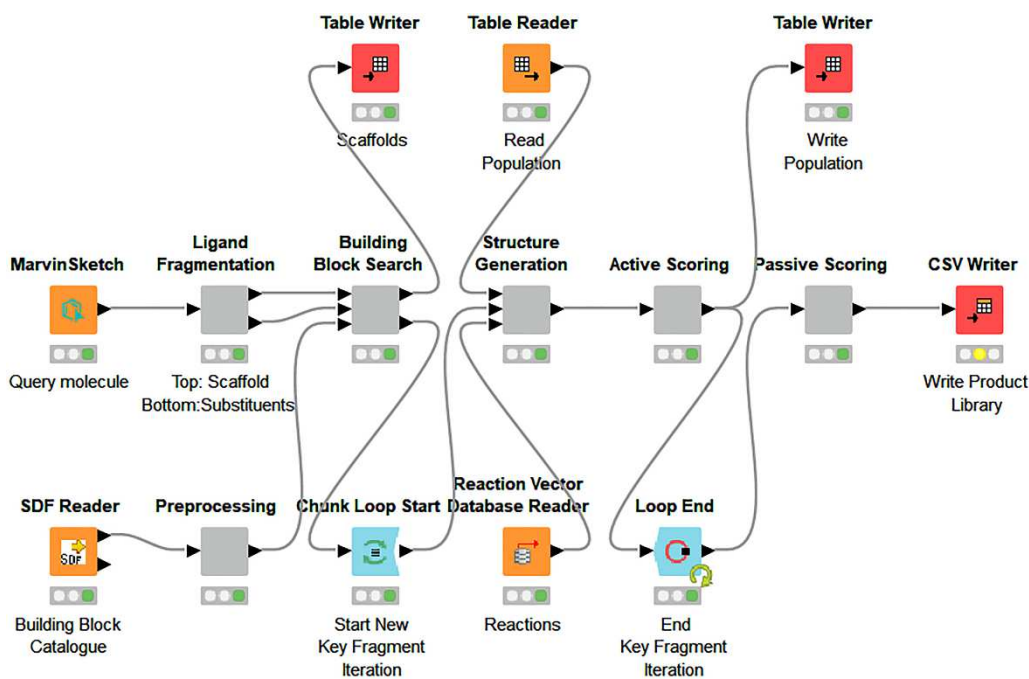


Figure 6.3: RENATE KNIME workflow.

Note that RENATE relies on two assumptions in order to produce the input data for the structure generator. First, key fragments are identified by virtual fragmentation of the reference ligands, which can yield precursors that differ significantly from the building blocks used in the actual synthetic routes. Second, starting materials and reagents are defined using heuristics that were determined upon the features and limitations of the reaction vector-based design framework. Hence, the validity of these assumptions needs to be assessed.

6.3. Top 200 Drugs 2017 Validation

6.3.1. Introduction

A collection of drugs from the top 200 medicines prescribed in the US in the year 2017 is used as a reference set for validating RENATE. Following the application of some filtering rules, the remaining drugs are fragmented and their fragments are used as queries to assess whether the algorithm can recreate the drugs, or at very least, generate similar structures. This experiment provides a retrospective validation, which aims to verify the assumptions made by the algorithm (i.e., correct ligand fragmentation and starting material/reagent role assignment and ranking) and that the selected building

blocks and reaction vector databases enable an effective search in the drug-like space. Once the core of RENATE is validated, it can be combined with more complex scoring functions for actual *de novo* design applications, i.e., to design new compounds that have improved properties compared to their reference ligands.

6.3.2. Data Selection

The list of 200 top prescribed drugs in the US in the year 2017 was obtained from the ClinCalc database (<https://clincalc.com/DrugStats/>). ClinCalc generates statistics on drugs according to data released annually by the US government (<https://www.meps.ahrq.gov/>). The drugs were drawn using MarvinSketch and converted into SMILES structures which were sanitised using RDKit, then salts and ions were stripped to obtain only one molecule per entry. Molecules were processed to produce the following descriptors: 'NumAtoms', 'NumFusedRings', 'NumRings', and 'NumLipinskiViolations'. 'NumLipinskiViolations' was calculated on the rule-of-five proposed by Lipinski and colleagues (1997). A series of filters was applied to obtain a benchmark set of drug-like molecules suitable for testing in a fragment-based design framework: minimum 20 total atoms, maximum 3 fused rings, minimum 2 rings, maximum 1 Lipinski's violation. 92 molecules were retained.

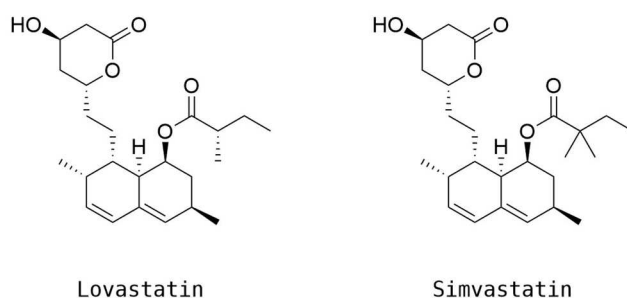


Figure 6.4: Lovastatin and Simvastatin: Two cholesterol-lowering medications from the family of statins which differ by only a methyl group.

Next, all pairwise similarities were calculated using binary Morgan fingerprints (Radius 2) 1024-bit and the Tanimoto coefficient and only one molecule was retained for every pair with similarity greater than or equal to 0.6 in order to maximise the diversity in the dataset. For example, Lovastatin and Simvastatin (Figure 6.4) were

both present in the original set, with 0.742 Tanimoto similarity; hence, only one of the two structures was retained. The similarity filtering yielded 73 structures which are reported in Appendix B.

The selected structures were described using the RDKit Descriptor Calculation node, to produce a number of descriptors according to the drug-like properties proposed by Lipinski and colleagues (1997): 'ExactMW', SlogP, 'NumLipinskiHBD', and 'NumLipinskiHBA'. Distributions are reported in Figure 6.5, which shows that all the compounds roughly fall within the drug-like domain, except for Levothyroxine ('ExactMW' = 777 Da) and Ergocalciferol ('SlogP' = 7.6).

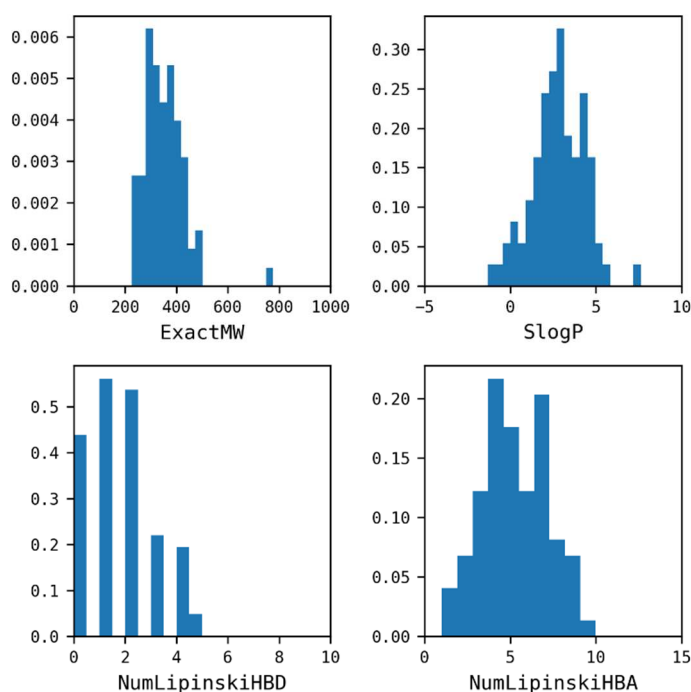


Figure 6.5: Lipinski's RO5 property distributions covered by the 73 drugs selected for the validation of RENATE.

6.3.3. Building Block and Reaction Vector Selection

A collection of 862,458 building blocks was obtained from Enamine in August 2018 (<https://enamine.net/building-blocks/>). Structures were sanitised using RDKit, neutralised, then duplicates were filtered by InChIKeys. 746,272 structures were retained and selected as a source of starting materials and reagents.

Count Morgan fingerprints (Radius 2) 1024-bit and Euclidean distance were selected as molecular descriptors and distance metric, respectively, for the scoring of building blocks (*building block search* module) and structure generation products (*structure generation* and *scoring* modules). The selection of structural fingerprints was aimed at maximizing the chance of reproducing the original queries for the purpose of the validation, rather than generating novel compounds.

The 92,530 USPD reaction vectors and 7,109 JMC 2018 reaction vectors, described in Sections 5.4.4 and 5.5.3, respectively, were selected as sources of reaction vectors.

6.3.4. Method

The 746,272 structure Enamine reagent set was first filtered by InChIKeys using the 73 drug structures selected in Section 6.3.2 as queries. This operation prevents the algorithm from selecting the complete drug structure as a starting material or reagent during the design, hence compromising the validation. Each drug molecule was then processed by RENATE as described in Section 6.2. Two design procedures were carried out using the USPD and JMC 2018 databases, respectively. The parameters used in the experiments are reported in Table 6.1:

Parameters
minFragmentSize=1, MinKeyFragSize=5, MaxStartingMaterials=750, MaxReagents=1000, QueryHeavyAtomsAddThreshold=0.25, NumProductsCycle=4000, NumFinalProducts=1000

Table 6.1: Top 200 Drugs 2017 design RENATE parameters.

6.3.5. Results and Discussion

Results from the USPD and JMC 2018 pipelines were collected. In both cases, 11 drugs (15%) failed the BRICS decomposition (starred in Appendix B), while 62 queries (85%) were successfully processed. The decomposition mainly failed due to the lack of rules for the fragmentation of sigma bonds between aromatic and aliphatic rings. These failures indicate that BRICS lacks some important fragmentation rules. Some examples of failed queries are reported in Figure 6.6.

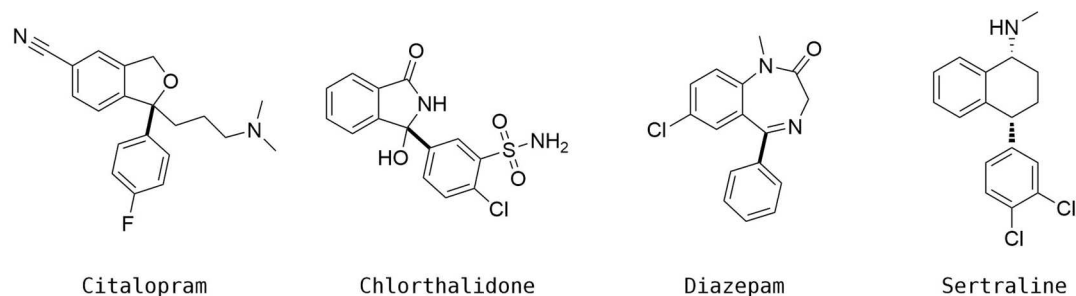


Figure 6.6: Queries that failed the BRICS decomposition. Potential fragmentation bonds are highlighted in bold.

Candidates generated from the successful queries (i.e., 1000 candidates per query) were processed as follows. The designed structures were rescored against their reference drugs using four binary fingerprints: RDKit Morgan (Radius 2) 1024-bit (equivalent of ECFP4) and CDK ECFP4 (structural fingerprints), RDKit FeatMorgan (Radius 2) 1024-bit (equivalent of FCFP4) and CDK FCFP4 (pharmacophoric fingerprints) (Landrum, 2016) (Clark, Sarker and Ekins, 2014). Query ligands' chiral centres were flattened and charges were neutralised to enable the correct comparison between reference drugs and product structures generated by the algorithm. One top scoring compound (closest reproduction) per query ligand was retained, then the pair-wise similarities (best compound-query) from each design pipeline were statistically analysed. Statistics are reported in Table 6.2:

Design	Binary Fingerprint	Min	Max	Mean	Median
USPD	RDKit-ECFP4	0.19	1.00	0.62	0.60
	CDK-ECFP4	0.18	1.00	0.62	0.61
	RDKit-FCFP4	0.29	1.00	0.64	0.64
	CDK-FCFP4	0.23	1.00	0.65	0.64
JMC 2018	RDKit-ECFP4	0.15	1.00	0.51	0.48
	CDK-ECFP4	0.12	1.00	0.50	0.48
	RDKit-FCFP4	0.16	1.00	0.51	0.45
	CDK-FCFP4	0.21	1.00	0.52	0.52

Table 6.2: Statistics from the pair-wise similarities between queries and their corresponding best compounds from USPD and JMC 2018 designs.

The USPD and JMC mean and median values in Table 6.2 show that the USPD pipeline produced structures 24% (mean) and 29% (median) more similar to their corresponding queries compared to the compounds from the JMC 2018 pipeline,

respectively. Minimum values are also generally better for the USPD compounds, in particular for the pharmacophoric fingerprints suggesting a higher presence of compounds that are functionally similar to the queries, although the algorithm did not attempt to optimise this property. These results are not surprising since the USPD database contains 13 times the number of reaction vectors in the JMC 2018 database; hence, the first is expected to enable a more extensive exploration of the chemical space.

The USPD and JMC pipelines reproduced 6 and 1 queries, respectively, which is a reasonable achievement considering the assumptions made by RENATE (i.e., pseudoretrosynthesis and starting material/reagent role assignment and ranking), and the uncertain presence of the correct reagents and reaction vectors necessary for the reproduction of the original queries. More specifically, 3, 2, and 1 queries from the USPD design were regenerated via 1-step, 2-step, and 3-step routes, respectively, while the JMC design regenerated only 1 query using a 2-step route. These results support the selection of the USPD database compared to JMC as a source of reaction vectors for *de novo* design applications.

Each top scoring compound per query from the USPD design was manually inspected to estimate qualitatively the performance of the algorithm. Some examples of top scoring candidates and their queries, sorted by increasing similarity, are reported in Figure 6.7, which shows that candidates in the range of similarities between 0.3-0.4 present low similarities to the original queries. These structures cannot be accepted as analogues of the queries since they show different shapes and/or shuffled functionalities. Conversely, candidates become very similar to their query molecules for similarities greater than 0.5. The top scoring candidate for Tizanidine (0.53 similarity) presents minor variations on the five-membered ring, and Cephalexin's candidate (0.78 similarity) differs only in the substitution of an amino group with a methyl. The USPD and JMC experiments reported 70% and 47% best scoring candidates with similarity greater than 0.5, respectively, substantiating the selection of the USPD database. These results suggest that the algorithm can explore (i.e., direct the search towards the right region of chemical space) and also exploit (i.e., reproduce the reference drugs or at least

generate very similar candidates) effectively the chemical space when sufficient amounts of building blocks and reactions plus an effective scoring function are provided.

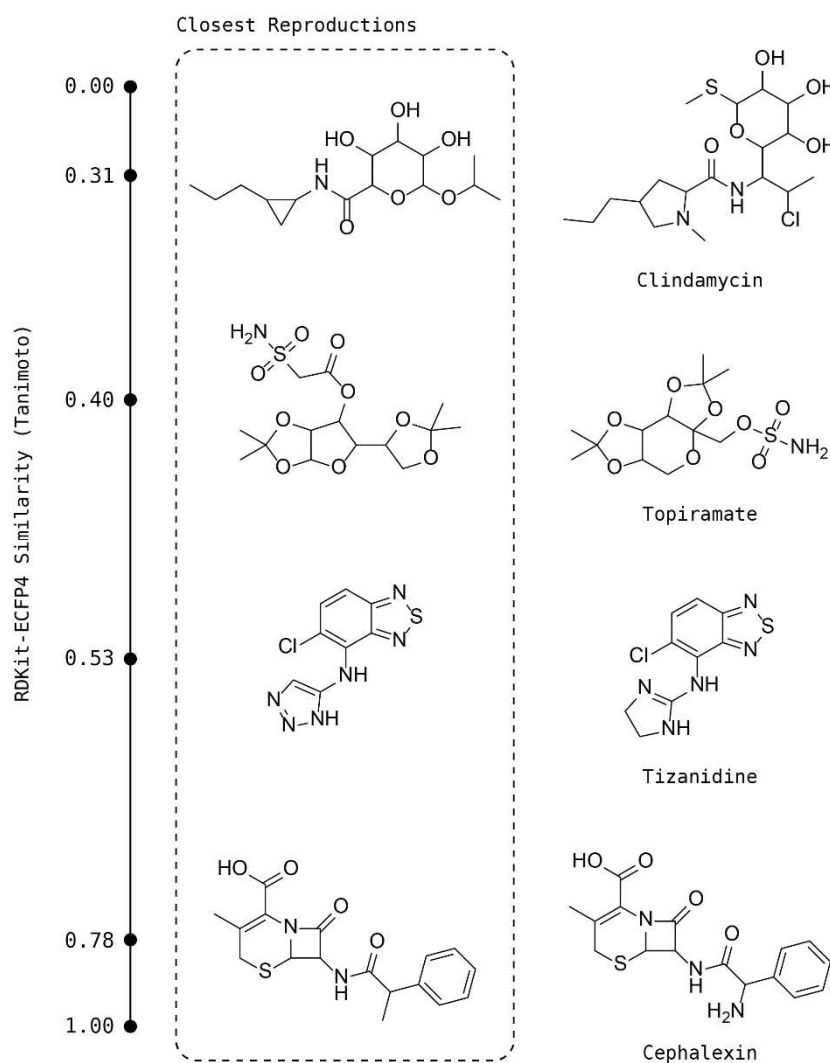


Figure 6.7: Examples of some best candidate-drug pairs according to RDKit-ECFP4 similarity generated from the USPD design pipeline.

The reproduced queries were also inspected on their virtual synthetic routes, which were compared with the actual routes used to produce the drugs. The original patents were used as references. Table 6.3 summarises the number of virtual and real synthetic steps per reproduced query. The average ratio between actual and virtual numbers of synthetic steps is 3.4. The manual inspection of virtual and real synthetic schemes revealed that none of the drugs was reproduced using their original references, rather, most of the virtual routes were completely different from those presented in the patents.

In particular, Glipizide and Glyburide produced schemes similar to the real routes, while Brimonidine, Diclofenac, Naproxen, and Rivaroxaban reported very different syntheses.

Design	Drug	Virtual Steps	Actual Steps (Patent Reference)
USPD	Brimonidine	1	3 (US3890319A)
	Glipizide	2	2 (DE2012138)
	Glyburide	2	3 (DE1283837)
	Levofloxacin	1	7 (US4382892A)
	Naproxen	1	8 (US3896157)
	Rivaroxaban	3	4 (US7157456B2)
JMC 2018	Diclofenac	2	4 (DE1793592)

Table 6.3: Comparison between virtual and real synthetic steps for the drug structures reproduced during the design.

There are a number of reasons that can explain these results. First, some of the reference patents were not issued in the US, hence they cannot be reproduced using the USPD database. Second, patents often describe combinations of small and cheap building blocks, whereas RENATE makes use of a catalogue that also contains complex and expensive reagents which can be very close to the queries (e.g. the original preparation of Naproxen describes how to synthesize naphthalene derivatives and their further functionalisation (8 steps), whereas RENATE promptly identified an analogue of Naproxen as a starting material then converted it (1 step) into the actual drug using a simple reagent and an appropriate transformation. Levofloxacin also reported similar results. Third, the use of pseudoretrosynthesis often does not permit the decomposition of ligands into their actual precursors. For example, transformations such as ring closures or functional eliminations (e.g. the original Rivaroxaban preparation) cannot be backtracked using this method because they leave no trace of their occurrence on structures. Fourth, real syntheses often involve chiral purifications and account for the presence of competing functionalities, hence they involve stereochemistry and protection chemistry.

Reaction vectors do not encode chirality and they deal with only limited extensions of reaction centres; hence, they often produce simplified schemes that do not reflect the chemistry required to carry out the syntheses, although they can still provide useful

references that only require adjustments. These limitations inspired the development of the reaction class recommendation models presented in Chapter 8.

To conclude, results from this experiment have provided evidence for the validation of the core of RENATE. The algorithm performs valid fragmentations, retrieves useful reagents by means of the key fragment information, and combines reagents correctly, yielding drug-like molecules that are closely similar or even identical to their original queries.

6.4. PARP1 Inhibitor Design

6.4.1. Introduction

Following its computational validation, RENATE is applied prospectively in a real *de novo* design scenario, where it is integrated within a multi-objective *de novo* design workflow to yield new synthetically accessible inhibitors with improved brain penetration for a known biological target. A series of new components are implemented in the scoring module to achieve this goal. Finally, a number of selected candidates are synthesised and evaluated computationally on their predicted pharmacokinetic properties.

6.4.2. Target and Ligand Selection

Poly(ADP-ribose) polymerase-1 (PARP1) is a nuclear enzyme activated by DNA damage (i.e., DNA strand breaks) and is involved in DNA repair. PARP1 promotes the recruitment of DNA repair factors by catalysing the covalent addition of ADP-ribose moieties on these enzymes using NAD⁺ as the donor of ADP-ribose (Satoh and Lindahl, 1992). These processes are reported in Figure 6.8.

Over-activation of PARP1 leads to a drastic reduction of NAD⁺ levels, affecting ATP production and cell functions (Alano *et al.*, 2010), which can lead to the development of chronic diseases such as cancer, diabetes, neurodegenerative diseases, and viral infections (Amé, Spenlehauer and de Murcia, 2004). Due to its role, PARP1 has been the subject of numerous studies which resulted in the discovery of many small-molecule inhibitors.

Four compounds received FDA approval as chemotherapeutic agents; hence, a large amount of both structural and ligand information is available for this target. However, the brain penetration of most of PARP inhibitors is strongly reduced by two efflux transporters (*anti-targets*) at the blood-brain barrier (BBB), namely P-glycoprotein (P-gp or Pgp) and breast cancer resistance protein (BCRP) (de Gooijer *et al.*, 2018), making these compounds less effective against brain cancer or neurodegenerative diseases such as Alzheimer's Disease or Parkinson's Disease.

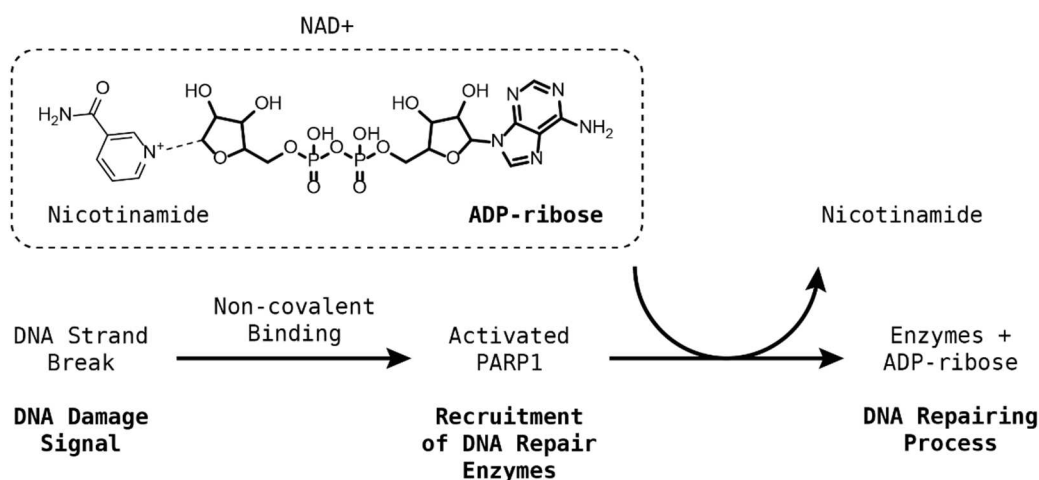


Figure 6.8: DNA repairing mechanisms mediated by PARP1: a DNA strand break activates PARP1 which in turn activates a series of enzymes responsible for DNA repairing.

The aim of this study is to use RENATE to propose new synthetically accessible inhibitors with improved brain penetration, by exploiting the variety of data available on PARP1, Pgp, BCRP, and the BBB.

The selection of PARP1 as a target for this case study is supported by the availability of crystal structures in the Protein Data Bank (PDB) complexed with five potent inhibitors which also share similar interactions with three residues in the DNA-binding domain (DBD): the FDA approved drugs (Olaparib (2014), Rucaparib (2016), Niraparib (2017), Talazoparib (2018)) and a known second generation inhibitor (PJ34 (Garcia Soriano *et al.*, 2001)). The structures of these compounds are reported in Figure 6.9. More details on their binding modes are reported in Section 6.4.4.5.

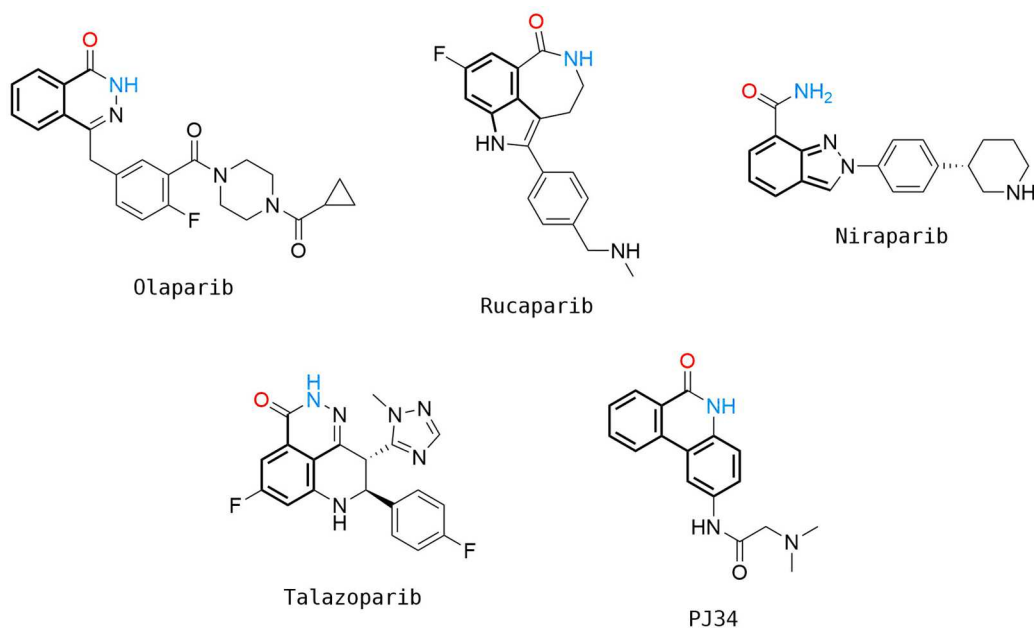


Figure 6.9: PARP1 inhibitors, for which crystallographic data is available in the PDB, showing similar interactions with several protein residues. Groups involved with hydrogen bonding as donors and acceptors are coloured in blue and red, respectively, while substructures involved with π - π stacking interactions are highlighted in bold.

6.4.3. Design Strategy

The main goal of this experiment was to design novel PARP1 inhibitors with improved brain penetration. To achieve this result, the activity towards PARP1 has to be maintained, the affinity towards the anti-targets (Pgp and BCRP) has to be reduced, and the penetration of the blood-brain barrier has to be promoted. The optimisation of multiple properties during the creation of novel compounds is referred to as multi-objective *de novo* design (Schneider, 2002), specifically *ligand-based* when the algorithm is mainly driven by the information extracted from reference molecules.

The FDA approved drugs reported in Section 6.4.2 constitute an excellent starting point for RENATE, which only requires to be modified on its scoring methods in order to produce novel candidates with the desired properties. For this purpose, eight scoring components were integrated within the algorithm described in Section 6.2. The new components are reported in Figure 6.10, which describes the combination of five *active* components applied sequentially to drive the algorithm at each step of the design, plus

three *passive* components placed at the end of the process to support the selection of the final candidates.

The *active* components consist of a similarity search module using pharmacophoric fingerprints, which aims to retrieve building blocks that produce interactions similar to the query fragments, and four machine learning models to score the structures generated by the algorithm. The models consist of a PARP1 activity regression (QSAR) model and Pgp-substrate, BCRP-substrate, and BBB-penetration classification models.

The *passive* components consist of a *reactive group conversion* unit, which attempts the conversion of the substructures that are identified as reactive into different functionalities, and substructure and property filters, and finally a docking model. Note that compounds that are converted by the *reactive group conversion* unit, are rescored by the *active* components. The use of docking completes the design process by simulating the interaction of the candidates within the binding pocket of PARP1, hence exploiting the availability of structural data in order to further discriminate them in three-dimensional space. The scoring components are described in more detail in Section 6.4.4.

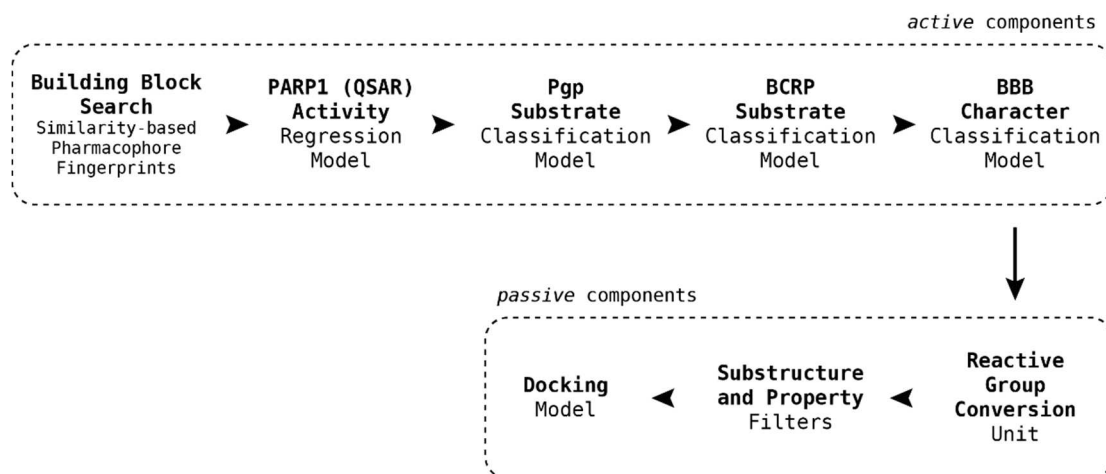


Figure 6.10: PARP1 design *scoring* module. *Active* components drive the algorithm at each step of the design, while *passive* components are applied at the end of the process to refine the selection of the most promising candidates.

6.4.4. Setup Selection

6.4.4.1. Similarity Search

Count FeatMorgan fingerprints (Radius 2) 1024-bit and Euclidean distance were selected as molecular descriptors and fingerprint distance metric, respectively, for the scoring of building blocks (*building block search* module). The selection of pharmacophoric fingerprints is aimed at maximising the chance of retrieving isosteric replacements of the query key fragments.

6.4.4.2. PARP1 QSAR Model

A small-molecule PARP1 activity dataset described by 2,371 entries was obtained from ChEMBL 24 database (ID: ChEMBL3105) on January the 24th 2019. Only entries associated with activity values and known units of measurement were retained, then activities were converted into pIC₅₀ values expressed as μM (10^{-6} molar) concentrations. Entries were sanitised, salts and ions were stripped, and canonical SMILES were generated using RDKit. SMILES structures associated with multiple activities were grouped and values were averaged. The operation returned 1,864 entries, which were described using the methods listed in Table 6.4. The 2D descriptor set was refined to 169 features via backward feature elimination.

Type	Molecular Descriptor	Features
Fingerprint	Avalon	1024
	FeatMorgan (Radius 2) (Binary)	1024
	FeatMorgan (Radius 2) (Count)	1024
	Morgan (Radius 2) (Binary)	1024
	Morgan (Radius 2) (Count)	1024
	RDKit	1024
Descriptor	2D (Atom/Bond Counts, BCUT, Chi and Kappa, GCUT, SlogP, SMR, VSA)	185

Table 6.4: Selection of molecular descriptors for the *scoring* components.

Each descriptor was evaluated by training a Random Forests (RF) regressor (`sklearn.ensemble.RandomForestRegressor`) using 80% of the data, and predicting the activities of the remaining 20%. True and predicted activities from each model validation

were used to compute R^2 , mean absolute error (MAE), and mean squared error (MSE). The operation was repeated 15 times per descriptor using random sampling to produce average metrics. Metrics definitions are reported on page 249. Results are described in Table 6.5, where metrics are reported with their mean standard deviations in brackets:

Molecular Descriptor	R^2	MAE	MSE
Avalon	0.73 (\pm 0.02)	0.46 (\pm 0.03)	0.42 (\pm 0.04)
FeatMorgan (Radius 2) (Binary)	0.74 (\pm 0.02)	0.46 (\pm 0.02)	0.41 (\pm 0.04)
FeatMorgan (Radius 2) (Count)	0.75 (\pm 0.02)	0.46 (\pm 0.03)	0.40 (\pm 0.04)
Morgan (Radius 2) (Binary)	0.74 (\pm 0.02)	0.46 (\pm 0.03)	0.40 (\pm 0.04)
Morgan (Radius 2) (Count)	0.75 (\pm 0.01)	0.45 (\pm 0.03)	0.39 (\pm 0.04)
RDKit	0.73 (\pm 0.02)	0.47 (\pm 0.02)	0.43 (\pm 0.04)
2D Descriptors	0.66 (\pm 0.03)	0.54 (\pm 0.03)	0.54 (\pm 0.06)

Table 6.5: PARP1 QSAR model validation results.

Table 6.5 shows similar results for all the fingerprints, which performed better than the descriptor model. Hyper-parameters of the best performing model were optimised by running a 5-fold cross-validation using Evolutionary Optimisation (Goldberg and Holland, 1988). The best configuration was then tested again on 20% of the data. Cross-validation and optimised model validation metrics are reported in Table 6.6.

Morgan (Radius 2) (Count) Model	R^2	MAE	MSE
Best Cross-validation Model	0.76	0.45	0.38
Optimised Model	0.79	0.41	0.31

Table 6.6: PARP1 QSAR Morgan (Radius 2) (Count) model best-cross validation and optimised model validation metrics.

6.4.4.3. Pgp-BCRP Substrate and BBB Penetration Classification Models

Three small-molecule datasets containing Pgp and BCRP substrate classification (i.e., substrate / non-substrate) data and BBB penetration data (i.e., BBB+ / BBB-) were collected from the literature. Datasets are described in Table 6.7:

Dataset	Entries (Class)	Source
Pgp	243 (substrate), 241 (non-substrate)	(Poongavanam, Haider and Ecker, 2012)
BCRP	164 (substrate), 99 (non-substrate)	(Hazai <i>et al.</i> , 2013)
BBB	1,437 (BBB+), 401 (BBB-)	(Yang <i>et al.</i> , 2019)

Table 6.7: Pgp, BCRP, BBB classification dataset descriptions.

Each dataset was standardised as illustrated in Section 6.4.4.2, then encoded using the descriptors in Table 6.4. Two 2D descriptor sets of 44 and 103 features were determined via backward feature elimination for Pgp and BCRP datasets, respectively. For each set, a series of RF classifiers (`sklearn.ensemble.RandomForestClassifier`) were trained using 80% of the data, and used to predict the classes of the remaining 20% of the data. The operation was repeated 15 times per descriptor using random sampling in order to produce average metrics.

True and predicted classes from the model validations were used to compute *weighted* Recall, Precision, F1-score, and MCC (see page 247). Average metrics are reported in Appendix B. The 2D descriptors resulted in improved models for Pgp and BCRP whereas, fingerprints performed better for the BBB models with count Morgan fingerprints selected. The most promising models were further investigated by optimising their hyper-parameters as reported in Section 6.4.4.2. Cross-validation and optimised model validation metrics are reported in Table 6.8.

Model	Validation	Recall	Precision	F1-score	MCC
Pgp	Best Cross-validation Model	0.76	0.76	0.76	0.51
	Optimised Model	0.80	0.81	0.80	0.62
BCRP	Best Cross-validation Model	0.73	0.73	0.73	0.42
	Optimised Model	0.79	0.80	0.78	0.55
BBB	Best Cross-validation Model	0.93	0.93	0.93	0.79
	Optimised Model	0.92	0.91	0.91	0.75

Table 6.8: Pgp, BCRP, and BBB model best-cross validation and optimised model validation metrics.

The models were further validated by classifying the PARP1 inhibitors selected for the design. The impact of Pgp and BCRP efflux on the blood-brain barrier distribution of the reference PARP1 inhibitors is described in the literature (Rottenberg *et al.*, 2008) (Jaspers *et al.*, 2015) (Durmus *et al.*, 2015) (Scott, 2017) (de Gooijer *et al.*, 2018) (Yu *et al.*, 2019).

Results from the validation are shown in Table 6.9, which suggests that the sequential application of the models can support the identification of substrates of Pgp and BCRP. For example, although the Pgp model predicted Rucaparib and Talazoparib

as non-substrates (i.e., wrong predictions), the BCRP model then flagged them as substrates; hence, in a *de novo* design context, these compounds would be discarded.

Query Ligand	BBB Character	Pgp Affinity	BCRP Affinity
Olaparib	BBB+	Substrate (Substrate)	Substrate (Substrate)
Rucaparib	BBB+	Non-substrate (Substrate)	Substrate (Substrate)
Niraparib	BBB+	Substrate (Substrate)	Substrate (Substrate)
Talazoparib	BBB+	Non-substrate (Substrate)	Substrate (Substrate)
PJ34	BBB+	Non-substrate (N.A.)	Substrate (N.A.)

Table 6.9: Pgp, BCRP, and BBB model classification results for the PARP1 inhibitors selected for the design (true classes are reported in brackets).

6.4.4.4. Reactive Group Conversion and Additional Filters

The *reactive group conversion* method was implemented as follows. The SMARTS definitions proposed by Hann and colleagues (1999) were selected for the identification of reactive patterns using RDKit. First, compounds identified as reactive were sent to the structure generator where only functional transformations were applied to them (e.g. introductions, conversions, eliminations, etc.) to attempt the removal of their reactive groups. This procedure can generate more than one product per reactive starting material. Second, transformed compounds were passed through the filter again, where those identified as still reactive (i.e., that could not be converted into something non-reactive) were discarded.

The substructure and property filters were configured to filter out compounds with specific substructure patterns (represented as SMARTS) and which violated more than one Lipinski rule, and were implemented using the RDKit Molecule Catalog Filter (`rdkit.Chem.rdfiltercatalog`) and the CDK Lipinski's Rule-of-five (`RuleOfFiveDescriptor`) modules, respectively. The following SMARTS definitions were selected: BRENK (unwanted functionalities related to potential toxicity or poor pharmacokinetics) (Brenk *et al.*, 2008), NIH (annotated compounds with problematic functionalities) (Doveston *et al.*, 2015), PAINS (pan assay interference patterns) (Baell and Holloway, 2010), ZINC (drug-likeness and problematic functional groups) (<http://blaster.docking.org/filtering/>).

6.4.4.5. Docking Model

Five crystal structures of *Homo sapiens* PARP1 catalytic domains in complex with their inhibitors were retrieved from the PDB. PARP1 and its ligands are represented in Figure 6.11, while PDB IDs and crystallographic resolutions are reported in Table 6.10.

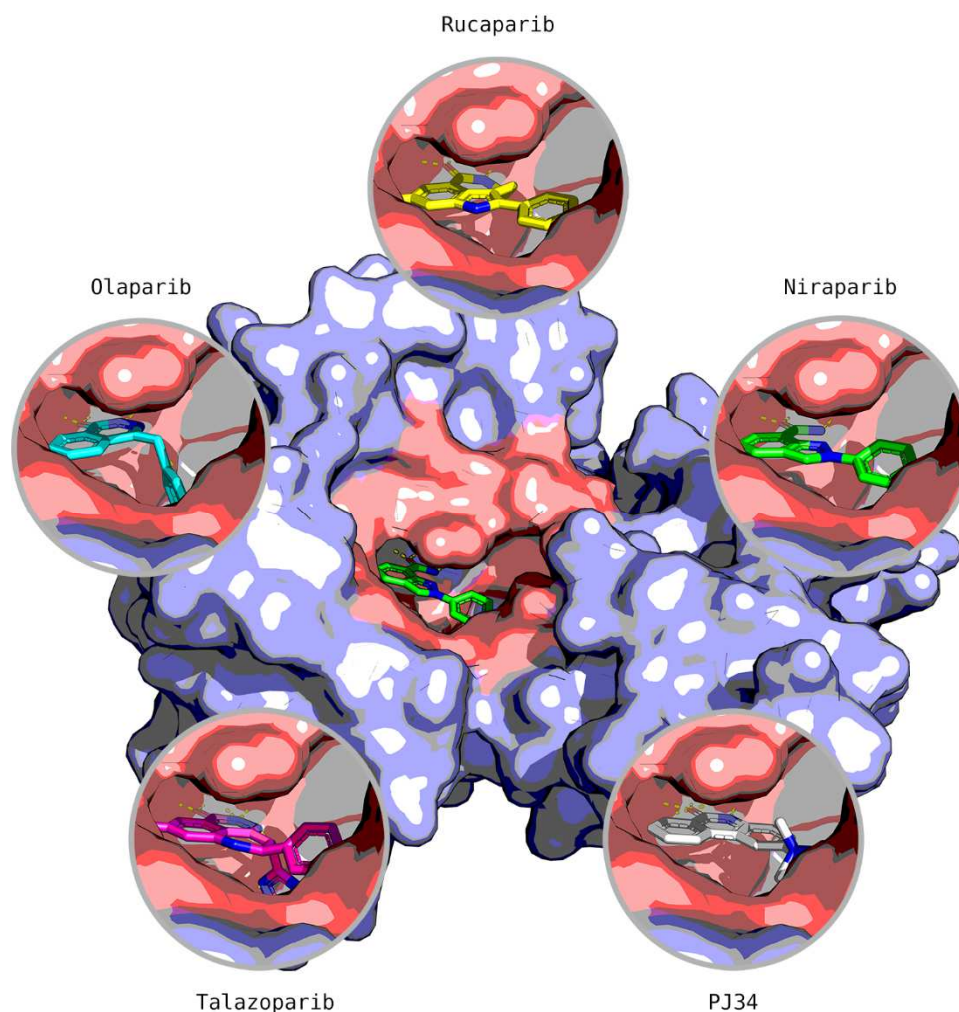


Figure 6.11: PARP1 catalytic domain (PDB ID: 4R6E) and its complexed ligands. The DNA binding pocket is coloured in salmon pink while the rest of the protein is in light purple.

Inhibitor	PDB ID	Resolution (Å)
Niraparib	4R6E	2.2
Talazoparib	4UND	2.2
Olaparib	5DS3	2.6
Rucaparib	4RV6	3.19
PJ34	4UXB	3.22

Table 6.10: PARP1 crystallographic data description.

The analysis of the selected structures identified three interacting residues that are conserved across the ligands: Gly863 and Ser904 are responsible for the formation of hydrogen bonds while Tyr907 produces π - π stacking interactions with the electron-dense areas of the inhibitors (Figure 6.12). These key interactions have been reviewed in several works (Ferraris, 2010) (Ekblad *et al.*, 2013) (Shen, Aoyagi-Scharber and Wang, 2015), and (Kumar, P.T.V. and Arunachalam, 2019).

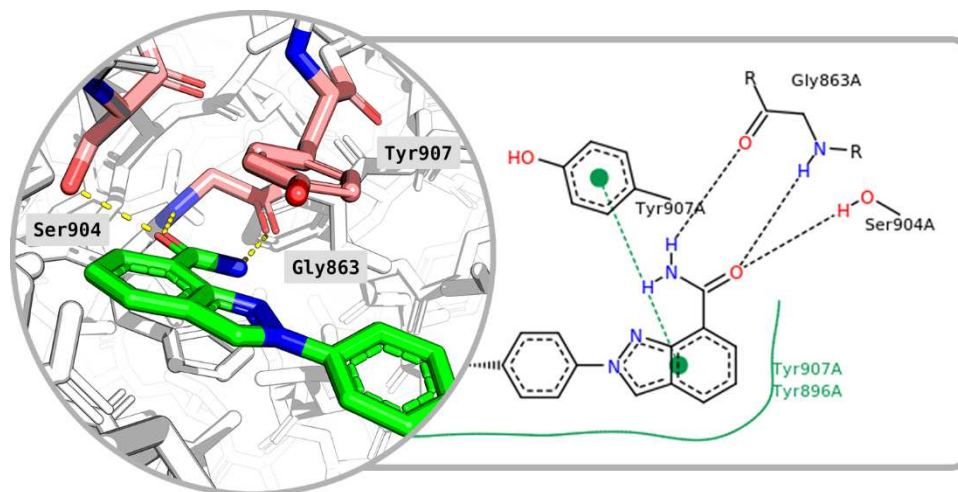


Figure 6.12: PARP1 3D (left) and 2D (right) key residue interactions with Niraparib. Yellow and black dashed lines indicate hydrogen bonds in 3D and 2D representations, respectively. Green solid lines show hydrophobic interactions and green dashed lines show π - π stacking interactions in the 2D diagram.

The superimposition of all five protein structures produced a good qualitative overlapping, suggesting their suitability for cross-docking (i.e., ligand interchangeability across structures). Therefore, the inhibitors were processed through a ligand preparation workflow as follows. Molecules were sanitised and aromatised using RDKit, salts and ions were stripped, protonation/deprotonation states were calculated at pH 7.4 using MOE (Chemical Computing Group ULC and ULC, 2019), stereocentres were enumerated and their minimised 3D conformations were produced using RDKit (MMFF94 optimisation). The five inhibitors plus four extra stereoisomers (i.e., (3R)-Niraparib and (11S,12S; 11R,12S; 11R,12R)-Talazoparib) were obtained.

The docking was carried out using GOLD (Jones *et al.*, 1997). The best resolved structure (PDB ID: 4R6E) was selected as the reference protein for the docking, and its

co-crystallised ligand (Niraparib) was selected to define the binding site. Water molecules were extracted, and PLP and GoldScore functions were selected for pose and interaction scoring, respectively. The GOLD parameters are reported in Table 6.11. Each of the five inhibitors (and the associated stereoisomers) were docked.

Parameters
autoscale = 2, radius = 10, save_lone_pairs = 0, early_termination = 0, docking_fitfunc_path = plp, rescore_fitfunc_path = goldscore,

Table 6.11: General PARP1 docking parameters in GOLD.

Each ligand generated 10 docking poses, which were compared with the original conformations of the co-crystallised ligands from the selected proteins (Table 6.10). The superimpositions between docked and original poses are reported in Figure 6.13 and Figure 6.14 for Niraparib and the other ligands, respectively. Mean PLP.Fitness and GoldScore.Fitness scores and the number of consistent poses (i.e., correct overlap with the co-crystal) are reported in Table 6.12, where the original drugs are shaded in grey.

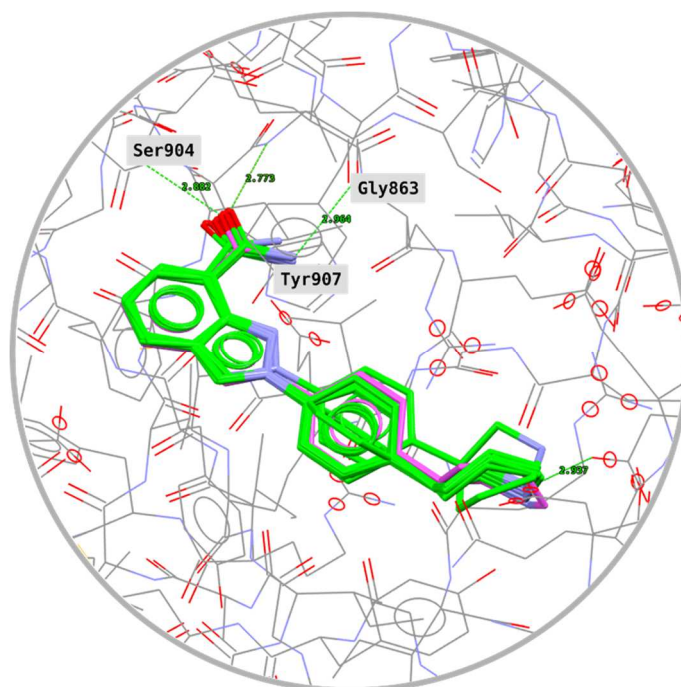


Figure 6.13: Overlap between docked (green) and experimental (purple) poses of Niraparib.

Figure 6.13, Figure 6.14, and Table 6.12 show that GOLD was generally capable of reproducing a good number of consistent poses and key interactions, providing strong

evidence for the validation of the docking model. The largest variance across poses was found for Olaparib, although the portion of the molecule responsible for the binding with the key residues still produced a tight visual overlap with its reference pose.

Query	Mean PLP.Fitness	Mean GoldScore.Fitness	Pose Consistency
Olaparib	91.87	51.16	8/10
Rucaparib	88.57	62.82	10/10
(3S)-Niraparib	96.02	50.45	10/10
(3R)-Niraparib	85.97	57.12	6/10
(11S,12R)-Talazoparib	83.54	47.01	10/10
(11R,12S)-Talazoparib	66.08	46.56	0/10
(11S,12S)-Talazoparib	68.70	55.61	2/10
(11R,12R)-Talazoparib	80.69	59.49	8/10
PJ34	86.98	53.67	10/10

Table 6.12: PARP1 docking validation scores and numbers of consistent poses per ligand. The original drugs are highlighted in grey.

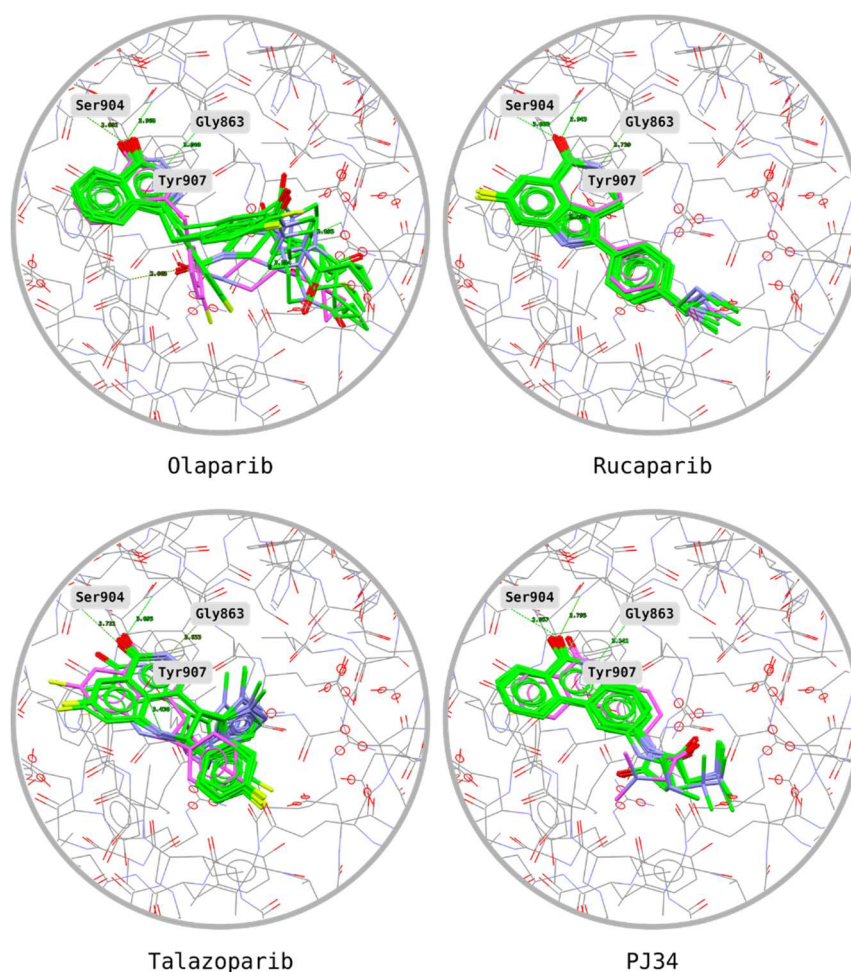


Figure 6.14: Overlap between docked (green) and experimental (purple) poses of the other inhibitors.

The inspection of the extra stereoisomers revealed that every structure produced lower PLP.Fitness scores and smaller numbers of interesting poses compared to the actual inhibitors. These results substantiate the reliability of the model for the identification of promising candidates.

An additional model accounting for the presence of active waters in the binding site was also evaluated, and produced results comparable with the model with no waters suggesting that the waters in the active site do not influence the binding mode of ligands in the region of interest; hence, the water model was not considered for the study.

6.4.5. Method

The five inhibitors described in Section 6.4.2 and 6.4.4.5 were processed by RENATE using the classified 92,530 USPD reaction vector database (Section 5.4.4), the 746,272 Enamine structures as a source of starting materials and reagents (see Section 6.3.3), and the components described in sections 6.4.3 and 6.4.4 as the *scoring* module. In the case of Talazoparib, the decomposition step failed due to using the fragmentation rules in BRICS which do not recognise the two sigma bonds between aromatic and aliphatic rings as a fragmentation pattern as discussed in Section 6.3.5. Therefore, the fragmentation rule was added to the BRICS set manually. The relevant parameters used in the experiments are reported in Table 6.13, while those not reported were configured with default values as reported in Table 6.1.

Query	Run	Parameters
Olaparib	1	minFragmentSize=1, MaxStartingMaterials=30, MaxReagents=25000, NumProductsCycle=2000
Rucaparib	1	minFragmentSize=5, MaxStartingMaterials=30, MaxReagents=25000, NumProductsCycle=2000
Rucaparib	2	minFragmentSize=5, MaxStartingMaterials=750, MaxReagents=1000, NumProductsCycle=4000
Niraparib	1	minFragmentSize=5, MaxStartingMaterials=30, MaxReagents=25000, NumProductsCycle=2000
Niraparib	2	minFragmentSize=5, MaxStartingMaterials=750, MaxReagents=1000, NumProductsCycle=4000
Talazoparib	1	minFragmentSize=1, MaxStartingMaterials=750, MaxReagents=1000, NumProductsCycle=4000
PJ34	1	minFragmentSize=5, MaxStartingMaterials=750, MaxReagents=1000, NumProductsCycle=4000

Table 6.13: PARP1 design RENATE parameters.

Only compounds with the highest predicted pIC₅₀ by ranking, no affinity with Pgp and BCRP (i.e., classified as ‘non-substrate’), and BBB+ character were retained at the end of each design step. Final candidates were processed through the ligand preparation workflow described in Section 6.4.4.5. The prepared candidates and their queries were docked using the no water model. The docking generated 10 poses per compound.

6.4.6. Results and Discussion

Results from the BRICS fragmentation are reported in Appendix B, where each query is associated with its corresponding key fragments (scaffold and substituents). The numbers of candidates generated from each design cycle and their enumerated stereoisomers are reported in Table 6.14:

Query	Run	Original Candidates	Enumerated Candidates
Olaparib	1	1,000	1,354
Rucaparib	1	678	769
Rucaparib	2	477	559
Niraparib	1	1,000	1,174
Niraparib	2	1,000	1,064
Talazoparib	1	990	1,339
PJ34	1	1,000	1,694

Table 6.14: PARP1 design original and enumerated candidates.

Table 6.14 provides some insights on the outcome of the design. Rucaparib and Talazoparib are characterised by complex scaffolds and connections such that they are less likely to match the reaction centres in the reaction vector database, which can explain why a smaller number of candidates were generated compared to the other queries. Conversely, Niraparib and PJ34, which have simpler scaffolds (i.e., that are more likely to match the reaction centres in the database), reached the maximum number of final candidates specified by the algorithm parameters. These results are in agreement with the general principles of reaction vector-based design: the structural characteristics of the starting materials and reagents affects the number of applicable reaction vectors, which in turn affects the number of final products. Table 6.14 also provides information on the stereochemical content generated by the ligand preparation workflow. This can be quantified by producing ratios between the number of compounds

in the original and enumerated sets: Rucaparib (1.13 - 1.17) and Niraparib (1.06 - 1.17) generated libraries with fewer chiral compounds, while Olaparib (1.35), Talazoparib (1.35), and PJ34 (1.69) yielded more stereochemical content.

6.4.7. Compound Selection

Results from the docking were manually inspected to identify some promising candidates per reference query. Poses were sorted by PLP.Fitness (primary objective) and GoldScore.Fitness (secondary objective) scores. Following this, compounds were selected on the basis of two parameters, which are the quality of the interactions with the key residues, and the number of consistent poses showing valid interactions with the key residues. The scores from the validation of the docking model (Table 6.12) also drove the selection of the candidates for synthesis. The inspection of the products generated using Talazoparib as a reference ligand did not suggest the selection of any candidate since most of the structures did not present sufficiently strong interactions with the key residues.

A total number of 20 compounds was selected from the docking, which are reported in Appendix B along with their predicted activities, pose consistencies, average and standard deviations of the binding scores across poses. All the selected compounds were predicted to be active in the order of sub-micromolar, to be not substrates of Pgp or BCRP, and to have BBB+ character. None of the selected molecules is available to purchase from two known compound suppliers (eMolecules and MolPort), and no data on these compounds is available in the PubChem or ChEMBL databases.

Note that one of the candidates generated from Niraparib (Row760) was converted to an analogue (Row760c) by the medicinal chemists to overcome some reagent unavailability issues (see Section 6.4.8). This analogue was designed by applying the same reaction vectors used for the original candidate (see Appendix B). Some examples of selected candidates and their corresponding queries in the PARP1 binding pocket are described in Figure 6.15, where Talazoparib is not reported due to the lack of promising candidates.

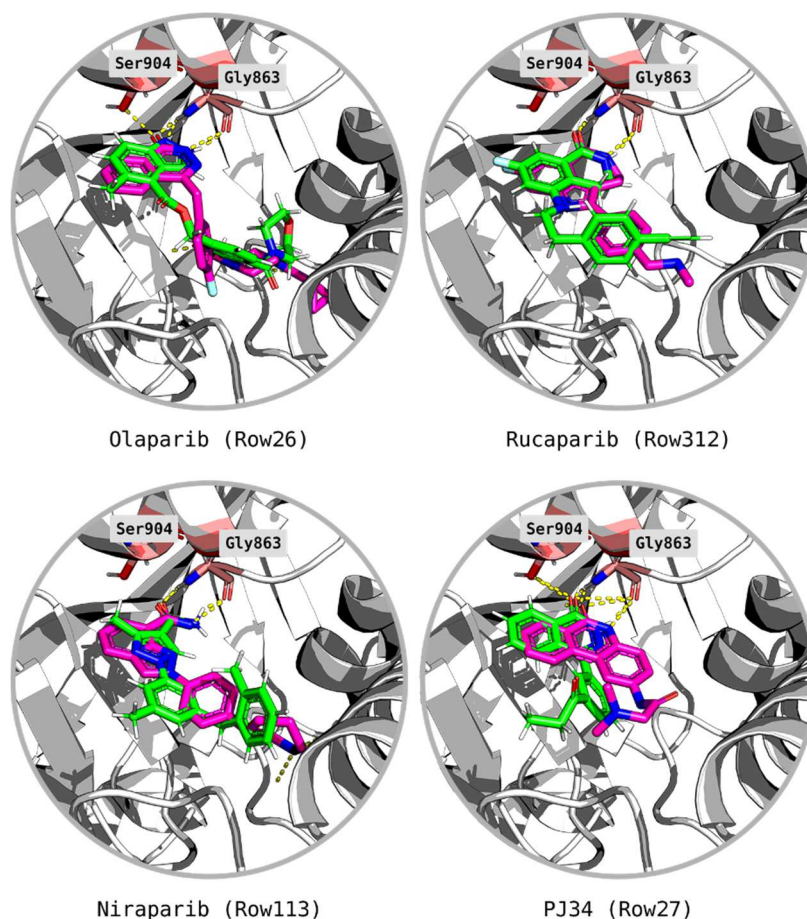


Figure 6.15: Overlap between candidates (e.g. Row26) (green) and reference drugs (e.g. Olaparib) (purple). Candidate IDs are reported in brackets. The residue Tyr907 is hidden to ease the view of the poses. Hydrogen bond interactions between protein and ligands are displayed in yellow.

Figure 6.15 shows a very tight overlap between candidates and reference drugs, and similar interactions with the key residues, although the designed structures are characterised by scaffolds and substituents that differ significantly from their original key fragments. Similarity values between candidates and queries, calculated using Morgan/FeatMorgan fingerprints (Radius 2) 1024-bit and Tanimoto metric, are also reported to quantify these differences. Olaparib and Row26 (0.26/0.30 similarities) have six- and five- membered rings fused with a benzene, respectively, that also differ in terms of functionalities and connections with the rest of their structures. Similar results were found for Niraparib and Row113 (0.26/0.38 similarity). Rucaparib and Row312 (0.40/0.40 similarity) are also quite diverse since Rucaparib has a very particular three-ring motif, whereas Row312 was generated using a scaffold identical to the one in

Olaparib. A very similar process occurred for PJ34 and Row27 (0.28/0.22 similarity) since PJ34 also has three fused rings, whereas Row27 was designed with a two-ring scaffold connected with an additional aromatic ring, which is described in the recent PARP1 patent literature (Peto, Jablons and Lemjabbar-Alaoui, 2016). Therefore, in both Rucaparib and PJ34 cases, RENATE performed successful scaffold hopping by adopting motifs that were already present in annotated compounds (i.e., interesting areas of the chemical space for PARP1 inhibitors), even though none of these structures directed the selection of the building blocks during the design. Furthermore, the inspection of the QSAR model data revealed that none of the entries contained the motif adopted by Row27, hence suggesting that RENATE can be used to propose novel scaffolds.

A limitation of RENATE is that it can sometimes produce structures that contain shuffled key fragments compared to those in reference ligands. This is due to the heuristics applied by the algorithm (Section 6.2) and the use of fingerprint-based scoring methods (Section 6.4.4), which are not capable of verifying whether the global shape and features of candidates match the references. Most of these molecules are generally filtered out by the scoring functions but some of them can still describe valid interactions by chance. These compounds might still be of interest but they are not produced using a rational approach. For example, Talazoparib reported a high number of these products due to its key fragment configuration and structural complexity, which as previously discussed, restricted the number of structures generated by the algorithm, hence reducing the chance of finding better solutions.

Examples of valid and invalid candidates from Talazoparib are reported in Figure 6.16. Talazoparib can be seen as a three fragment molecule of configuration B-A-C: The main interacting scaffold (A) plus two substituents (B and C, five- and six-membered rings, respectively), which are directly connected to the scaffold. Although Row2 and Row606 are both predicted to be active by the QSAR model, they are considered as valid and invalid candidates, respectively, since the first has a configuration identical to the query (B-A-C), while the second has a different configuration (A-B-C). Thus, the

selection of the final PARP1 candidates was confirmed prior to verification of whether the selected structures were generated correctly.

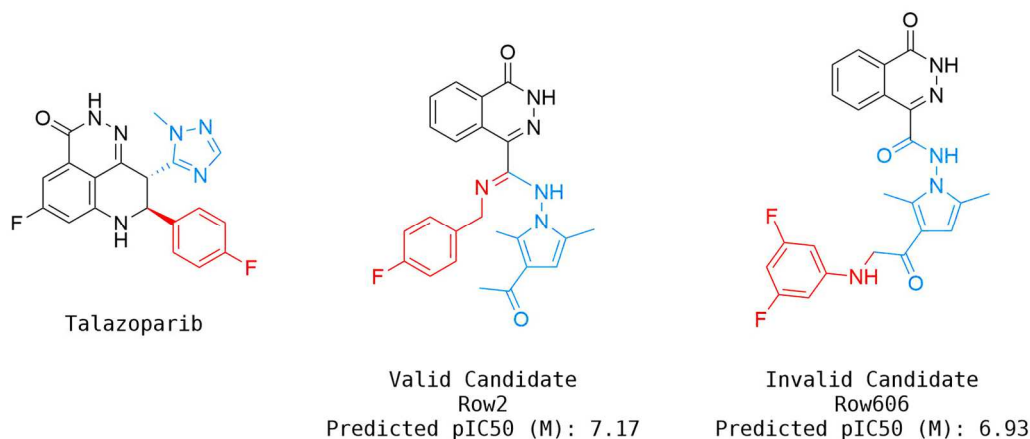


Figure 6.16: Examples of valid and invalid candidates designed by RENATE using Talazoparib as a query (A, B, and C fragments are coloured in black, blue, and red, respectively).

6.4.8. Compound Synthesis

As mentioned earlier, this prospective experiment has the major goal of demonstrating that reaction vectors can be used to suggest viable synthetic routes to *de novo* designed compounds. The proposed synthetic routes were therefore examined in consultation with medicinal chemistry experts, and only those compounds with what seemed like feasible synthetic routes were retained. A total of 8 compounds (2 per query) were submitted for synthesis by medicinal chemists at Evotec. The proposed routes were inspected in detail by medicinal chemistry experts and adjusted according to three factors: reagent availability (e.g. cost of building blocks, delivery times), additional steps (e.g. protection chemistry), and successful conditions (e.g. more robust reactions, catalysts, solvents, etc.). Proposed and adjusted routes are reported in Appendix C for the selected compounds, where additional steps are highlighted in dashed squares. The outcomes of the attempted syntheses are described in Figure 6.17.

The proposed reactions, focussing at the reaction centres they describe, were performed correctly to obtain sufficient quantities of Row26, Row86, Row514, Row760c, and Row847, whereas other candidates were obtained by applying different strategies.

In most cases, routes were modified because building blocks were not available or expensive (e.g. Row760 (0.1 g - \$ 779.0), Row847 (0.1 g - \$ 729.0)).

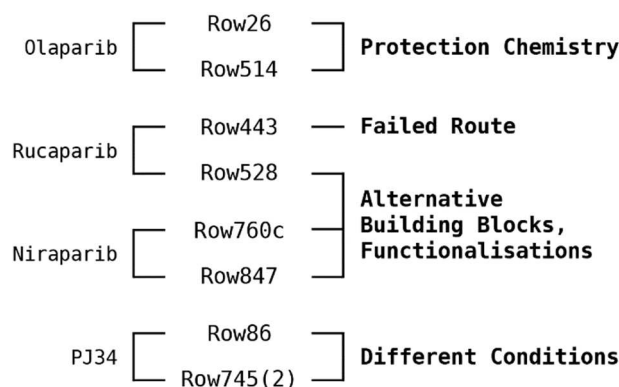


Figure 6.17: Compound synthesis summary scheme. The diagram describes on the left the names of the reference ligands (e.g. Olaparib), which are connected to their candidate structures (e.g. Row26). Candidate structures are associated with short descriptions on the right side of the chart, which describe the additional/alternative chemistry (e.g. Protection Chemistry) adopted to obtain the candidates.

Some procedures remained faithful to the original suggestions, whereas others required more pronounced adjustments. For example, Row86 and Row745(2) (PJ34 candidates) were obtained via organolithium conditions rather than Grignard generation since the latter was considered less robust; hence, these molecules were obtained through procedures very similar to their original routes. Note that Row745(2) and Row86 were produced as racemates of a single diastereomer and enantiomers, respectively. Other routes describing minor adjustments are those of Row26 and Row514 (Olaparib candidates), where some protection chemistry was introduced to overcome the limitations of the reaction vector approach (see Section 6.3.5).

More important modifications are described in the synthesis of Row847 (Niraparib candidate), which was initially formed using a precursor of the building block proposed by the algorithm. Consequently, the use of a precursor required further functionalisation (i.e., extra steps) in order to obtain the final compound. A similar process is described for Row528 (Rucaparib candidate) with the exception that the precursor also required a different reaction to form a C-C bond between the two aryl rings. The use of an alternative reaction, however, does not invalidate the route proposed by the algorithm.

Further adjustments were applied for Row760, which was converted into Row760c due to building block unavailability. This analogue was designed by preserving a route similar to that suggested for the original compound, yet the construction of the alternative building block required four extra steps including protection chemistry.

The remaining case is Row443 which could not be obtained even after the medicinal chemists' intervention. This compound also required the preparation of an alternative building block due to the prohibitive cost of the original carbonyl chloride. However, the reaction between the new reactant and the main scaffold yielded multiple products among which the desired structure was not observed. The formation of multiple products is probably due to the comparable reactivity of the indole N-H and the C-H alpha to the ketone in the main scaffold. Hence, the synthesis of this compound required major modifications which could not be applied due to time reasons.

A total number of 7 compounds out of 8 were eventually obtained. The precursor used in the syntheses of Row86 and Row745(2) and an intermediate of Row528 were also retained for the activity assay. These compounds are identified as intermediates (I) in Appendix C. Compound quantities, purities, and numbers of proposed and actual synthetic steps are reported in Table 6.15:

Query	Candidate	Quantity	Purity	Proposed Steps	Actual Steps
Olaparib	Row26	4.0 mg	94%	2	4
Olaparib	Row514	8.2 mg	88%	2	5
Rucaparib	Row528 (I)	74.6 mg	98%		
Rucaparib	Row528	3.7 mg	91%	1	4
Niraparib	Row760c	1.2 mg	96%	2	5
Niraparib	Row847	6.2 mg	98%	1	4
PJ34	Row86 (I)	Commercial	94%		
PJ34	Row745(2)	2.5 mg	87%	1	2
PJ34	Row86	15.0 mg	97%	1	1

Table 6.15: Summary of results of the PARP1 design synthesised compounds. The quantity and purity obtained are reported along with the proposed synthetic steps and the actual steps which were needed to obtain the compounds.

Table 6.15 shows that the compounds were obtained in sufficient quantity and purity for screening purposes, except for Row514 and Row745(2) which reported purities

less than 90%. The average ratio between the actual and proposed number of synthetic steps is 2.6. Although this coefficient has no statistical value, it still provides some indications that can be used for synthesis planning during the selection of candidates generated using reaction vectors.

6.4.9. Estimation of BBB Penetration

The second objective of the prospective experiment was to design compounds with desired therapeutic properties, in this case, represented by improved brain penetration compared with their references. The original plan involved the experimental testing of the synthesised compounds and their reference drugs, to first determine their activities, then a series of additional experiments would have been carried out to quantify the compound affinities to Pgp and BCRP, and to finally measure their diffusion through a model of the BBB. However, the experimental testing has been postponed indefinitely due to technical (i.e., problems with the activity assay) and financial issues.

As an alternative to the experimental tests, designed candidates and reference drugs were further evaluated computationally to predict a number of pharmacokinetics (PK) properties that have been connected to the ability of compounds to work effectively inside the blood-brain barrier. Compounds and their predicted properties are reported in Table 6.16. Row760 (original candidate) and Row760c (synthesised analogue) are both reported for a comparison purpose. The properties in Table 6.16 are described on their units and use in pharmacokinetics estimation on page 250. Properties were calculated at Evotec using the ChemAxon library (logD, HBD, TPSA) and three internal models (Caco2 A-B, CNS MPO v1 and v2).

Table 6.16 shows that, according to the guidelines reported on page 250, the selected candidates are generally associated with valid properties for effective BBB penetration, except for Row26, Row443, and Row745(2), which fall slightly outside the optimal 'logD' range (between 0 and 3); and Row760c, which did not improve the scores of its reference drug (Niraparib). As reported previously, Row760c was obtained by manual modification of Row760, which instead presents a higher 'CNS MPO' compared to Niraparib. The

fact that Row760c presents remarkably worse ‘Caco2 A-B’ and ‘CNS MPO’ scores compared to Row760, indicates that even very small structural modifications can result in dramatic changes in PK properties.

Query	Candidate	logD	HBD	TPSA	Caco2 A-B	CNS MPO
Olaparib	Query	1.98	1	82	5.6	5.08/4.83
Olaparib	Row26	3.29	1	70	9.3	4.60/4.99
Olaparib	Row514	2.86	1	85	4.9	4.72/4.90
Rucaparib	Query	1.30	3	57	3.2	4.47/3.72
Rucaparib	Row443	3.24	0	48	9.7	4.78/5.40
Rucaparib	Row528	2.00	2	41	6.8	4.54/4.05
Niraparib	Query	0.93	2	73	2.9	4.40/3.90
Niraparib	Row760	0.51	2	69	2.7	4.79/4.29
Niraparib	Row760c	0.26	3	80	1.6	4.58/3.83
Niraparib	Row847	1.97	1	55	10.7	4.50/4.25
PJ34	Query	1.63	2	61	7.4	5.50/5.00
PJ34	Row86	2.98	2	62	8.4	4.83/4.82
PJ34	Row745(2)	3.06	1	59	9.7	5.02/5.30

Table 6.16: PARP1 design selected candidates’ pharmacokinetic properties.

The analysis of ‘Caco2 A-B’ and both ‘CNS MPO’ scores points out that most of the selected candidates are predicted to have higher or similar scores compared to their corresponding drugs; hence, these results substantiate the strategy adopted for the design of the new inhibitors. Row847 is an example of a compound that reported an improvement of all objectives compared to its reference drug (Niraparib): the ‘HBD’ atoms were reduced from 2 to 1; ‘TPSA’ was reduced from 73 to 55; ‘Caco2 A-B’ was drastically increased from 2.9 to 10.7; ‘CNS MPO’ v1 was increased from 4.40 to 4.50, and v2 was increased from 3.90 to 4.25. The ‘CNS MPO’ scores in Table 6.16 also indicate that the queries Rucaparib and Niraparib present values that fall outside the range suggested by the guidelines for optimal brain penetration.

6.5. Conclusions

In this chapter, the development and implementation of a new pseudoretrosynthetic reaction-based *de novo* design algorithm referred to as RENATE have been described. RENATE has been validated computationally using a set of top prescribed drugs in the US to assess the principles and assumptions on which it is based. The validation of the

algorithm was also aimed at determining a promising setup for its use in real *de novo* design. Consequently, RENATE has been applied to a case study concerning the design of inhibitors with improved brain penetration for the biological target PARP1 to demonstrate that reaction vectors can provide useful suggestions for compound synthesis, and that they can be applied for successful *de novo* design. Results from the design were inspected to select a number of promising candidates, which were eventually synthesised using the routes generated by the algorithm as guidelines. The compounds obtained from the synthesis and their references were then evaluated on their estimated PK properties to obtain some preliminary evidence on the effectiveness of the design. The next chapter describes the development of a reaction classification model, and its application on the prediction of two external datasets.

Chapter 7: Reaction Classification

7.1. Introduction

Reaction classification has been considered as a topic of interest for many years, with applications ranging from the categorical indexing of reaction collections, to showing the formal similarity between different classes of reactions (Bawden, 1991) (Warr, 2014). The same techniques applied for drug discovery purposes can be used to analyse the content of reaction datasets, with the aim of identifying classes and routes that are more successful in medicinal chemistry, or to augment existing drug design tools, for example, by enabling the selection of specific types of transformations. Although different reaction classification methods have already been proposed in the past, and some rule-based algorithms such as CLASSIFY (Kraut *et al.*, 2013) and NameRxn (NextMove Software, 2017) are currently available on the market as commercial software, machine learning offers an alternative approach to these, by exploiting the increasing availability of public reaction data. In particular, classified collections of reaction examples can be used to train a supervised algorithm to generate a classification model.

In this chapter, a machine learning model for reaction classification is developed using the US pharmaceutical patent data processed in Section 5.4. The approach is broadly similar to that described by Schneider and colleagues (2015) yet with some important differences. First, the model capabilities are extended to classify a much larger set of reaction classes than in the published method. Second, the data used to train the model is described by means of the reaction vectors developed at Sheffield with the aim of maximising the compatibility between the classification model and the existing *de novo* design algorithms. Third, the model is also combined with a confidence estimation method in order to assess the reliability of individual predictions.

The chapter is organised as follows. Section 7.2 describes the development of a 50-class prototype model using a procedure similar to that reported by Schneider and

colleagues (2015). The main difference to Schneider’s work is the use of dynamic fingerprints that are derived from reaction vectors (see Section 5.3). The aim of this experiment is to provide some preliminary results on the performance of the dynamic fingerprints in reaction classification. Section 7.3 describes the scaling-up of the model to a much higher number of reaction classes in order to yield an effective tool that can be used on real collections of unclassified data. Note that the scalability of the model can be considered as a non-trivial challenge that is undertaken in this study. Each component of the model is investigated to determine its impact on the classification task, with the aim of maximising both model effectiveness and efficiency. The SHREC system (see Section 5.4.6) is also introduced at this stage and the best performing model is augmented with a confidence estimation module to assess the reliability of the predictions. Finally, Section 7.4 reports the application of the model on two datasets of unclassified reactions obtained from two different data sources in order to highlight the potentials and the limitations of this new approach, and to demonstrate how reaction classification can be used to get immediate qualitative and quantitative insights on the composition of reaction datasets.

7.2. 50-class Model

7.2.1. Introduction

A 50-class model is initially investigated to obtain some preliminary evidence on the best combination of fingerprint-type and machine learning method for reaction classification. This procedure also allows the comparison of the results with Schneider and colleagues (2015) whose experiments are also based on a 50-class model. Although works by Patel and colleagues (2009) and Hristozov and colleagues (2011) previously established that AP2+AP3 reaction vectors were most effective for *de novo* design, a systematic screening on several fingerprint types is still required to determine which configurations are best for classification purposes. Therefore, different versions of dynamic reaction vector fingerprints are investigated in this section. Three single atom-pair fingerprints are evaluated to examine the effect of increasing the proximal

environment encoded with the reaction centre itself. A combination of AP2+AP3 atom pairs, which corresponds to the default configuration used by the reaction vector structure generation tool, is also evaluated. At this stage, the class labels are those provided by NextMove Software based on NameRxn.

7.2.2. Data Selection

The four USPD fingerprint datasets described in Table 5.5 were processed as follows. First, the 50 most populated reaction classes were retained from the original sets, then classes were randomly sampled according to the minority class sizes. This technique is known as *balanced subsampling* or *downsampling* and it is generally used in machine learning to reduce the bias towards the most populated classes and enable a more representative validation. Second, atom-pair columns containing only zeros were removed. An example of fingerprint dataset pre-processing is illustrated in Figure 7.1 for the USPD AP2+AP3 dataset:

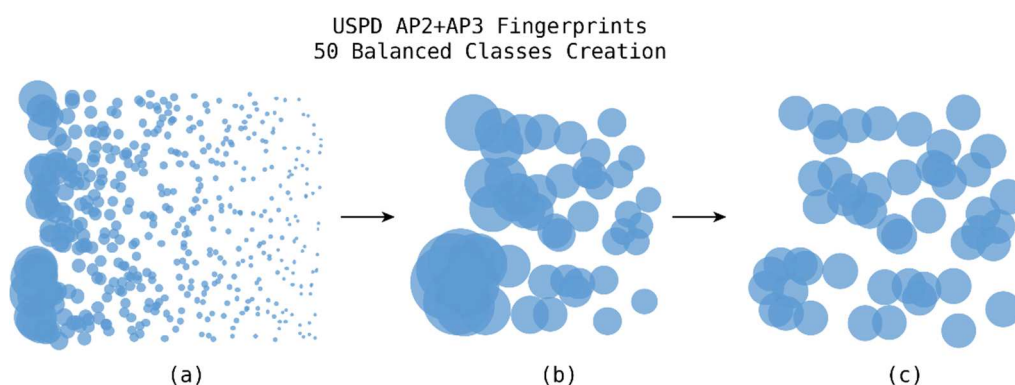


Figure 7.1: Creation of a balanced collection of 50 reaction classes from the USPD AP2+AP3 fingerprint dataset: (a) 727 imbalanced classes; (b) selection of 50 most populated classes; (c) downsampling according to the minority class.

The results from the dataset pre-processing are reported in Table 7.1, which shows numbers of reaction vectors and atom pairs for each fingerprint-type. Contents vary across atom-pair levels in the number of reaction vectors in each class, and therefore the total number of examples in each dataset. The pre-processed AP2 set describes a significantly lower number of unique reaction vectors compared to the other sets.

Pre-processed Dataset	Total Number of Reaction Vectors	Retained Atom pairs	Reaction Vectors per Class
AP2	10,000	1,167	200
AP3	25,650	2,103	513
AP2+AP3	25,700	3,146	514
AP4	25,500	2,292	510

Table 7.1: Pre-processed 50-class fingerprint datasets. The total number of unique reaction vectors is shown for the different types of fingerprints (number of rows in the datasets), along with the number of unique atom pairs (the number of columns in the datasets), and the number of unique reaction vectors in each class.

Reaction class coverages also differ from each other, as illustrated in Figure 7.2, which shows that the coverage of reaction classes in the pre-processed AP2 set is different to the extended fingerprint sets (i.e., AP3, AP2+AP3, and AP4), which share the same set of classes.

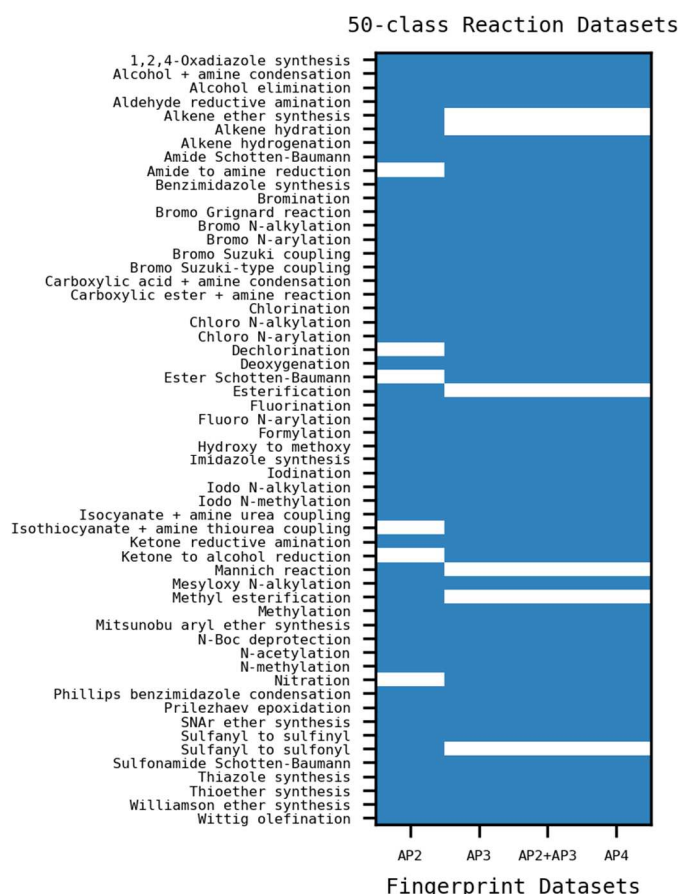


Figure 7.2: Reaction class coverage across fingerprint datasets. Represented classes are shown in blue and missing classes are in white.

As argued in Section 5.4.5, this is due to the nature of AP2 fingerprints which do not encode any information on the structural environment outside the reaction centre, hence resulting in lower discrimination compared to other atom-pair types. For example, the class “Ketone to alcohol reduction”, which describes a C=O reduced to a CH-OH, is represented by a small number of unique AP2 vectors, hence this class does not appear in the 50 most populated classes in the AP2 dataset. Therefore, the pre-processed AP2 set contains classes that describe a higher variety of reaction centres compared to the other pre-processed sets.

The datasets were further processed for the validation as follows. Each dataset was partitioned into a training set (40%) and a test set (60%) using stratified sampling on the reaction classes to preserve the distribution of examples across the classes. For the AP2+AP3 dataset, the training set was arbitrarily fixed at 10,000 reactions (~40%) to reproduce conditions similar to those reported by Schneider and colleagues (2015). The results of the partitioning process are shown in Table 7.2.

Pre-processed Dataset	Training Set	Examples per Class	Test Set	Examples per Class
AP2	4,000	80	6,000	120
AP3	10,660	205	14,990	308
AP2+AP3	10,000	200	15,700	314
AP4	10,200	204	15,300	306

Table 7.2: Training and test internal validation datasets across fingerprint types.

The USPDA fingerprint dataset described in Table 5.5 was also processed to yield an external set for the validation of the AP2+AP3 models. Only the classes contained in the pre-processed USPD set were retained, then the atom-pair columns in the USPDA set were adjusted as described in Section 5.3. Finally, reaction vectors which were already described in the USPD set were excluded from USPDA to ensure no overlap between the training and external validation sets. The final USPDA external validation set consisted of 15,193 reaction vectors. The USPDA set class distribution is plotted in Figure 7.3, which shows that the class composition of the USPDA set is imbalanced, with the minority class reporting fewer than 100 examples. Consequently, the USPDA classes were not downsampled to avoid the excessive reduction of the set size.

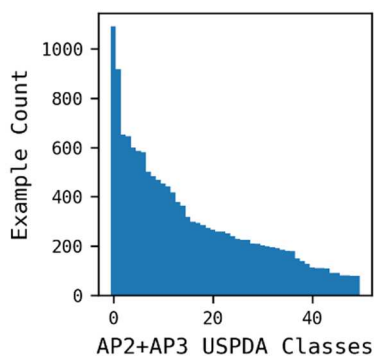


Figure 7.3: 50-class AP2+AP3 USPDA dataset class distribution sorted by vector count by descending order.

7.2.3. Method

The pre-processed datasets were used to validate four off-the-shelf machine learning classifiers from scikit-learn (<https://scikit-learn.org/stable/>) (Pedregosa *et al.*, 2011): Random Forests (RF) (`sklearn.ensemble.RandomForestClassifier`), K-Nearest Neighbors (kNN) (`sklearn.neighbors.KNeighborsClassifier`), Support Vector Machine (SVM) (`sklearn.svm.LinearSVC`), and Gradient Boosted Trees (GB) (`sklearn.ensemble.GradientBoostingClassifier`). Model parameters were not tuned at this stage. Classifier parameters are reported in Table 7.3. The training sets formed the input to the classifiers and the resulting models were used to infer the reaction classes for the entries in their corresponding test sets.

Classifier	Parameters
RF	<code>n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=4, random_state=11, verbose=1, warm_start=False, class_weight=None</code>
kNN	<code>number of neighbors=3, weights by distance='False'</code>
SVM	<code>cost=1.0, kernel='linear', degree=3, gamma=0, coef0=0.0, shrinking=True, probability=True, Epsilon=0.001, cache_size=3,866 MB, Nu=0.5, Loss-Epsilon=0.1</code>
GB	<code>loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, min_impurity_split=None, init=None, random_state=15, max_features=None, verbose=1, max_leaf_nodes=None, warm_start=False, presort='auto'</code>

Table 7.3: Classifier parameters.

7.2.4. Results and Discussion

Results were analysed quantitatively by comparing true and predicted classes from each individual class and as *macro* averages according to the following metrics: Recall, Precision, and F1-score. Metric definitions are reported on page 247. Average metrics for the internal validation are given in Table 7.4, which describes the performance of the 50-class classification models on the USPD test sets reported in Table 7.2, based on four fingerprint types and four classifiers. The RF classifier performed slightly better than the other models in all cases except the AP4 fingerprint dataset. This provides evidence of the robustness and versatility of this classifier, even if trained without optimised parameters. GB and SVM also performed well in most cases. The performance improves moving from AP2 to AP2+AP3 fingerprints, indicating that is necessary to encode some extended reaction environment to improve the discrimination between classes. The AP2+AP3 combined fingerprint reported better metrics compared to any of the single fingerprint types, and the performance is comparable with the models described by Schneider and colleagues (2015). AP4 fingerprints reported lower performance compared to AP3 and AP2+AP3 possibly due to the inclusion of noise from the extended environment that is not relevant for class discrimination.

Fingerprint-type	Classifier	Recall	Precision	F1-score
AP2	RF	0.80	0.80	0.80
	kNN	0.59	0.61	0.59
	SVM	0.76	0.77	0.76
	GB	0.78	0.80	0.79
AP3	RF	0.87	0.87	0.87
	kNN	0.75	0.76	0.75
	SVM	0.87	0.87	0.87
	GB	0.85	0.86	0.85
AP2+AP3	RF	0.90	0.90	0.90
	kNN	0.79	0.80	0.79
	SVM	0.89	0.89	0.89
	GB	0.89	0.90	0.90
AP4	RF	0.80	0.80	0.79
	kNN	0.65	0.67	0.65
	SVM	0.81	0.81	0.81
	GB	0.76	0.77	0.76

Table 7.4: Macro averages of recall, precision, and F1-score metrics across fingerprint types in the 50-class model internal validation.

The individual class performance was also evaluated by producing a series of normalised confusion matrices. Results are shown in Figure 7.4.

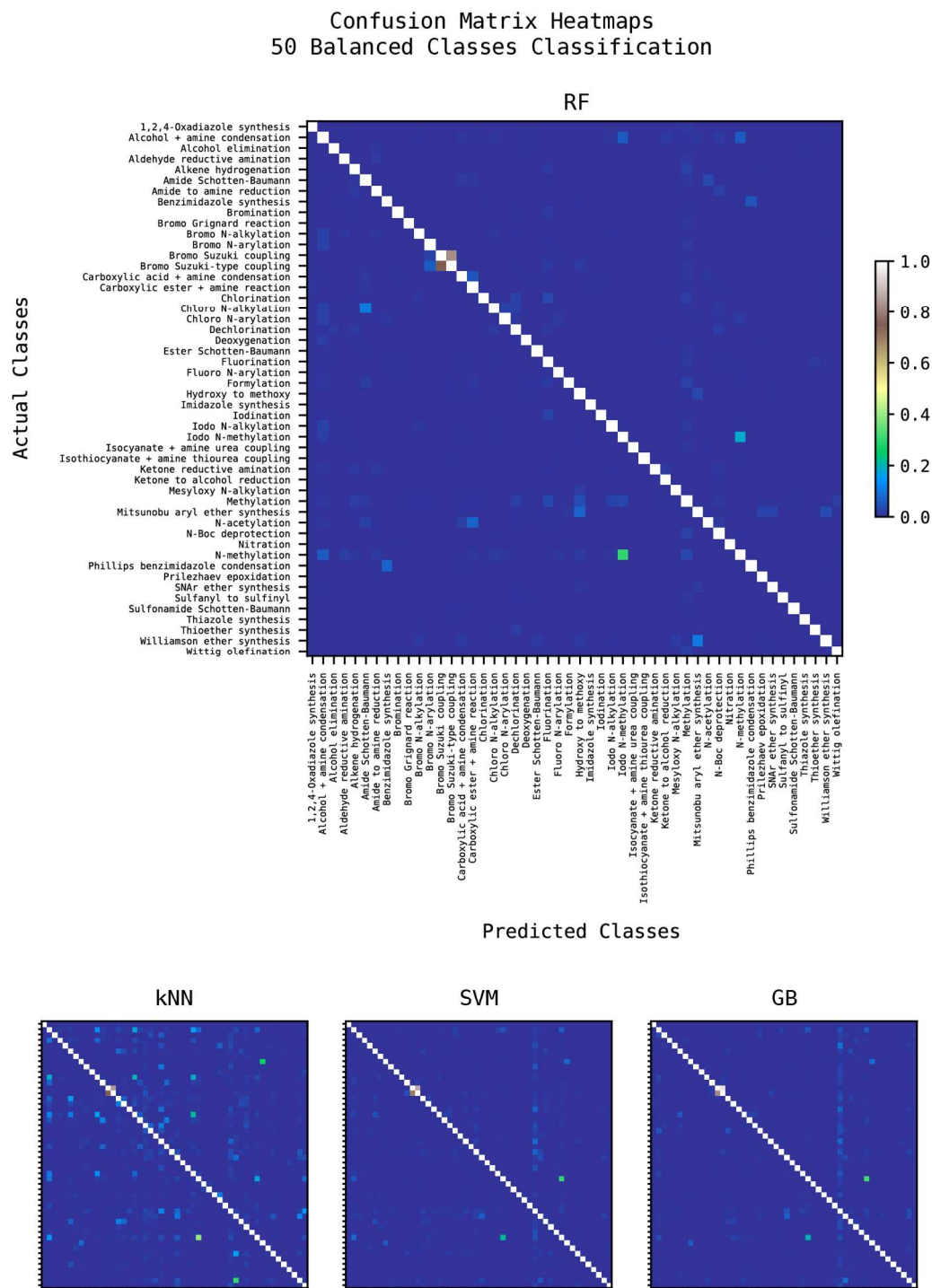


Figure 7.4: Normalised confusion matrices of the internal validations of the 50-class AP2+AP3 models across different classifiers.

Matrices show similar trends for most of the classes across different classifiers, suggesting a lack of discriminative information for those associated with lower scores. For example, coupling reactions such as “Bromo Suzuki coupling” vs. “Bromo Suzuki-type coupling”, and “N-methylation” vs. “Iodo N-methylation” cannot be distinguished effectively from each other using reaction vectors for two reasons. The first couple differs only in the reaction conditions through which the reactions occur, which are not encoded by reaction vectors, whereas the second pair represents the same reaction class in a generic and in a specific form. These findings are also comparable with the results described by Schneider and colleagues (2015).

The possible presence of different classes describing nearly identical vectors in the US patent sets was also foreseen as an issue in Section 5.4.6, where a new labelling system referred to as SHREC was introduced in order to merge these clashing classes. The results from the validation of the 50-class classification model further substantiated the replacement of the NameRxn labels with those described in the SHREC. Classes with small reaction centres such as “Methylation” or “Alcohol + Amine Condensation”, where extended environments are characterised by significantly different atom pair features, were also found to contribute negatively to the model performance. In these cases, the large difference between examples belonging to the same class can lead the classifier towards a misclassification of the unseen examples.

Next, an external validation was carried out on the pre-processed USPDA test set described in Section 7.2.2, to confirm the selection of AP2+AP3 as the default fingerprint-type for reaction classification. The distribution of examples in this dataset is imbalanced, hence it reflects more realistically the class distributions in datasets to which the model might be applied. The model performance was evaluated using *micro* and *weighted* Recall, Precision and F1-score metrics. *Macro* averages were not considered since they are not suitable for the evaluation of imbalanced datasets. Average metrics from the external validation are described in Table 7.5, which shows trends comparable to Table 7.4, where kNN reports the lowest performance. These results further supported the selection of AP2+AP3 as the default fingerprint for reaction classification.

Average	Classifier	Recall	Precision	F1-score
<i>Micro</i>	RF	0.86	0.86	0.86
	kNN	0.69	0.69	0.69
	SVM	0.85	0.85	0.85
	GB	0.85	0.85	0.85
<i>Weighted</i>	RF	0.87	0.86	0.86
	kNN	0.73	0.69	0.69
	SVM	0.86	0.85	0.85
	GB	0.87	0.85	0.86

Table 7.5: *Micro* and *weighted* averages of Recall, Precision, and F1-score metrics in the 50-class model external validation.

7.3. 336-class Model

7.3.1. Introduction

The 50-class model experiments only provided evidence for the selection of a suitable fingerprint-type for reaction classification since the application of these models on real datasets would lead to an output characterised by a maximum number of 50 labels, which is far smaller than the number of reaction classes that exist in reality. For this reason, the approach is extended to a larger dataset consisting of a much large number of reaction classes. The results from the 50-class model validation also suggested the optimisation of the model by replacing the NameRxn labels according to the SHREC system described in Section 5.4.6. The methods are presented first and include hyperparameter optimisation of the classifiers, selection of weights for reaction classes and confidence estimation using both in-built methods and conformal prediction. These are then followed by the results.

7.3.2. Data Selection

The AP2+AP3 USPD and USPDA fingerprint datasets described in Table 5.5 were processed as follows. Reaction classes described by fewer than 30 examples were filtered out from the USPD set then used as references to filter out the same classes from the USPDA set. Classes involving stereochemistry were also removed since stereochemistry is not encoded in the current implementation of reaction vectors. As before, reaction vectors contained in the USPD set were removed from the USPDA set to yield an

external set of unseen examples. Atom-pair columns containing only zeros were removed from the USPD dataset, then the atom-pair columns in the USPDA dataset were adjusted as described in Section 5.3. Finally, the datasets were further processed by replacing their NameRxn labels with their corresponding SHREC labels. Results are reported in Table 7.6.

Pre-processed Dataset	Total Number of Reaction Vectors	Number of Classes	Retained Atom pairs	Median Number of Vectors per Class
USPD	111,981	336	4,119	129.5
USPDA	25,026	335	4,119	29

Table 7.6: Pre-processed 30-example fingerprint datasets.

The resulting USPD and USPDA datasets described in Table 7.6 were selected as training/internal validation and external validation sets, respectively. The USPD dataset allows the training of a 336-class classification model, while the USPDA set contains the same classes except for the “C-C Bond Formation (Methylation) (Blanc chloromethylation)” class, which will therefore not be evaluated externally. The USPD set was then partitioned into 40% training and 60% test data using stratified sampling to preserve the distribution of examples across the classes. This resulted in 44,792 unique reaction vectors in the training set with a median number of 52 examples per class; and 67,189 unique reaction vectors in the test set with a median of 77.5 examples per class. Note that both USPD and USPDA sets now describe imbalanced distributions of examples per class as shown in Figure 7.5.

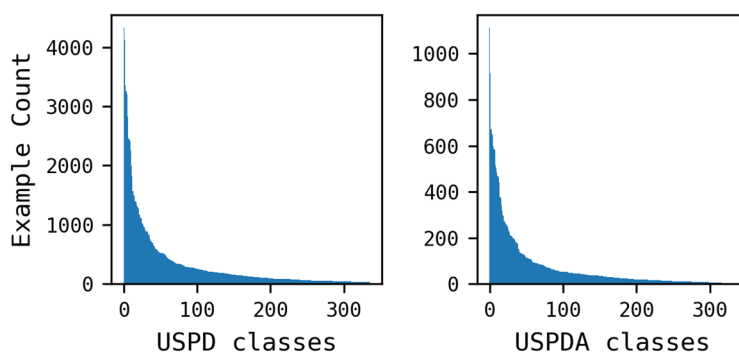


Figure 7.5: 30-example fingerprint dataset class distributions sorted by vector count by descending order.

7.3.3. Methods

7.3.3.1. Hashed and Dynamic Fingerprints

The validation of the fingerprint datasets generated in Section 7.3.2 and their comparison with their equivalent hashed versions distributed by RDKit (Landrum, 2016) were carried out. The four classifiers and their parameters reported in Table 7.3 were selected, along with an additional gradient boosted trees module named XGBoost (XGB) (`xgboost.sklearn.XGBClassifier`). The parameters used on the XGBoost classifier are reported in Table 7.7:

Classifier	Parameters
XGB	<code>nthread=6, booster='gbtree', base_score=0.5, colsample_bylevel=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3, min_child_weight=1, missing=None, n_estimators=100, objective='binary:logistic', reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=11, silent=1, subsample=1</code>

Table 7.7: XGBoost (XGB) parameters.

A preliminary validation was first performed by predicting the data that was used to train the classifiers to ensure that each model was generated correctly. Classifiers were trained using the 44,792 vector USPD training set described in Section 7.3.2, then used to infer predictions on the training data. Following this, models that produced valid metrics in the training data validation were evaluated using the 67,189 vector USPD test set and the USPDA external set described in Section 7.3.2 and Table 7.6, respectively.

7.3.3.2. Training Times

The determination of the training times for the selected classifiers is a necessary operation in order to determine the most efficient algorithms for hyper-parameter optimisation. This is particularly important since the new USPD training set describes a much higher number of reactions compared to the set used for the 50-class model, hence, the parameter optimisation of certain classifiers can be inefficient. The procedure was carried out as follows. An additional stratified sampling of 20% was performed on the 44,792 vector USPD training set described in Section 7.3.2. This procedure simulated

the creation of a 5-fold partition, which corresponds to the amount of training data typically used during an actual parameter optimisation. The classifiers described in Table 7.3 and Table 7.7 were trained using the 5-fold partition and their training times were determined. kNN was not evaluated due to its poor performance in the previous experiment. The procedure was performed using Python 3.6.2 on an Intel® Core™ i7-3770 CPU @ 3.40GHz × 8 processor workstation equipped with 32 GB RAM and running Ubuntu 16.04.

7.3.3.3. Hyper-parameter Optimisation

The two classifiers retained from the previous experiments, specifically RF and SVM, were investigated using two sequential strategies for hyper-parameters optimisation: Random Search (RS) and Evolutionary Optimisation (EO) were applied to first identify promising regions of parameters, then to exploit these regions in order to find the best classifier configurations, respectively. In RS, a distribution of parameters is configured along with a fixed number of tuning iterations (e.g. 100). The algorithm works by providing the best coverage of that distribution without being biased by the model scores. This way, parameters are explored without focusing on specific regions, and promising areas can be identified then exploited by EO algorithms, which are instead driven by the scores generated from previous configurations. In both RS and EO, a k-fold cross-validation procedure was adopted to minimize training times and to generalise the model performance.

The RS was run using a scikit-learn module as the main framework to drive the search (`sklearn.model_selection.RandomizedSearchCV`), whereas the EO was run using a scikit-learn-compatible framework (<https://github.com/DEAP/deap>). Distributions of parameters for RS and EO are reported in Table 7.8 and Table 7.9, respectively. The RS distribution was defined arbitrarily, whereas the EO distribution was determined upon the results from the RS. The SVM distribution in the EO corresponds to the same used in the RS since no areas to cut-off were identified. The number of iterations for the

RS was set to 100, and *weighted* F1-score was used as a reference to drive the EO algorithm.

Classifier	Parameter Distributions
RF	<code>{"n_estimators": np.arange(50, 250, 5), "max_depth": sp_randint(20, 100), "max_features": sp_randint(50, 150), "min_samples_split": sp_randint(2, 20), "min_samples_leaf": sp_randint(1, 5)}</code>
SVM	<code>{'C': list(scipy.stats.expon(scale=100).rvs(size=100)), 'max_iter': [1000, 5000, 10000, 15000], 'class_weight': ['balanced', None]}</code>

Table 7.8: Random Search parameter distributions.

Classifier	Parameter Distributions
RF	<code>{"n_estimators": np.arange(50, 150, 5), "max_depth": np.arange(70, 120), "max_features": np.arange(120, 150), "min_samples_split": np.arange(2, 5), "min_samples_leaf": [1]}</code>
SVM	<code>{'C': list(scipy.stats.expon(scale=100).rvs(size=100)), 'max_iter': [1000, 5000, 10000, 15000], 'class_weight': ['balanced', None]}</code>

Table 7.9: Evolutionary Optimisation parameter distributions.

Both RS and EO were sequentially performed as follows. Each parameter configuration was tested by running a 5-fold cross-validation on the 44,792 vector USPD training set described in Section 7.3.2, then results from all the iterations were inspected to determine the best parameters according to their corresponding *weighted* F1-scores. RF and SVM were then retrained using their best configurations and used to infer predictions on the 67,189 vector USPD test set and the USFDA external set described in Section 7.3.2 and Table 7.6, respectively.

7.3.3.4. Class Weights

Imbalanced training datasets are known to cause a bias towards highly populated classes when applied in data learning (Witten *et al.*, 2016). A potential solution to this problem is to apply class weights during the classifier training, in order to modulate the bias associated with those classes, as well as reducing the importance of the low performing ones. The RF model selected from the previous experiment was investigated using three sets of weights: default settings of 1.0 (i.e., all classes weighted equally) as a control; *balanced* weights; and *empirical* weights.

Balanced weights are calculated in scikit-learn according to the heuristic shown in Equation 7.1 inspired by King and colleagues (2001), where the weight (w_y) associated with a given class 'y' is calculated by dividing the total number of examples ($n_samples$) by the product of the total number of classes ($n_classes$) and the number of examples in the class 'y' ($y_samples$):

$$w_y = \frac{n_samples}{(n_classes \times y_samples)}$$

Equation 7.1: Pseudo equation for the calculation of balanced weights.

Empirical weights were determined by inspecting the validation results from the RF Evolutionary Optimisation. In particular, results were inspected by sorting the classes in descending order of the number of false positives, then a number of arbitrary weights were assigned to those classes associated with a high number of false positives and a low F1-score. False negatives were not investigated because single arbitrary weights cannot be used directly to promote certain classes since scikit-learn already applies the maximum weight (1.0) on them by default. A plausible approach for class promotion would involve the reduction of all the other class weights, but this procedure would require a higher manual intervention with the risk of over-tuning the classifier. The weight settings selected for the experiment are reported in Table 7.10:

Classifier	Weights
1	None
2	'balanced'
3	{'C-N Bond Formation (N-arylation) (Bromo)': 0.8, 'C-C Bond Formation (Methylation)': 0.3, 'C-N Bond Formation (Amination)': 0.6}
4	{'C-N Bond Formation (N-arylation) (Bromo)': 0.6, 'C-C Bond Formation (Methylation)': 0.1, 'C-N Bond Formation (Amination)': 0.4}
5	{'C-N Bond Formation (N-arylation) (Bromo)': 0.8, 'C-C Bond Formation (Methylation)': 0.1, 'C-N Bond Formation (Amination)': 0.8}

Table 7.10: Weight settings.

The selected weight sets were tested as follows. Five RF classifiers were configured with the best parameter configurations determined from the EO procedure. Each classifier was tuned using a weight set from Table 7.10, then trained using the 44,792

vector USPD training set described in Section 7.3.2. The biased classifiers were used to infer predictions on the 67,189 vector USPD test set and the USPDA external set described in Section 7.3.2 and Table 7.6, respectively.

7.3.3.5. Training Data Efficiency

The experiments reported so far have been conducted using classifiers fed with only 40% of the training dataset. An additional study to monitor the effect of increasing amounts of training data on the classifier performance was carried out to collect information on the possible acquisition of more training examples. Optimised parameters and weights for the RF classifier are summarised in Table 7.11:

Parameters	Class Weights
{'n_estimators': 145, 'max_depth': 113, 'max_features': 142, 'min_samples_split': 2, 'min_samples_leaf': 1}	{'C-N Bond Formation (N-arylation) (Bromo)': 0.6, 'C-C Bond Formation (Methylation)': 0.1, 'C-N Bond Formation (Amination)': 0.4}

Table 7.11: Random Forests (RF) optimised parameters and weights.

The procedure was carried out as follows. The optimised RF classifier was trained using the 111,981 vector USPD set described in Table 7.6, according to increasing amounts of data: The training set size was varied from 2% to 100% in 2% intervals using stratified sampling with three datasets produced for each size by varying the seed in the stratification algorithm. More specifically, the training datasets were generated by sampling the USPD set according to the trends: 2%, 4%, 6%, etc. This technique is preferable to the gradual removal of examples (e.g. 98%, 96%, 94%, etc.) when trying to maximise the example diversity across datasets. After each training step, the classifier was used to infer predictions on the USPDA set described in Table 7.6.

7.3.3.6. Confidence in Predictions

Individual class metrics such as Recall, Precision and F1-score indicate how a model performs overall in the classification of a given class. However, it can also be useful to obtain confidence estimates for individual predictions. Two confidence estimation methods were investigated to achieve this. Built-in probability scores (see Section 4.3.2.1) and Conformal Prediction (CP) (see Section 4.4). In both methods, values are

included in a range between 0.0 and 1.0, where higher values correspond to higher confidence. The selected methods were applied to determine levels of confidence that separate true and false predictions. These levels can be used as thresholds for the selection of entries that are likely to be predicted correctly.

- **Built-in Probability Scores**

The procedure carried out for the determination of the confidence levels using built-in probability scores is as follows. The optimised RF classifier was trained using the entire 111,981 vector USPD set described in Table 7.6, then used to infer classes and probabilities on the USPDA external set described in Table 7.6. Only the highest probability and its corresponding predicted class were retained for each test instance.

- **Conformal Prediction**

A Python implementation (<https://github.com/donlnz/nonconformist>) of the ICP framework described in Section 4.4 was integrated with the optimised RF classifier identified in the previous stages.

The procedure carried out for the determination of the confidence levels using CP is as follows. The optimised RF classifier was trained and calibrated using 90% and 10% of the 111,981 vector USPD set described in Table 7.6, respectively. Although a higher percentage of training data is usually recommended for use of CP for QSAR prediction (Eklund *et al.*, 2015), for example 30%, increasing the accuracy of the conformal predictor by decreasing the accuracy of the underlying algorithm was not thought to be desirable. The partitioning was performed using a stratification algorithm on the reaction classes. Results are described in Table 7.12. The classifier was then used to infer classes along with their confidence and credibility scores on the USPDA external set described in Table 7.6.

Partitioned Dataset	Number of Reactions	Median Number of Vectors per Class
USPD Proper Training set (90%)	100,782	116.5
USPD Calibration set (10%)	11,199	13

Table 7.12: USPD *proper training* and *calibration* sets.

7.3.4. Results and Discussion

7.3.4.1. Hashed and Dynamic Fingerprints

The results of applying the models to the training data are shown in Table 7.13. True and predicted classes from the model validations were used to compute *weighted* F1-scores for the selected configurations. Recall and Precision are not reported due to their high correlation with F1-score (see Table 7.4). Invalid models are highlighted in grey and labelled as “Not Valid”. The column headed Sheffield represents the AP2+AP3 dynamic reaction fingerprints.

Classifier	RDKit 1024	RDKit 2048	RDKit 4096	Sheffield (4119 bits)
RF	1.00	1.00	1.00	1.00
kNN	0.52	0.52	0.52	0.89
SVM	0.02	0.02	0.02	0.96
GB	0.05	0.01	0.79	0.02
XGB	0.96	0.96	0.96	0.94

Table 7.13: *Weighted* F1-scores of the training data validation.

Table 7.13 shows that not all the models were trained correctly, although RDKit and dynamic fingerprints describe very similar contents. The failure of some models, shown by the shading in Table 7.13, can be attributed to both training data composition and the parameters used to train the classifiers. For example, the SVM classifier worked only with the use of dynamic fingerprints, thus the hashing procedure could have generated data that the algorithm misinterpreted with that particular configuration. However, the interpretation of the failures still remains difficult to fully accomplish due to the multifactorial nature of the process of model training.

Table 7.14 shows the performance of the models on the internal test set and the external validation set. Small increases in performance are seen as the vector length increases, with the dynamic fingerprints performing generally better than their static equivalents. This can be generally explained by the higher number of atom pairs (4,119) encoded by the dynamic fingerprints. However, this trend is inconsistent for the kNN classifier where the RDKit reports significantly lower scores, hence the non-hashed nature of the dynamic fingerprints potentially improved the validation metrics. Due to

its poor performance compared to the other methods, kNN is excluded from the experimental pipeline. In addition, the GB-RDKit 4096-bit model, which performed efficiently during the training stage, resulted in a zero F1-score in both internal and external validations. This result can be possibly interpreted as an extreme overfitting in the training data. The results from this experiment demonstrate that the dynamic fingerprints are generally comparable to the reaction fingerprints provided by RDKit when applied for reaction classification using machine learning.

	Classifier	RDKit 1024	RDKit 2048	RDKit 4096	Sheffield (4119 bits)
Internal set	RF	0.86	0.88	0.87	0.90
	kNN	0.18	0.18	0.18	0.79
	SVM	Not Valid	Not Valid	Not Valid	0.89
	GB	Not Valid	Not Valid	0.00	Not Valid
	XGB	0.87	0.89	0.89	0.90
External set	RF	0.80	0.82	0.82	0.85
	kNN	0.02	0.02	0.02	0.72
	SVM	Not Valid	Not Valid	Not Valid	0.84
	GB	Not Valid	Not Valid	0.00	Not Valid
	XGB	0.82	0.84	0.84	0.86

Table 7.14: *Weighted* F1-scores of internal and external validations.

7.3.4.2. Training Times

Training times from the 5-fold partition training are reported in Table 7.15:

Classifier	Absolute Training Time	Relative Training Time
RF	3.773 seconds	1
SVM	7.170 seconds	1.9
GB	5555.772 seconds - 92.60 minutes	1,472.5
XGB	95942.313 seconds - 1,599.00 minutes - ~27 hours	25,468.7

Table 7.15: 5-fold partition training times.

The results show vastly increased training times for the gradient boosted-based classifiers in comparison with other algorithms. More specifically, both GB and XGB reported training times thousands of times longer than the other classifiers. Note that although both estimators were configured with a number of models equal to 100, they reported very different times compared to each other. In addition, this result is in contrast with the configurations used for these two classifiers. The scikit-learn implementation of GB used in this experiment did not support multi-threading, whereas

XGB was configured in order to use a number of threads equal to 6, hence XGB was expected to run faster than GB. Therefore, GB and XGB are excluded from the experimental pipeline due to their long training times.

7.3.4.3. Hyper-parameter Optimisation

True and predicted classes from the validations were used to compute *weighted* and *micro* F1-scores. The cross-validation and the internal-external validation results from the RS are reported for RF and SVM in Table 7.16 and Table 7.17, respectively.

Classifier	Best Parameters	Weighted F1-score
RF	{'max_depth': 82, 'max_features': 148, 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 85}	0.83
SVM	{'C': 5.91, 'class_weight': None, 'max_iter': 15000}	0.79

Table 7.16: RS cross-validation best parameters and scores.

Classifier	Internal Validation		External Validation	
	Weighted F1-score	Micro F1-score	Weighted F1-score	Micro F1-score
RF	0.90	0.90	0.85	0.85
SVM	0.87	0.87	0.85	0.85

Table 7.17: RS internal and external validation F1-scores.

Results from Table 7.17 shows that RF did not improve compared to the validation using default parameters described in Table 7.14, substantiating the robustness of this classifier, whereas worse and better performance are reported in the internal and external validations of SVM, respectively. These results suggest a possible reduction of overfitting for the SVM classifier.

Figure 7.6 and Figure 7.7 show a series of 2D scatter plots for RF and SVM, respectively, where parameters and their corresponding *weighted* F1-scores are described on the x and y axes, respectively. The RF plots in Figure 7.6 show wide model robustness for every range of parameters, except for *max_depth* and *min_samples_leaf* where trends show some poorly performing regions. Trends also indicate the presence of multiple local traps such as the region between the values 50 and 70 of for *max_depth*, and the region between the values 80 and 120 of for *max_features*. These local optimum areas were excluded from the RF parameter distribution used in the EO stage.

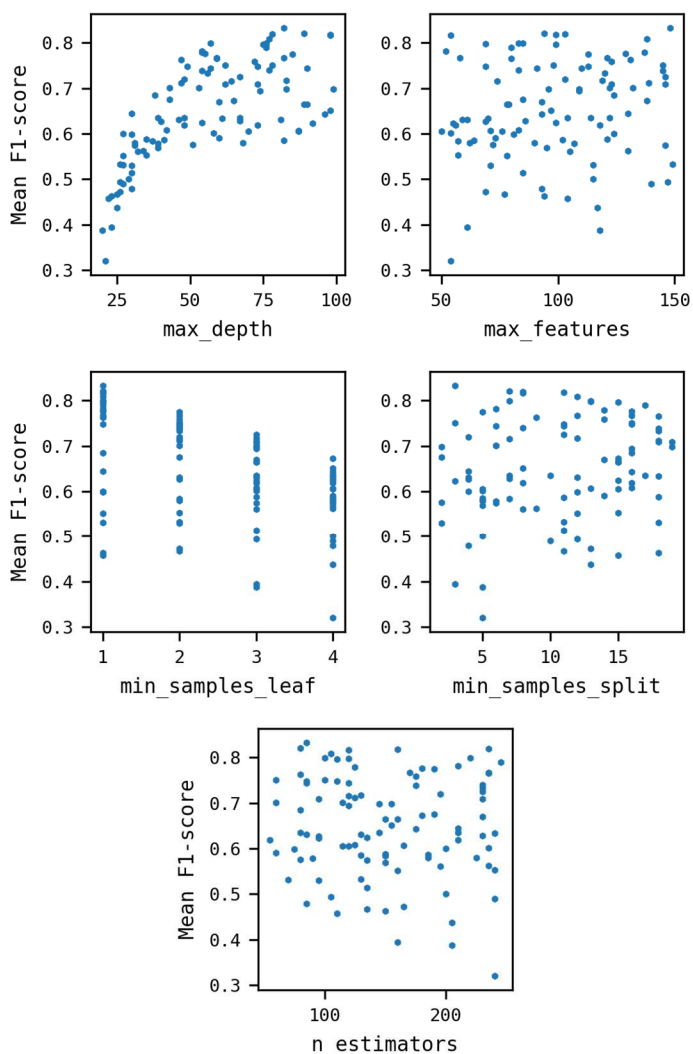


Figure 7.6: RF performance-against-parameter trends in the RS.

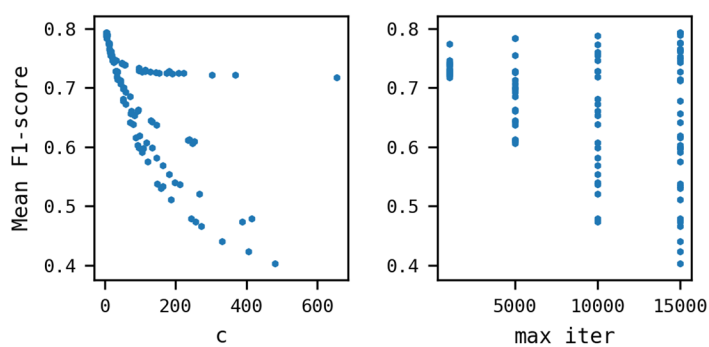


Figure 7.7: SVM performance-against-parameter trends in the RS.

The SVM trends described in Figure 7.7 report a rapid decrease in performance as the c parameter increases. This parameter is very important since it is strictly related to the ability to generalize the model fitting for the correct prediction of unseen data.

The parameter *max_iter* shows a broader distribution of metrics for a higher number of iterations, whereas it reports a narrower but worse distribution of performance for lower numbers of iterations. These trends do not suggest the presence of any local trapping area, thus they were not used to determine a narrower parameter distribution for the EO stage.

The cross-validation and the internal-external validation results from the EO are reported for the RF and SVM classifiers in Table 7.18 and Table 7.19, respectively:

Classifier	Best Parameters	F1-score
RF	{'n_estimators': 145, 'max_depth': 113, 'max_features': 142, 'min_samples_split': 2, 'min_samples_leaf': 1}	0.89
SVM	{'C': 0.94, 'max_iter': 10000, 'class_weight': None}	0.88

Table 7.18: EO cross-validation best parameters and scores.

Classifier	Internal Validation		External Validation	
	Weighted F1-score	Micro F1-score	Weighted F1-score	Micro F1-score
RF	0.90	0.90	0.85	0.85
SVM	0.89	0.89	0.84	0.84

Table 7.19: EO internal and external validation F1-scores.

Results from Table 7.19 show that RF did not improve compared to the previous results described in Table 7.14 and Table 7.17, whereas SVM reported worse performance compared to the metrics found in the RS internal and external validations. This result suggests that the EO identified an SVM configuration less capable of catching the unseen data variance, although Table 7.18 describes better cross-validation scores for both classifiers compared to the metrics found in the RS. These results suggest the selection of RF as a main classifier for the development of the reaction classification model.

Metrics (i.e., Recall, Precision, and F1-score) for the individual reaction classes for the internal and external validations of RF are reported in Figure 7.8, which describes slightly better trends in the internal validation compared to the external validation, also confirming that the metrics are highly correlated to each other. Each plot describes a very broad variance in performance when the number of training examples in a class is lower than 100. This behaviour is shown by a large percentage of the classes in the training set since the median number of examples corresponds to 52 (see Section 7.3.2).

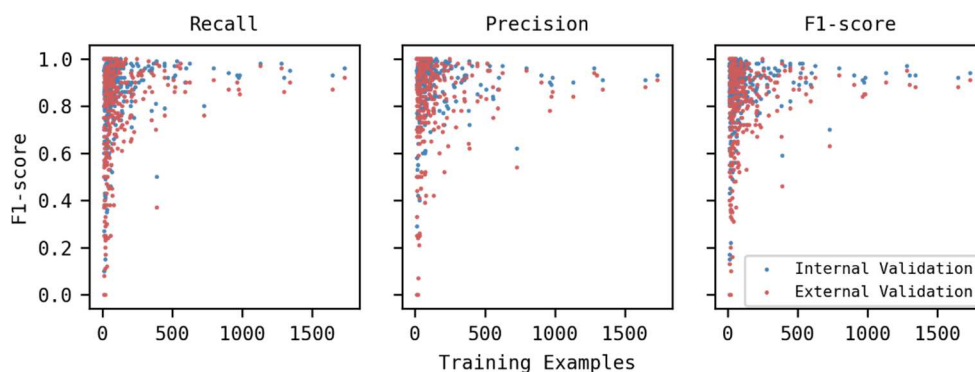


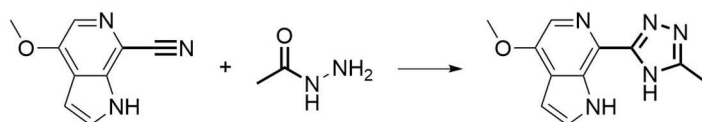
Figure 7.8: Correlation scatter plot of training data and performance metrics for internal (USPD test set - blue) and external (USPDA external set - red) validations of the RF classifier. The x-axes represent the number of examples in each class in the training data. Each dot represents one reaction class.

The large performance variance associated with low numbers of training examples can be explained by the intrinsic nature of each reaction class and the varieties of reaction centres that it potentially describes. For example, although the “Synthesis (1-2-4-Triazole)” class contains only 13 examples in the training data, a F1-score equal to 0.97 is reported on 19 unseen examples (internal validation). This class performed well because it generally describes very similar reactions centres. Conversely, a F1-score of 0.7 is reported on the “C-C Bond Formation (Methylation)” class for 1094 test reactions (internal validation), although its corresponding training set contained 729 examples. This is because a generic methylation can involve many different reaction centres, and therefore it is not an easy class to match using the current implementation of reaction vectors. The same issue was already reported in the analysis of the 50-class model results described in Section 7.2.4. Hence, classes described by a small number of AP2s and a high variance in AP3 features are all affected by this issue. Results suggest the possible introduction of new features to improve the performance of those classes. Examples of training and test examples that show this behaviour are reported in Figure 7.9. Figure 7.8 also highlights that the variance in performance is strongly reduced when the number of training examples increases. A minimum threshold of 150 examples per class returns lowest F1-score equal to 0.59 for the “C-N Bond Formation (Amination)” class (internal validation), which, as for methylation, consists of a small reaction centre presented in a wide variety of extended environments. A threshold of 250 examples per class returns

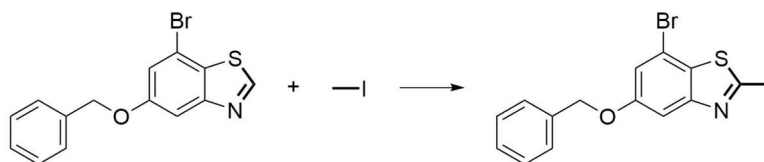
lowest and median F1-scores equal to 0.70 and 0.93, respectively. Therefore, the use of a bigger and balanced source of training data is expected to yield better performing models in the future.

Training set examples:

Synthesis (1-2-4-Triazole)
US07348337B2-2008-29_1

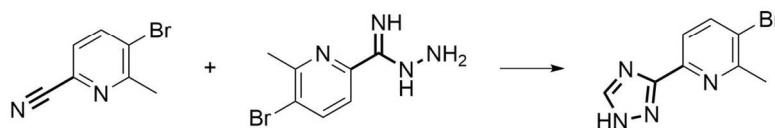


C-C Bond Formation (Methylation)
US09376441B2-2016-150



Test set examples:

Synthesis (1-2-4-Triazole)
US08569494B2-2013-51



C-C Bond Formation (Methylation)
US04250099-1981-1

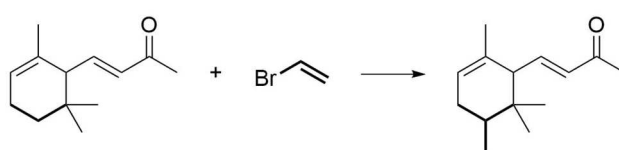


Figure 7.9: Examples of classes involving more (“C-C Bond Formation (Methylation)”) or less (“Synthesis (1-2-4-Triazole)”) variable reaction centres.

7.3.4.4. Class Weights

The ten classes with the highest numbers of false positives in the EO validations along with their F1 scores, are described in Table 7.20 and Table 7.21, respectively. Three classes with large numbers of false positives and low F1-scores were associated with three sets of empirical weights: “C-N Bond Formation (Methylation)”, “C-N Bond

Formation (Amination)”, and “C-N Bond Formation (N-arylation) (Bromo)”. “C-C Bond Formation (Methylation)” was heavily penalised due to its high number of false positives, whereas “C-N Bond Formation (N-arylation) (Bromo)” and “C-N Bond Formation (Amination)” were subjected to minor penalisations.

Reaction Class	False Positives	F1-score
C-C Bond Formation (Methylation)	538	0.68
C-C Bond Formation (Condensation) (Carboxylic acid + amine)	210	0.93
C-N Bond Formation (N-alkylation) (Bromo)	203	0.94
Functional Conversion (Hydrogenation) (Alkene to alkane)	190	0.94
C-N Bond Formation (N-methylation)	173	0.90
C-N Bond Formation (N-arylation) (Chloro)	172	0.90
C-N Bond Formation (Amide formation) (Schotten-Baumann)	166	0.93
C-N Bond Formation (Amination)	141	0.56
C-N Bond Formation (N-arylation) (Bromo)	138	0.78
C-O Bond Formation (Etherification) (Williamson)	136	0.91

Table 7.20: 10 classes with the highest number of false positives in the EO internal validation.

Reaction Class	False Positives	F1-score
C-C Bond Formation (Methylation)	346	0.61
C-N Bond Formation (N-arylation) (Chloro)	127	0.83
C-C Bond Formation (Condensation) (Carboxylic acid + amine)	125	0.88
Deprotection (N-t-Butyloxycarbonyl) (N-Boc)	123	0.87
Functional Conversion (Hydrogenation) (Alkene to alkane)	117	0.90
C-C Bond Formation (Coupling) (Suzuki) (Bromo)	106	0.82
C-N Bond Formation (N-methylation)	104	0.85
C-N Bond Formation (N-arylation) (Bromo)	95	0.70
C-N Bond Formation (N-alkylation) (Bromo)	92	0.91
C-O Bond Formation (Etherification) (Williamson)	77	0.85

Table 7.21: 10 classes with the highest number of false positives in the EO external validation.

True and predicted classes from the model validations were used to compute *micro* and *weighted* F1-scores for the selected weight sets to determine the best weight configuration. Results are reported in Table 7.22:

Classifier	Internal Validation		External Validation	
	<i>Weighted</i> F1-score	<i>Micro</i> F1-score	<i>Weighted</i> F1-score	<i>Micro</i> F1-score
1	0.90	0.90	0.85	0.85
2	0.87	0.87	0.82	0.82
3	0.90	0.90	0.85	0.85
4	0.90	0.90	0.85	0.85
5	0.90	0.90	0.85	0.85

Table 7.22: Class-weighted internal and external validation F1-scores.

Table 7.22 describes the variation in performance across the selected configurations. Classifier 1 was trained using default weights (i.e., equal to 1.0 for each class and was used as a control for the process. Classifier 2 (*balanced* weights) scores are generally worse for both the internal and external validation compared to the other configurations. A possible explanation for this is related to the imbalanced nature of the validation sets. A classifier trained with some bias towards the most populated classes might actually perform better than an unbiased classifier on those datasets. However, this hypothesis could not be verified due to the lack of balanced test data. Classifiers 3, 4, and 5 (*empirical* weights) do not report any variation in the global performance of their corresponding models.

The ten classes with the highest number of false positives are also reported for the classifiers 3, 4, and 5, for both internal and external validations in Figure 7.10 and Figure 7.11, respectively, in order to determine the effect of the *empirical* weights on the individual classes:

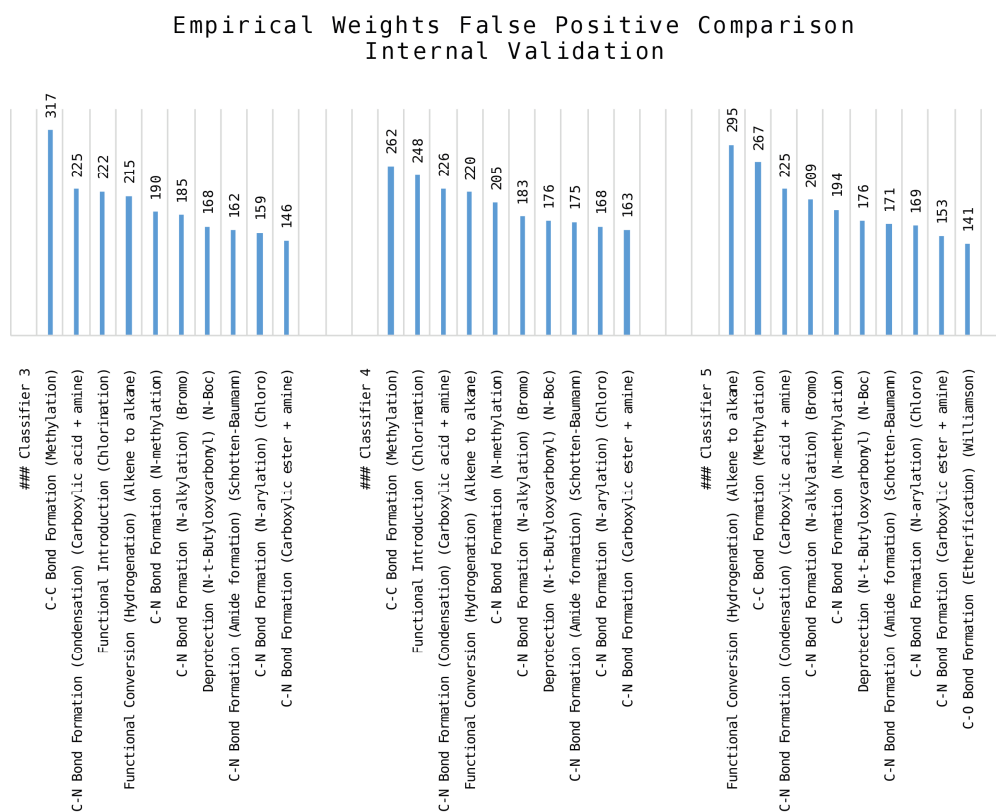


Figure 7.10: False positive trends for classifiers 3, 4, 5 from the internal validation.

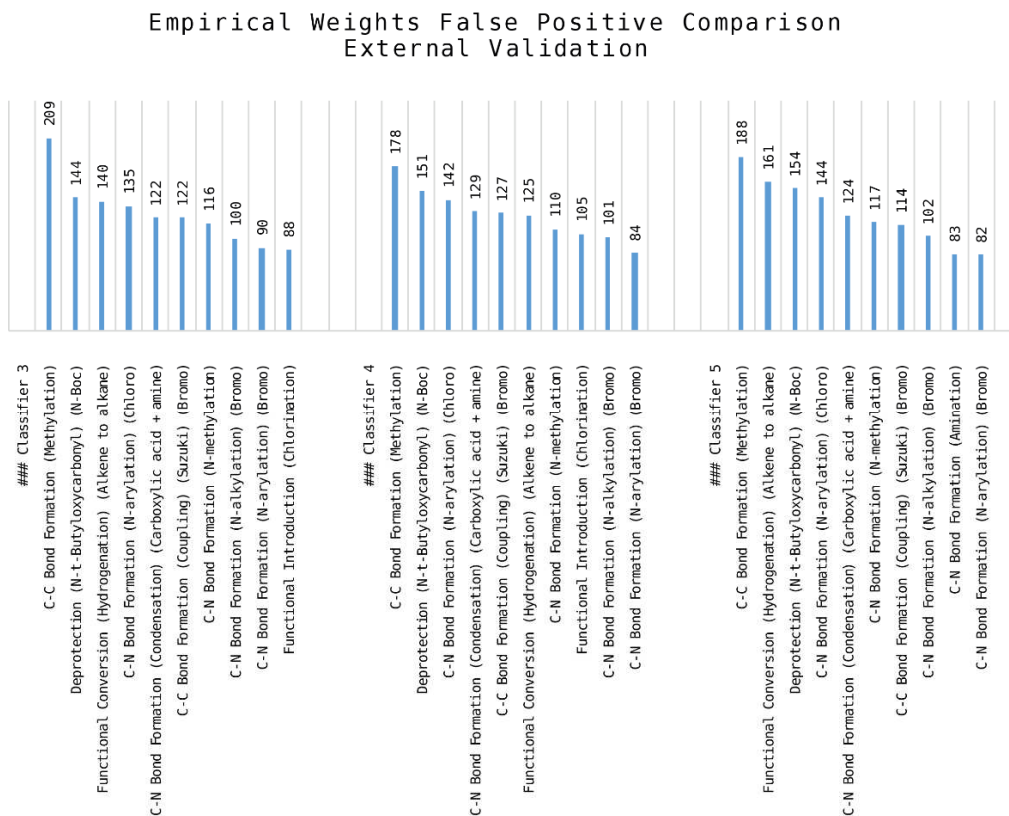


Figure 7.11: False positive trends for classifiers 3, 4, 5 from the external validation.

Figure 7.10 and Figure 7.11 show that in both validations *empirical* weights generally improved the performance of the class “C-C Bond Formation (Methylation)”, which decreased in the number of false positives and increased in F1-score (not reported here) compared to the original results in Table 7.20 and Table 7.21. “C-N Bond Formation (Amination)” and “C-N Bond Formation (N-arylation) (Bromo)” improved as well since they disappeared from the top ten classes with the highest number of false positives. Classifier 4 reported the best false positive distribution in both validations, therefore, suggesting the selection of this weight configuration for the classification model.

7.3.4.5. Training Data Efficiency

True and predicted classes from the model validations were used to compute *micro* and *weighted* F1-scores to investigate the effect of the training set size. Results are reported in Figure 7.12, which shows consistent trends on both plots, demonstrating

that the combination of RF and dynamic vectors produced efficient models at almost any percentage of the training data. *Micro* and *weighted* F1-score trends are closely comparable, except for very low amounts of training data (i.e., lower than 10%) where the *weighted* scores are slightly worse than the *micro* scores. The best *micro* F1-scores were found using a percentage of training data higher than 86%, whereas the best *weighted* F1-scores were found with a percentage of training data higher than 92%. The general performance trends show that after a steep increase in performance between 0 and 20%, the curve reaches a plateau beyond which there are diminishing gains.

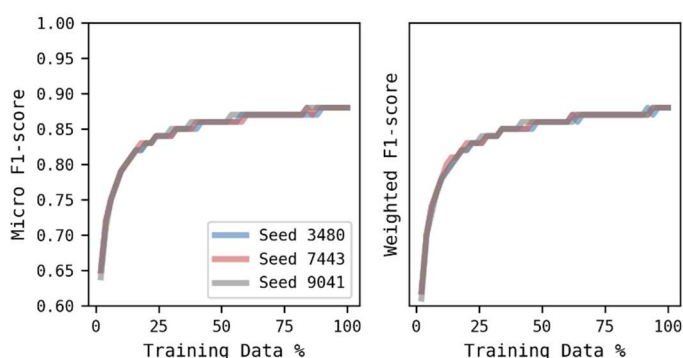


Figure 7.12: *Micro* (left) and *weighted* (right) F1-scores trends at increasing amounts of training data on the prediction of the external data set.

7.3.4.6. Confidence in Predictions

- **Built-in Probability Scores**

The confidence levels associated with true and false predictions, using the built-in probability scores, were evaluated as follows. Entries were rounded at two decimal positions using the *half up* method, for example, the value 0.015 was rounded to 0.02, and -0.015 was rounded to -0.01 . Entries were then binned into 98 bins ranging from 0.03 to 1.00.

The absolute numbers and ratios of true and false predictions associated with each probability level are plotted in Figure 7.13, which shows on its left side that the number of correct predictions increases steadily as the probability scores increase. However, the

chart does not show clearly how false predictions change due to their lower absolute numbers compared to the true predictions. This trend is better described on the right side of the chart. For instance, a probability equal to 0.22 results in 49% true and 51% false predictions.

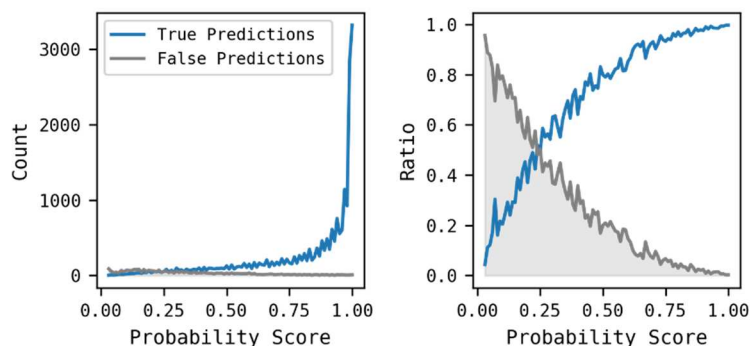


Figure 7.13: Absolute numbers (left) and ratios (right) of true and false predictions associated with each level of probability.

Estimations were additionally evaluated by selecting a series of arbitrary probability score cut-offs along with their corresponding *weighted* F1-scores and percentages of filtered entries. Results are reported in Table 7.23:

Probability Cut-off	<i>Weighted</i> F1-score	Percentage of Filtered Entries
0.00	0.88	0.00
0.15	0.90	3.65
0.25	0.93	7.81
0.35	0.94	13.37
0.45	0.96	17.05
0.60	0.97	25.26
0.80	0.99	39.76

Table 7.23: Variations of performance (left) and percentage of filtered entries (right) associated with different probability cut-off levels.

Table 7.23 shows how the classification performance improves by removing entries with low probability values. When the model is trained on the entire 111,981 vector USPD set, the *weighted* F1-score is 0.88 even without applying any confidence score filtering, which can be already be considered good performance for the classification of an external data set. The performance of the model increases as the probability cut-off

is increased, by sacrificing an increasing percentage of reactions for which predictions are considered as not reliable. The performance improves even for low cut-off values, ranging from 0.15 to 0.35, where the percentage of filtered reactions is under 15%.

- **Conformal Prediction**

The confidence levels associated with true and false predictions, using CP, were evaluated as follows. Scores were first rounded at two decimal positions according to the *half up* method, as described previously. Two separate binning processes were then carried out. The confidence scores were binned into 9 bins ranging from the values 0.92 to 1.00; and the credibility scores were binned into 93 bins ranging from 0.08 to 1.00. The absolute numbers and ratios of true and false predictions associated with each confidence and credibility level are plotted in Figure 7.14:

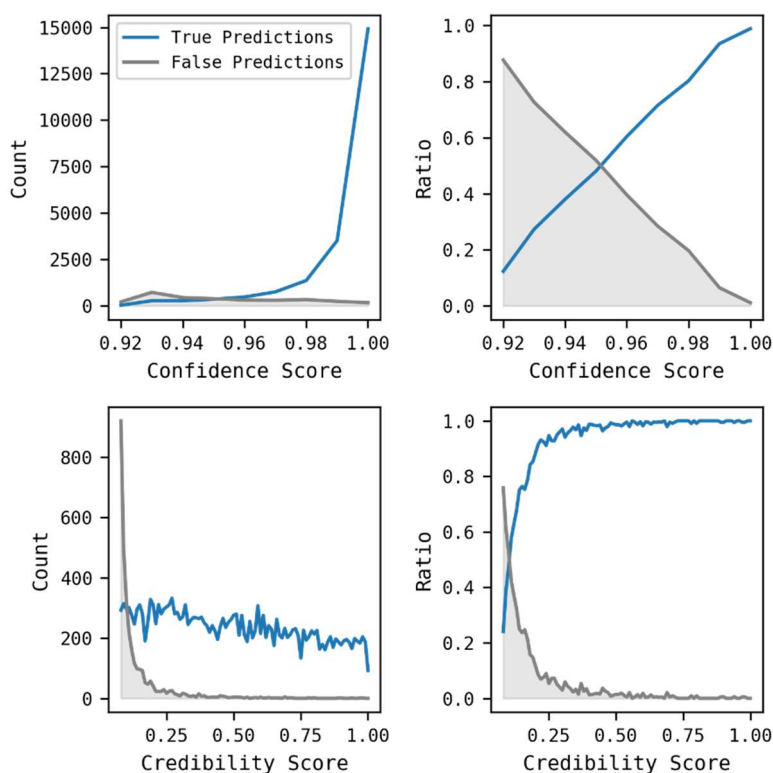


Figure 7.14: Absolute numbers (left) and ratios (right) of true and false predictions associated with each level of confidence (top) and credibility (bottom).

The top-left chart in Figure 7.14 shows a trend similar to that in Figure 7.13, although in this case, the range of scores is significantly smaller. The top-right plot

demonstrates how a satisfactory separation between true and false predictions is achieved only when the confidence score tends to the value 1. These results are supported by the theory of conformal prediction: the highest *p-value* indicates how close the observed prediction is to the typical distribution of results for a given class, but it does not provide information on the presence of other high *p-values* associated with other classes. This effect was verified by plotting the credibility scores which show how far the predicted class is from the rest of the possible class predictions. The bottom-left and the bottom-right plots report a much broader separation between true and false predictions. Both plots show that the percentage of wrong predictions remains constantly very low for a credibility score greater than 0.3.

Estimations were additionally evaluated by selecting a series of arbitrary credibility score cut-offs along with their corresponding *weighted* F1-scores and percentages of filtered entries. Results are reported in Table 7.24:

Credibility Cut-off	<i>Weighted</i> F1-score	Percentage of Filtered Entries
0.00	0.88	0.00
0.09	0.91	4.73
0.10	0.93	8.04
0.12	0.95	12.43
0.15	0.96	17.72
0.20	0.98	24.74
0.25	0.99	36.94

Table 7.24: Variations of performance (left) and percentage of filtered entries (right) associated with different credibility cut-off levels.

Table 7.24 shows the trade-off between F1 score and number of entries filtered out as the credibility cut-off increases. The trends obtained using CP are comparable to those seen using the probability scores in RF, with the performance improving notably even for low cut-off values ranging from 0.09 to 0.12, where the percentage of filtered entries remains under 15%.

The results obtained from the assessment of both confidence estimation methods provide insights on the use of numerical cut-offs to enhance the reliability of the model, for example, by only assigning classes to reactions that have a high chance of being

correctly predicted. It should be noted, however, that these specific values are not directly transferable to other data sets since they will vary according to the composition of the test set. Although the results from both confidence estimation methods are comparable and also dependent on the composition of the test data set as for the RF probability scores, the statistical basis of CP suggests the selection of this method for confidence estimation in the reaction classification model.

7.3.5. KNIME Implementation

The reaction classification using the RF-CP model was implemented as an automated KNIME workflow as shown in Figure 7.15:

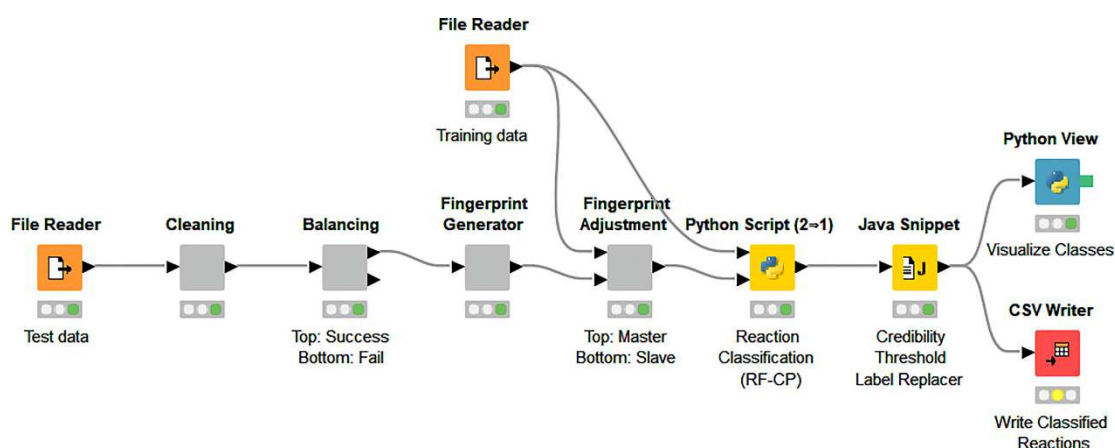


Figure 7.15: RF-CP classification KNIME workflow.

“Cleaning” and “Balancing” nodes correspond to the unmapped compound removal and Reaction Balancing Tool nodes described in Section 5.2, respectively. “Fingerprint Generator” and “Fingerprint Adjustment” are the equivalents of the dynamic reaction fingerprint conversion and adaptation algorithms described in Section 5.3, respectively. The “RF-CP” Python script is the core of the workflow. It encapsulates the entire classification algorithm along with parameters, class weights, and conformal prediction, into a single node that outputs reactions, classes, and scores. On the one hand, the drawback of this approach is that it requires the model to be trained every time the classification is run. On the other hand, a huge amount of memory (~16 GB with the entire USPD set), necessary for the creation of the model, is generated only temporary to produce the classification. Finally, a simple script (“Credibility Threshold Label

Replacer”) is used to replace the labels that fall under a given credibility threshold, with the label “Unclassified” which indicates that those predictions are not reliable. To conclude, two nodes are connected to export the results in table format, or to visualise them in a pie chart.

7.4. Applications

7.4.1. Introduction

The classification of unseen reaction examples is the final goal in the development of a reaction classifier. These instances should always be pre-processed using the same standardisation and encoding protocol used in the preparation of the training data to maximise the effectiveness of the model. Successively, the classification data can be used to generate useful statistics, networks, or new models. In this section, the Evotec ELN and JMC 2008 collections, described in Section 5.5, are classified then analysed to highlight the importance of reaction classification. The selection of these two datasets is justified by the preliminary analysis of their dynamic fingerprints reported in Section 5.5.3, where the comparison between the total number of entries and atom pairs describing each dataset suggests that, ELN and JMC 2008 sets contain low and high varieties of extended reaction centres, respectively. Furthermore, the selection of these datasets is supported by their different sources, specifically industrial and literature data.

7.4.2. Evotec ELN

The Evotec ELN fingerprint dataset described in Table 5.12 was selected for the classification procedure. This dataset is described by 3,305 atom pairs, which correspond to 21% less the number of atom pairs used to fully describe the USPD fingerprint dataset (Table 5.5), although the ELN set contains almost 25% more entries compared to the USPD set. This suggests that the ELN data is less diverse than the USPD data. Consequently, the ELN atom pairs were adjusted to the training data atom pairs as described in Section 5.3. The Random Forest classifier combined with Conformal Prediction (RF-CP) was then used to classify the ELN entries including assigning confidence and credibility scores to the predictions. The distributions of scores are

plotted in Figure 7.16, which reports two distributions with trends similar to those reported in Figure 7.14: confidence scores are comprised within a short range of values (0.924-1.000) and mostly concentrated between 0.98-1.0. This result indicates that the model identified most of the examples as very similar to those used in the calibration set. Credibility scores are included in a larger range of values (0.075-1.000) with an intense peak on the lower bound. This suggests that some examples have high *p-values* for more than one reaction class.

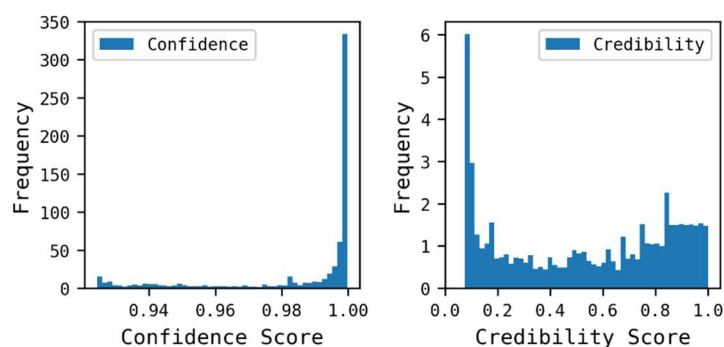


Figure 7.16: Confidence (left) and credibility (right) scores of the Evotec ELN data reaction classification.

Different credibility thresholds were then applied to determine the absolute numbers and percentages of filtered entries at each cut-off level. Results are reported in Table 7.25:

Credibility Threshold	Absolute Number (Percentage) of Retained Entries	Absolute Number (Percentage) of Filtered Entries
0.00	144,008 (100%)	0 (0%)
0.09	129,679 (90.05%)	14,329 (9.95%)
0.10	124,103 (86.18%)	19,905 (13.82%)
0.12	118,754 (82.46%)	25,254 (17.54%)
0.15	114,120 (79.25%)	29,888 (20.75%)
0.20	105,680 (73.38%)	38,328 (26.62%)
0.25	100,569 (69.84%)	43,439 (30.16%)

Table 7.25: Credibility score threshold filtering tests applied on the Evotec ELN data.

A credibility threshold of 0.12 was finally applied to remove the entries with very low chances of being correct predictions; in this case 17.5% of the reactions in the ELN set. This value was chosen based on the results reported in Section 7.3.4.6 where the

same credibility threshold resulted in 12.4% of the entries being removed from the USPDA data while the F1-score for the remaining entries increased to 0.95.

The classification data was analysed at different hierarchical levels: Level-1 (e.g. “C-C Bond Formation”) labels were grouped to produce the pie chart described in Figure 7.17 for comparison with the statistics on the superclasses reported by Schneider and colleagues (2016) in the USPD literature. Level-2 (e.g. “C-C Bond Formation (Coupling)”) and level-4 (e.g. “C-C Bond Formation (Coupling) (Suzuki) (Bromo)”) label statistics are reported in Table 7.26 and Table 7.27, respectively. Level-3 labels were ignored since they produced statistics very similar to the level-4 labels.

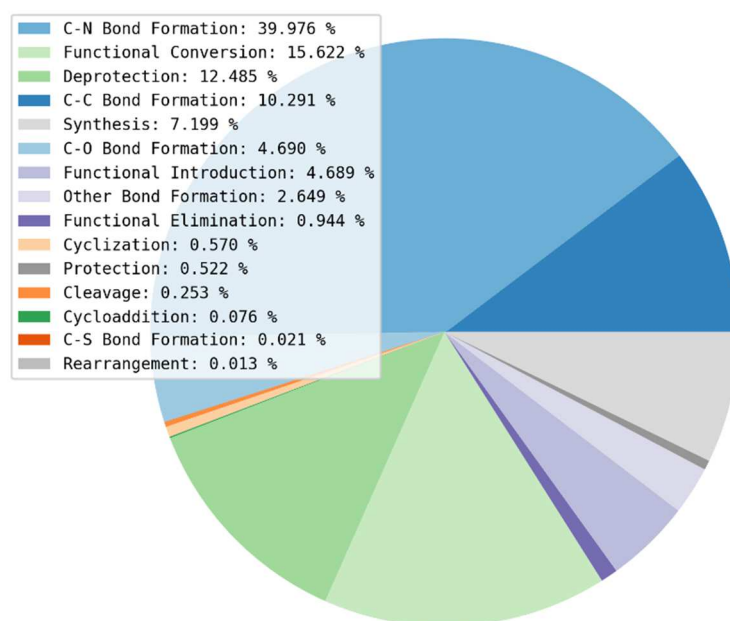


Figure 7.17: Level-1 classification of the Evotec ELN data.

Figure 7.17 provides a general description of the ELN composition: C-N, C-C, and C-O bond formations constitute almost 55% of the total composition of the dataset. This result is in accord with expectations since synthetic strategies in medicinal chemistry are usually bottom-up, hence they usually describe the growth of small fragments into drug-like molecules. Functional conversions describe almost 16% of dataset. This percentage is comparable to the sum of the Reductions, Functional Group Interconversions (FGI), and Oxidations percentages (17.3%) found in the USPD

literature since these classes are all grouped into a single class in the SHREC system. Functional introductions (~4.7%) also describe a result similar to that in the USPD literature (3.4%). The high percentage of functional conversions and introductions can be explained by their use in both molecule construction and optimisation phases. Deprotections (~12.5%) are generally more popular compared to protections (~0.5%), suggesting the use of protected building blocks as starting materials. “Synthesis” (~7.2%) is another frequent class, which describes examples related to the preparation of particular scaffolds such as Thioethers, Imidazoles, Pyrazolamines, Thiazoles, and similar heterocycles. This class resulted to be significantly higher in percentage if compared to the “Heterocycle formation” class (1.6%) described in the USPD literature. This result suggests the use of smaller building blocks and robust reactions for the preparation of bigger scaffolds in alternative to the use of commercially available functionalised blocks. These statistics are also supported by the analysis of the number of reactants in the data set: 63.2% of the entries were described by two reactants (i.e., C-C, C-N, C-O bond formations, and scaffold syntheses), 35.8% by only one reactant (i.e., functional introductions, conversions, and deprotections) and the remaining one per cent of reactions were split between 3, 4, and 5-reactant reactions.

Other classes report lower percentages because of their minor efficacy in the synthesis of compounds of pharmaceutical interest (e.g. “Other Bond Formation”), their unsuitable involvement in molecule construction (e.g. “Cleavage” or “Functional Elimination”), or because of the use of already functionalised reagents that allowed those classes to be skipped (e.g. “Cyclization” and “C-S Bond Formation”).

Table 7.26 describes in more detail which subclasses constitute the superclasses in Figure 7.17: C-N bond formations describe more than a third (6 out of 15) of the most applied subclasses, with strong support by “Condensation” and “N-arylation” which consist of almost 29,000 reaction examples (~24% of the total set). This means that one reaction every four in the dataset is a “C-N Bond Formation (Condensation)” or a “C-N Bond Formation (N-arylation)”. The remaining classes (“C-N Bond Formation (N-alkylation)”, “C-N Bond Formation (Amide formation)”, “C-N Bond Formation

(Amination)”, and “C-N Bond Formation (Carboxylic ester + amine)”) describe additionally almost 16,000 examples confirming that the creation of C-N bonds is a typical strategy in medicinal chemistry due to the robustness and versatility of these reactions in the construction of pharmaceutically relevant structures. It is important to point out that the class “C-N Bond Formation (Amination)” is not considered as a functional introduction in the SHREC because reaction vectors do not encode chemical environments outside the reaction centres, thus reactions involving building blocks containing an amine group are often confused with secondary or tertiary amine group introductions. “C-C Bond Formation (Coupling)” describes more than 10,000 examples indicating the high efficiency of this reaction class as well. A large number of “C-O Bond Formation (Etherification)” examples also indicate the relevance of structures linked as ethers (i.e., R_1-O-R_2 , where R is a hydrocarbon group) as an alternative to the “C-N” and “C-C” bond formations. Deprotections are dominated by three subclasses: t-Butyloxycarbonyl (BOC), methyl, and ethyl protective groups are used in more than 11,000 examples. Such a high number suggests the use of protected building blocks to enforce selective reactivity or to avoid catalyst poisoning as suggested by Schneider and colleagues (2016).

Level-2 Classification	Count
C-N Bond Formation (Condensation)	15,995
C-N Bond Formation (N-arylation)	12,667
C-C Bond Formation (Coupling)	10,198
Deprotection (N-t-Butyloxycarbonyl)	6,293
C-N Bond Formation (N-alkylation)	6,024
C-O Bond Formation (Etherification)	4,401
C-N Bond Formation (Amide formation)	4,013
C-N Bond Formation (Amination)	3,947
Functional Conversion (Reduction)	3,276
Other Bond Formation (Sulfonamide formation)	3,106
Deprotection (COO-Methyl)	2,984
Functional Introduction (Bromination)	2,359
Functional Conversion (Nitro to amino)	2,133
C-N Bond Formation (Carboxylic ester + amine)	1,985
Deprotection (COO-Ethyl)	1,796

Table 7.26: Top 15 reaction classes in the Evotec ELN data according to the level-2 labelling system.

Although the “Other Bond Formation” class is not included among the majority classes in the level-1 classification, the specific “Other Bond Formation (Sulfonamide formation)” class is represented by more than 3,100 examples of reactions, indicating its particular efficacy in the creation of S-N bonds between amines and sulphones. Despite its general popularity in the level-1 classification, the “Functional Conversion” superclass describes only one subclass in Table 7.26, which is “Functional Conversion (Nitro to amino)” with approximately 2,100 examples. This result suggests the presence of many different functional conversions contributing to the superclass statistics with no preferred subclasses. The opposite effect is seen for the “Functional Introduction” superclass, which is not very frequent compared to the other level-1 classes even though the “Functional Introduction (Bromination)” subclass is represented by more than 2300 examples in Table 7.26.

Level-4 Classification	Count
C-N Bond Formation (Condensation) (Carboxylic acid + amine)	14,211
C-N Bond Formation (N-arylation) (Chloro)	8,220
Deprotection (N-t-Butyloxycarbonyl) (N-Boc)	6,293
C-C Bond Formation (Coupling) (Suzuki) (Bromo)	4,820
C-N Bond Formation (Amide formation) (Schotten-Baumann)	3,874
C-N Bond Formation (N-alkylation) (Bromo)	3,229
Other Bond Formation (Sulfonamide formation) (Schotten-Baumann)	3,106
Deprotection (COO-Methyl) (COO-Me)	2,984
C-O Bond Formation (Etherification) (Williamson)	2,937
C-N Bond Formation (N-arylation) (Bromo)	2,429
Functional Introduction (Bromination)	2,359
Functional Conversion (Nitro to amino)	2,133
C-N Bond Formation (Carboxylic ester + amine)	1,985
C-N Bond Formation (N-alkylation) (Chloro)	1,828
Deprotection (COO-Ethyl) (COO-Et)	1,796

Table 7.27: Top 15 reaction classes in the Evotec ELN data according to the level-4 labelling system.

Table 7.27 preserves almost the same order compared to Table 7.26. On the one hand, some classes, such as “C-C Bond Formation (Coupling)” or “C-N Bond Formation (N-alkylation)”, lose their positions as a result of their further splitting into smaller subclasses (e.g. “C-N Bond Formation (Amination)” was split into four subclasses of which none is described in Table 7.27). On the other hand, classes such as “C-N Bond

Formation (Amide formation)” maintain their position in the ranking after adding further information to their labels. More specifically, “C-N Bond Formation (N-arylation)” is split into “C-N Bond Formation (N-arylation) (Bromo)” and “C-N Bond Formation (N-arylation) (Chloro)”, which preserve good positions in Table 7.27; “C-N Bond Formation (N-alkylation)” describe a similar result yielding “C-N Bond Formation (N-alkylation) (Bromo)” and “C-N Bond Formation (N-alkylation) (Chloro)”.

The addition of extra information levels did not affect several class counts at all due to two reasons: first, some classes such as “Functional Conversion (Nitro to amino)” or “Functional Introduction (Bromination)” are not further discriminated passing from the level-2 to level-4, so they preserve the same labels and counts, and second, the “Other Bond Formation (Sulfonamide formation)” which is transformed into “Other Bond Formation (Sulfonamide formation) (Schotten-Baumann)” still preserve the same count since it is the only one sulfonamide formation class in the dataset.

Time series were also produced using the classification data then analysed. These plots can be particularly useful if focused, for example, on the correlation between classes and financial (e.g. company profits) or scientific (e.g. successful properties in compounds) parameters. In particular, this type of analysis can be used to remove the bias on certain reaction classes, and identify those that are more effective. However, only a correlation study between classes is reported in this work due to the lack of accessibility on the company data. The results are not supposed to be exhaustive, rather they are intended to provide some hints on how reaction classification can bring useful information for decision making in drug discovery. Level-1 classification labels were selected due to their more generalised nature and the lower number of classes. Values (i.e., counts or yields) were split by year, then retained only for the years between 2010 and 2017 (years 2008 and 2018 were excluded due to their partial contents). A total of 115,778 reactions were retained. Class counts were also normalised by total counts per year. Counts are reported in Table 7.28, which describes a steady increase in the total number of entries since the introduction of the corporate ELN. The growth reaches a peak in 2014 then gradually drops by the end of 2017. This behaviour can be explained by the introduction of client

ELNs which are private notebooks that cannot be accessed internally. The use of private databases could have affected the composition of the classes as well.

2010	2011	2012	2013	2014	2015	2016	2017
7,082	9,760	14,695	16,075	20,407	19,879	15,839	12,041

Table 7.28: Reaction counts per year in the Evotec ELN.

Time series plots of absolute and normalised counts, are reported in Figure 7.18 and Figure 7.19, respectively. The first chart takes into account the absolute amounts of reactions carried out per year, to provide a global perspective on the creation of the dataset, while the second one scales the count information to enable a better comparison across years.

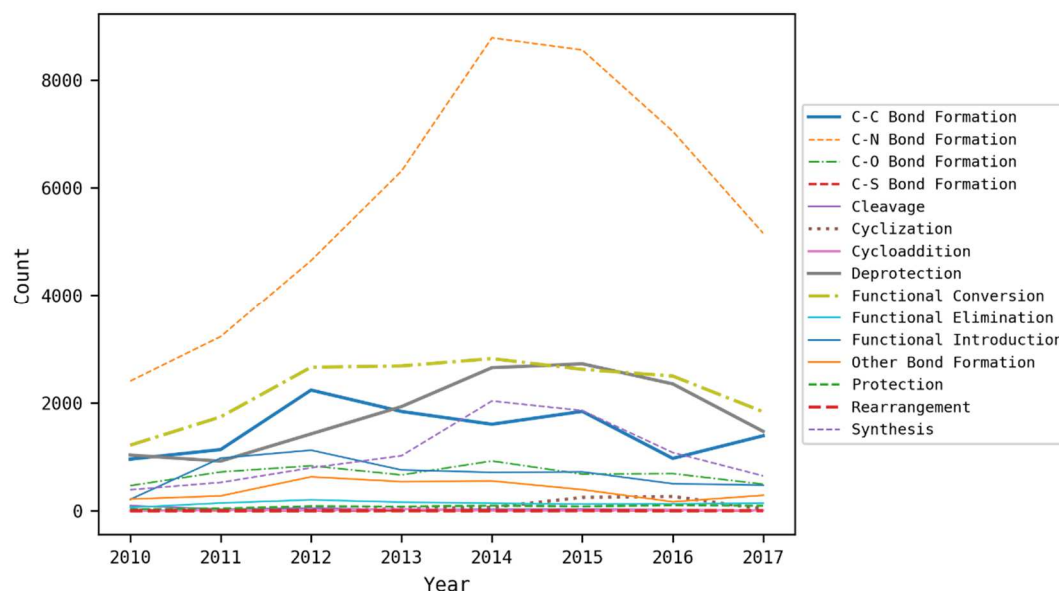


Figure 7.18: Absolute count time series of the Evotec ELN level-1 classes.

In both figures, the overwhelming presence of “C-N Bond Formation” compresses the other classes, although many of their trends are still clearly visible. Figure 7.18 describes an increasing trend for almost every class with a peak in 2014-2015, followed by a rapid decrease. Exceptions are “C-C Bond Formation”, “Functional Introduction”, and “Other Bond Formation” which are characterised by earlier peaks (i.e., 2011-2012), and increasing trends in 2017. Figure 7.19 provides a different perspective of the same scenario: “C-N Bond Formation”, “Deprotection”, and “Synthesis” report an increase

passing from early (2010 to 2012) to late years (2014 to 2016 excluding 2017). This general growth is obtained at the expense of the other classes such as “C-C Bond Formation”, “C-O Bond Formation”, or “Other Bond Formation”, which regain some positions only in 2017. As already reported in the literature (Brown and Boström, 2016) (Boström *et al.*, 2018) (Campbell, Macdonald and Procopiou, 2018), this result indicates a higher propensity towards the use of C-N bond formations due to their simplicity and robustness.

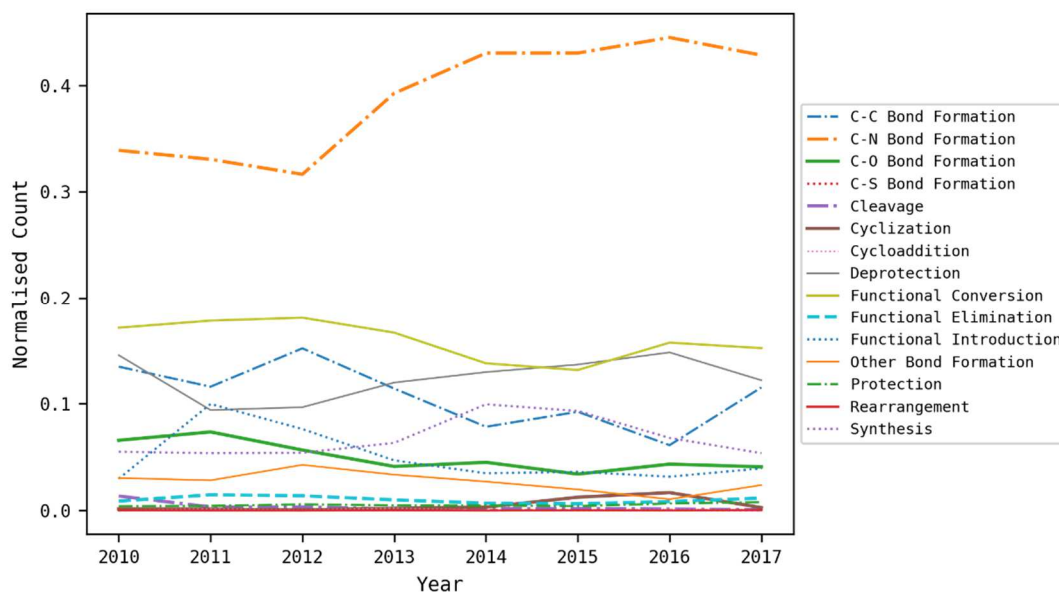


Figure 7.19: Normalised count time series of the Evotec ELN level-1 classes.

The correlation between normalised class counts was also inspected by calculating the Pearson correlation coefficient (R) for each class pair in the dataset, then represented as a heatmap in Figure 7.20, which can be analysed by selecting some reference classes such as bond formations, then by inspecting their combinations with the other classes: “C-C Bond Formation”, “C-O Bond Formation”, and “Other Bond Formation” show positive correlations with “Cleavage” and all of the functional-related class, whereas, they are negatively correlated with “C-N Bond Formation”, “Cyclization”, “Deprotection”, and “Synthesis”. Conversely, “C-N Bond Formation” shows opposite trends, suggesting that the substrates involved in these reactions do not need to be pre-functionalised *in-situ* (i.e., functional introduction or conversion) to react with each

other. This can result in a decrease in the number of steps required to obtain the final products, thus explaining the growing success of this class over time. This hypothesis could be further tested by comparing the average number of steps in routes containing and non-containing “C-N Bond Formation”. Furthermore, “C-N Bond Formation” and “Deprotection” show a positive correlation with each other, suggesting the deprotection of the products after the union of two building blocks through the formation of a C-N bond. “Synthesis” shows a positive correlation with “Deprotection” probably for the same reason. The negative correlation between “Functional Elimination” and “Deprotection” can be rationalised by considering that in general, both classes involve the elimination of functional groups, thus it would be unlikely to observe an increasing occurrence of these two classes at the same time. “C-S Bond Formation”, “Cycloaddition”, “Protection”, and “Rearrangement” do not show relevant relationships with the other reaction classes. This can be a consequence of their lower popularity in the dataset.

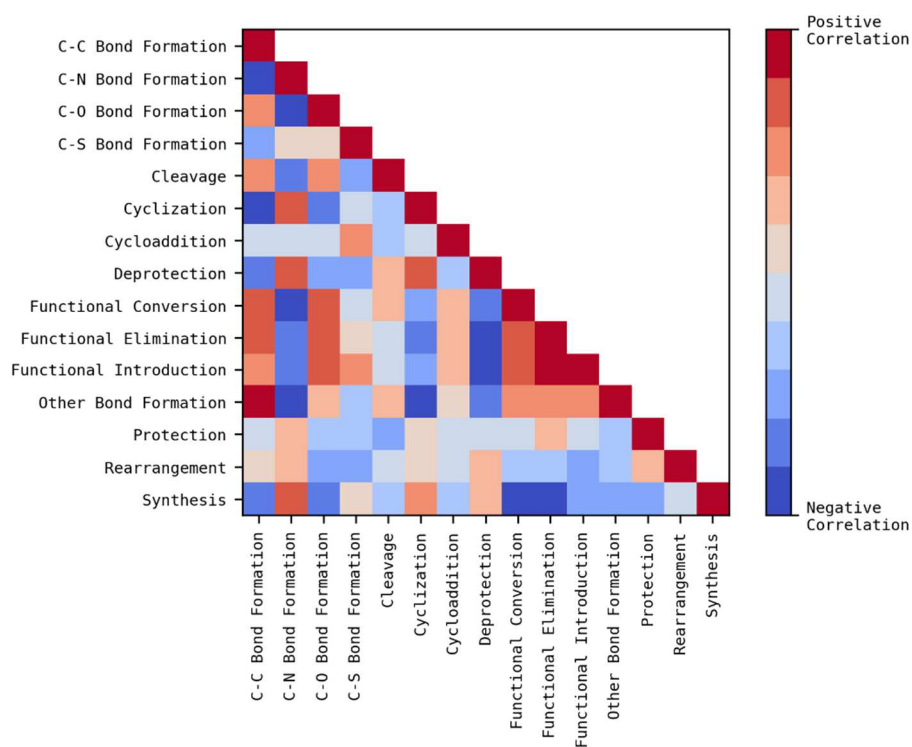


Figure 7.20: Heatmap that describes the lower triangular pairwise matrix of the Evotec ELN level-1 class correlation coefficients.

Yield time series plots were also analysed by reaction class. Multiple yields associated with the same entry were averaged and reactions for which no yield was reported were filtered out. In addition, classes described by less than 250 entries in the years between 2010 and 2017 were not analysed due to their large variance. A total number of 83,343 entries were retained. The yield time series is reported in Figure 7.21, which describes three different trends: increasing, decreasing, and stable yields. “Deprotection” and “C-C Bond Formation” describe increasing trends, while “Functional Elimination” and “Functional Introduction” decrease over time. The remaining classes report stable trends characterised by either low variance (i.e., “Functional Conversion”, “Synthesis”, and “C-N Bond Formation”) or high variance (i.e., “Cyclization”, “Other Bond Formation”, “C-O Bond Formation”). This typology of analysis could be readily implemented in the ELN framework to monitor how each different class perform over time with the aim of maintaining high global efficiency. For example, this can be used to assess the performance of the medicinal chemists in a specific time range, or to highlight differences in yield due to the impurity of reagents, after the introduction of a new chemical supplier.

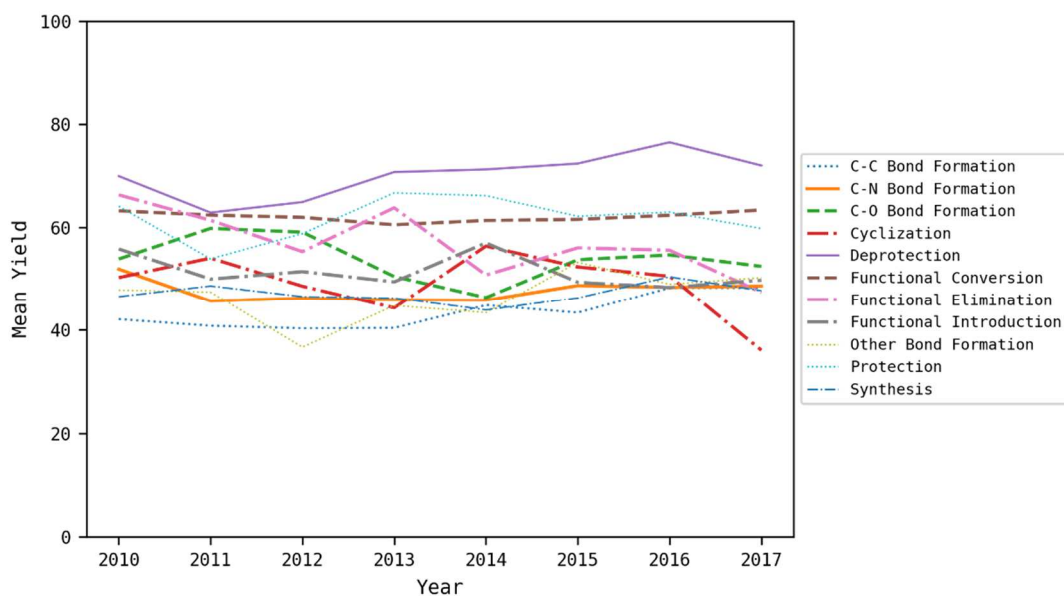


Figure 7.21: Yield time series of the Evotec ELN.

7.4.3. JMC 2008

The JMC 2008 fingerprint dataset described in Table 5.12 was selected for the classification procedure. This dataset is described by 5,331 atom pairs, which already suggest that a large variety of reaction centres are contained within it. Although the JMC 2008 set is represented by 90% fewer unique reaction vectors compared to the original USPD fingerprint dataset (Table 5.5), it requires almost 27% more atom pairs to be fully described. This preliminary result suggests that the data is more diverse than the patent data, which is not surprising given that the patent literature is aimed at capturing local regions of chemical space whereas the medicinal chemistry literature is more likely to consist of a greater variety of syntheses with no necessary pre-requirement for robustness or coverage of particular subspaces. Consequently, the JMC atom pairs were adjusted to the training data atom pairs as described in Section 5.3.

The RF-CP classifier was used to classify the JMC 2008 entries including assigning confidence and credibility scores to the predictions. The distributions of scores are plotted in Figure 7.22, which describes a narrow range of confidence values (0.924-1.000), similarly to the results found for the ELN data, although the JMC values are more spread. This indicates that the classifier still identified the majority of the JMC reactions as very similar to the reactions contained in the calibration set, although they presented lower similarities compared to the ELN reactions. The JMC credibility scores show a range of values identical to that found for the ELN data (0.075-1.000); however, the majority of the reactions are associated with lower scores. This means that the JMC data generally consists of examples with higher ambiguity compared to the ELN distribution reported in Figure 7.16, causing a decrease in distance between the first and second best *p-values* computed by the CP.

Different credibility thresholds were then applied to determine the absolute numbers and percentages of filtered entries at each cut-off level as reported in Table 7.25. Results are reported in Table 7.29.

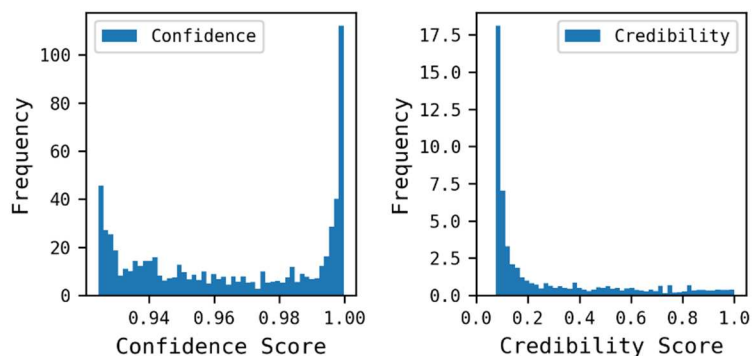


Figure 7.22: Confidence (left) and credibility (right) scores of the JMC dataset reaction classification.

Credibility Threshold	Absolute Number (Percentage) of Retained Entries	Absolute Number (Percentage) of Filtered Entries
0	19,209 (100%)	0 (0%)
0.09	13,335 (69.42%)	5,874 (30.58%)
0.10	11,632 (60.55%)	7,577 (39.45%)
0.12	9,779 (50.91%)	9,430 (49.09%)
0.15	8,339 (43.41%)	10,870 (56.59%)
0.20	6,994 (36.41%)	12,215 (63.59%)
0.25	6,308 (32.84%)	12,901 (67.16%)

Table 7.29: Credibility score threshold filtering levels applied on the JMC dataset.

The application of a threshold of 0.12 results in 49.09% of the JMC 2008 reactions being filtered out, compared to only 17.54% of the ELN reactions. A manual inspection of the filtered entries confirmed that most were not classified correctly. Two conclusions were drawn from these results. First, data from scientific literature tends to be more difficult to classify for the model due to its higher diversity in terms of extended reaction centres. Second, the use of the credibility score thresholds in a more difficult classification problem highlights the practical advantages of integrating the classification model within a CP framework to improve model reliability. In particular, the comparison between the ELN and JMC distributions provides evidence on the augmented nature of the classification tool: the model becomes aware of its own limits through the use of the CP. This characteristic can also be used to determine when more training data is required. The 9,779 reactions (50.9%) retained at the 0.12 credibility level were analysed as reported in Section 7.4.2 for the ELN data. Results are reported in Figure 7.23, Table 7.30, and Table 7.31.

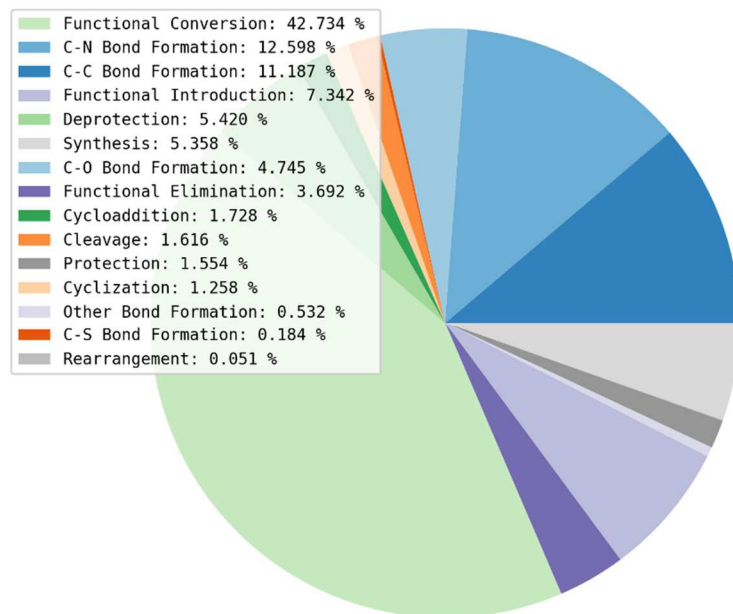


Figure 7.23: Level-1 classification of the JMC 2008 dataset.

Figure 7.23 shows different trends in comparison to Figure 7.17: “Functional Conversion” dominates all the other classes describing almost the 43% of the entire classification, compared to 15.4% of the ELN data. This suggests that these reactions were focused on scaffold modifications more than C-N, C-C, and C-O bond formations which constitute 28.5% of the total classification. The analysis of the number of reactants in the dataset supported this finding: 65.7% of entries were described by only one reagent and the remaining 34.3% by two reagents. “Functional Introduction” (7.3%) and “Synthesis” (5.4%) also describe a significant number of examples in the data, indicating their persistent roles in medicinal chemistry. Deprotections constitute only 5.4% of the total classification in comparison to 12.5% reported for the Evotec ELN, substantiating the existence of a positive correlation between C-N bond formations and deprotections.

The higher percentages of the minority classes such as “Functional Elimination” (3.7%) and “Cleavage” (1.6%) supports the fact that these reactions are generally avoided in industrial pharmaceutical chemistry where the objective is to construct the final products in the attempt of maximizing the atom economy. Conversely, the academic literature is usually more concerned with the presentation of new scaffolds

with particular properties, with limited regard for the number of steps used to obtain such molecules. The “Cycloaddition” (1.7%) (22.7 times higher) and “Cyclization” (1.3%) (2.2 times higher) classes are also more prevalent compared to the analysis on the ELN data.

Level-2 Classification	Count
Functional Conversion (Hydrogenation)	1,034
Functional Conversion (Reduction)	776
C-C Bond Formation (Coupling)	466
Functional Conversion (Alkene to epoxide)	307
Functional Conversion (Cyano to carboxy)	293
Functional Conversion (Oxidation)	293
Synthesis (Thioether)	293
Functional Conversion (Nitro to amino)	277
C-N Bond Formation (Condensation)	250
Functional Introduction (Hydroxylation)	244
Functional Conversion (Alcohol to alkene)	227
C-O Bond Formation (Esterification)	225
C-O Bond Formation (Etherification)	218
C-N Bond Formation (N-alkylation)	203
Functional Introduction (Bromination)	164

Table 7.30: Top 15 reaction classes in the JMC 2008 according to the level-2 labelling system.

Table 7.30 describes more specifically what classes contributed to each superclass in Figure 7.23: Functional conversions occupy seven of 15 positions in the ranking with “Hydrogenation”, “Reduction”, “Alkene to epoxide”, “Cyano to carboxy”, and “Oxidation” describing more than 2,700 examples which correspond to the 28% of the total composition of the dataset. These reactions tend to preserve almost the total number of heavy atoms in a given structure, thus they can only be used for structural activation of functionalization. Furthermore, the popularity of a particular scaffold synthesis class which is the “Synthesis (Thioether)” shows that the chemists focused on a particular motif which can be a typical in datasets covering short ranges of time. This is also supported by the presence of “Functional Conversion (Alkene to epoxide)” as the fourth most frequent class in the ranking. This class indicates a particular interest in the transformation of alkenes into their corresponding epoxides, which is not a common transformation observed in the preparation of molecules of pharmaceutical relevance.

Most frequent bond formations refer to ordinary subclasses such as “C-C Bond Formation (Coupling)”, “C-N Bond Formation (Condensation) or (N-alkylation)”, or “C-O Bond Formation (Esterification) and (Etherification)”. It is also worth noting that the C-C bond formation class reported almost twice examples compared to the most popular C-N bond formation class. This result is consistent with the analysis carried out by Schneider and colleagues (2016) where they highlighted increasing attention towards C-C bond formations in recent years.

Level-4 Classification	Count
Functional Conversion (Hydrogenation) (Alkene to alkane)	909
Functional Conversion (Reduction) (Aldehyde/ketone to alcohol)	528
Functional Conversion (Alkene to epoxide) (Prilezhaev)	307
Functional Conversion (Cyano to carboxy)	293
Synthesis (Thioether)	293
Functional Conversion (Nitro to amino)	277
Functional Conversion (Alcohol to alkene)	227
Functional Conversion (Oxidation) (Alcohol to aldehyde/ketone)	208
Functional Introduction (Hydroxylation) (Alkene hydration)	205
Functional Introduction (Bromination)	164
Cycloaddition (Diene + dienophile) (Diels-Alder)	160
C-O Bond Formation (Esterification)	155
Functional Elimination (Deoxygenation)	155
Functional Conversion (Sulfanyl to sulfinyl)	147
C-N Bond Formation (Condensation) (Carboxylic acid + amine)	141

Table 7.31: Top 15 reaction classes in the JMC 2008 according to the level-4 labelling system.

Table 7.31 almost preserves the same order reported in Table 7.30 except for a few classes: “C-C Bond Formation (Coupling)” and “C-N Bond Formation (N-alkylation)” are split into multiple classes, among which no one results to be sufficiently populated to appear on the top 15 classes. However, “Cycloaddition (Diene + dienophile) (Diels-Alder)”, “Functional Conversion (Sulfanyl to sulfinyl)”, and “Functional Elimination (Deoxygenation)” appear in the level-4 top 15 positions highlighting that the JMC data set composition is more related to particular transformations which are perhaps aimed at producing novel scaffolds. The presence of specific functional conversions and, in particular, of a functional elimination class among the top 15 classes describes a trend

diametrically opposed to the statistics found for the Evotec ELN data set and the US patent reactions.

7.5. Conclusions

In this chapter, the optimisation of the components for the development of an effective reaction classification model applicable on real collections of unclassified data has been described. A prototype of the model was first validated by reproducing conditions similar to those described in the work by Schneider and colleagues (2015), with special attention on using validation sets that did not contain any vectors used in the training phase to enable a more accurate evaluation of the model. The model's capabilities have been then scaled up to classify a much higher number of reaction classes, specifically from 50 to 336 classes. The extended model was accurately tuned and validated, then finally combined with a module for confidence estimation. The model was finally tested on two noisy datasets obtained from distinct sources of reactions to highlight the potentials and limitations of the approach. The next chapter describes the use of classification data for the development of a reaction class recommendation model that can be used to further enhance reaction-based *de novo* design.

Chapter 8: Reaction Class Recommendation

8.1. Introduction

Reaction vector-based *de novo* design is aimed at accounting for the synthetic accessibility of generated products by using structural transformations derived from known reactions to drive compound generation. However, reaction vectors only account for the structural changes that occur at the core of transformations, hence they do not consider the presence of external functionalities that can compromise the reaction outcome. Machine learning can be used to address this issue by exploiting data on known reactions to identify which reaction classes should be applied to starting materials according to their characteristics (e.g. molecular fingerprints). Consequently, the suggested classes can be used by the structure generator to limit the application of reaction vectors only to those classified as in the suggestions.

In this chapter, a machine learning model for reaction class recommendation is developed using the US pharmaceutical patent data processed in Section 5.4, as a source of starting materials labelled by reaction class. The model is constructed to be compatible with the reaction classification model described in Chapter 6. A systematic approach for the identification of the best model configuration and its scaling-up is undertaken. The reaction class recommender is then evaluated in a number of *de novo* design scenarios.

The chapter is organised as follows. Section 8.2 describes the basic principles of reaction class recommendation, also motivating its introduction and objectives in *de novo* design. Section 8.3 describes the evaluation of a broad selection of multi-label classification components, including molecular descriptors, label types, multi-label approaches and classifiers, assessed on a small amount data in order to rationalise the behaviour of the recommender and provide some insights for the selection of more promising configurations. Section 8.4 focuses on a narrower selection of components as well as introducing more sophisticated multi-label approaches to overcome the issues

related to the use of a much bigger training set. Configurations are investigated in detail with the aim of maximising both model effectiveness and memory efficiency. Section 8.5 aims at quantifying the model effectiveness in real applications, in particular for *de novo* design purposes. First, the model is used to produce suggestions to a set of starting materials for which the reaction classes had been determined. Second, the model is integrated into the reaction vector-based design framework to verify its effects on the enumeration of a compound library. Third, the model is integrated into the RENATE algorithm and used to repeat the validation described in Section 6.3 to quantify the effect of the recommender in a more realistic design context.

8.2. Theoretical Basis

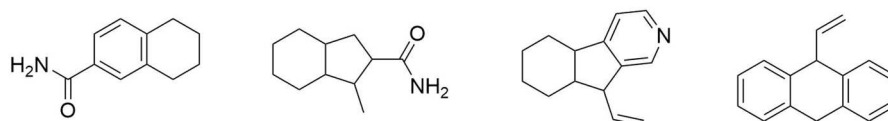
Molecular features can be linked to reactivity in a similar way to linking features to other properties such as activity or toxicity in more traditional drug discovery classification tasks. Typically, classification algorithms are configured to work through the association of a set of features (e.g. functional groups) with only one output label. In contrast, reaction class recommendation is configured as a multi-label classification problem, where the aim of classifiers is to identify which reaction classes are more likely to be applied to a given molecule by accounting for its entirety rather than considering only the portions of structure (i.e., reaction centres) that are involved in different reactions. Hence, the model should be able to detect the presence of functionalities that can reduce the reactivity of molecules or compete when certain reagents are present. For example, molecules containing a single amine (NH) group or a single hydroxyl (OH) group can potentially undergo several types of couplings and condensations with no issues. However, the presence of both functionalities in the same molecule would require a chemist to protect one of the two groups before proceeding with a reaction that could otherwise occur at multiple reaction centres. The reaction vectors alone are agnostic of the competing nature of the functional groups and may, therefore, result in virtual products that are unlikely in reality. The introduction of the recommender in the design framework aims to identify reaction classes that are more suitable to be applied, so that, these can be used by the structure generator to filter the reaction vectors in the database.

Reaction class recommendation is more appealing than specific reaction recommendation due to its versatility and low computational requirements. This is because, reaction class recommenders can be adapted to work with any set of reactions that contains compatible classification data, and they can be trained using smaller amounts of data compared to the potential number of specific reaction examples that would be necessary to train a reaction recommender. Furthermore, reaction class recommendation is expected to be less biased than reaction recommendation since it involves the application of groups of transformations rather than specific reaction centres, thus preserving the chance of exploring novel chemical space.

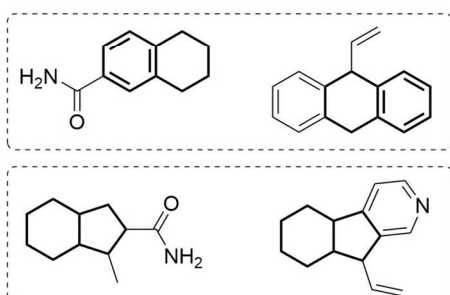
The information required to train the recommender is a set of molecular representations (e.g. functional group fingerprints) associated with multiple reaction classes. Such information is not directly accessible from public data, yet it can be mined from classified reaction examples by grouping together molecules that present the same features, as they will be expected to share a similar reactivity. This assumption is an application *a priori* of the *chemical similarity principle* by Johnson and Maggiora (Gerald M. Maggiora, 1992) to create the data that is necessary to train the reaction class recommender, and its validity in this context is strongly determined by the selection of appropriate molecular features for the grouping. Examples of grouping by reactivity using correct and incorrect features are reported in Figure 8.1, where the first example shows that the selection of inappropriate features would result in the grouping of structures which do not really share similar reactivity properties. The second example shows that the use of descriptors capable of encoding the typical functionalities involved in reactions would increase the chance of grouping the structures effectively. The selection of appropriate features for grouping different molecules into single representations is not the only condition to account for. The number of features should also be balanced to meet a compromise between generality and specificity. On the one hand, a low number of features will tend to generalise the entries, thus producing a smaller number of molecular descriptions associated with more classes. On the other hand, a high number of features will tend to discriminate more the entries, thus

generating a higher number of descriptions associated with fewer classes. The use of binary, integer, or decimal values is expected to produce different effects as well. Reaction classes are another factor to consider. On the one hand, the use of very generic classes would reduce the dimensionality of the problem but it would not necessarily produce a useful tool. For example, level-1 labels, which are obtained from the decomposition of level-4 labels (see Section 5.4.6), would include many subclasses within a single super-class, thus producing suggestions that are too general. On the other hand, the use of more specific labels would restrict the suggestions toward particular classes, thus excluding similar transformations that could be still desirable. For example, the level-4 class recommendation “C-C Bond Formation (Coupling) (Suzuki) (Bromo)” would exclude the Chloro and Iodo subclasses, although they represent analogue transformations with slightly different reagents.

Set of molecules for grouping by reactivity:



Incorrect description and grouping:



Correct description and grouping:

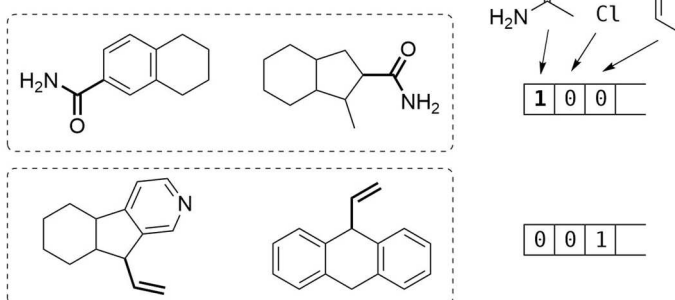


Figure 8.1: Examples of correct and incorrect grouping by molecular features.

The combination of multiple entries to form a single row associated with multiple class columns is generally indicated as *pivoting*. An example of feature encoding and label *pivoting* for the preparation of multi-label data for the class recommender is reported in Figure 8.2:

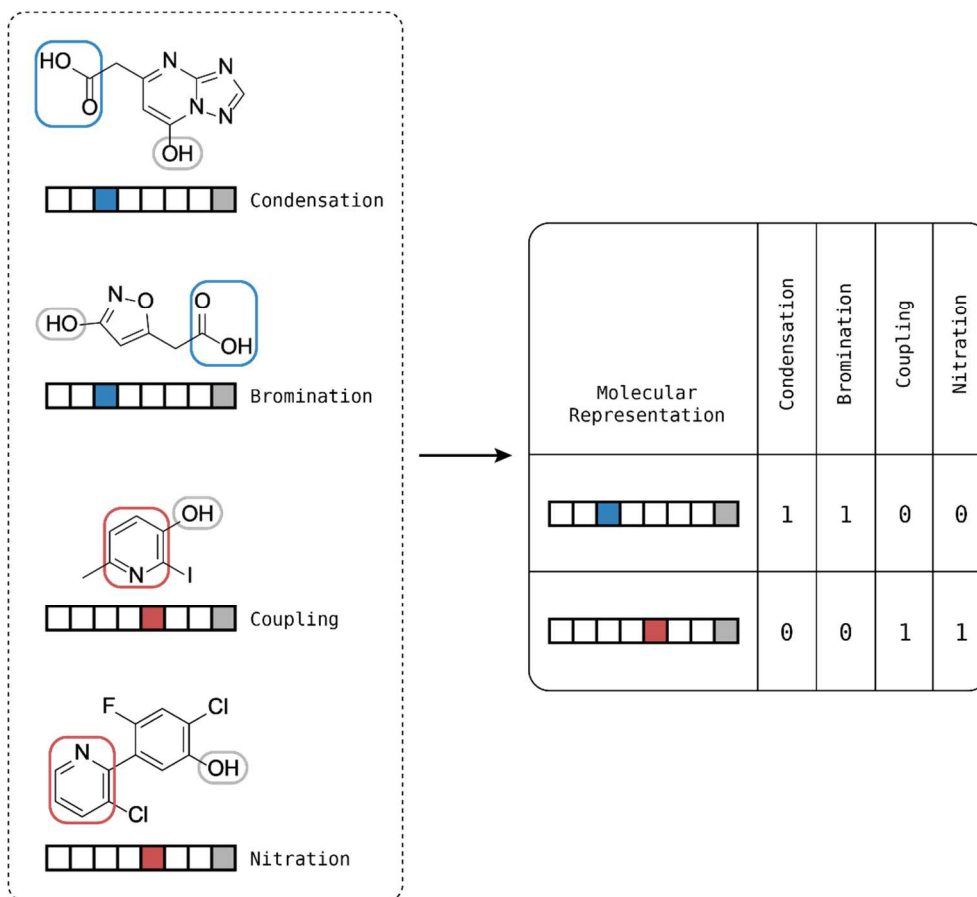


Figure 8.2: Example of feature encoding and label *pivoting*.

Another requirement to satisfy in the construction of an effective recommender is the compatibility between the data used to train and query the model. More specifically, if the recommender is eventually applied for *de novo* design purposes, it has to be trained using examples that represent what occurs at each step of the growing process in order to produce useful suggestions. In reaction vector-based *de novo* design the key molecule is the starting material, which is transformed or expanded by means of a set of reagents. In this context, a potential approach for the preparation of the training data, would first consist of decomposing a set of reaction examples into starting materials and reaction classes. For a given reaction, the extraction of the starting material can be done by

retaining the reactant that has the highest number of mapped atoms since it corresponds to the structure that is majorly preserved in the products. An example of starting material extraction is described in Figure 8.3. Alternatively, if the reactions are balanced in terms of heavy atom counts, the starting material will also correspond to the reactant with the highest number of heavy atoms.

Reaction example for starting material extraction:

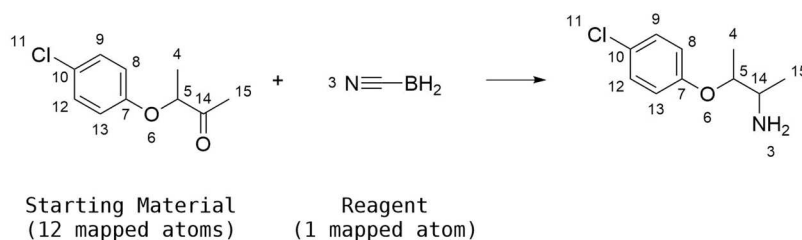


Figure 8.3: Example of mapping-based starting material extraction.

8.3. Proof of Concept Model

8.3.1. Introduction

This section presents a systematic investigation of a large distribution of molecular descriptors, label types, multi-label approaches, and validation methods, which are explored for the construction of a proof of concept (PoC) model, in order to provide information for the design of the final recommender.

8.3.2. Molecular Descriptors

A variety of different types of fingerprints were selected as descriptors for this study. Binary fingerprints were preferred over other methods due to their higher tendency toward data generalisation, with the aim of increasing the chance of grouping more entries into single entities. However, a pharmacophore-based count fingerprint was also assessed to compare its performance against its binary version. Chi and Kappa descriptors were also evaluated although they were expected to fail for three reasons: they encode information on the topology of the molecules, hence are not related to functional groups or reactivity; they encode continuous values, thus they are not expected to produce an effective grouping of different molecules in the dataset; they

encode only 13 features, while multi-label classification guidelines suggest using a number of features greater than the number of labels (i.e., reaction classes) (Read, 2010). The selected molecular descriptors are listed in Table 8.1. In addition, different configurations were also defined for some descriptors. For example, Avalon fingerprints were configured to have six lengths (256, 640, 1024, 2048, 4096, 8192), and FeatMorgan fingerprints were configured to have three different radius levels (1, 2, 3). A total number of 22 fingerprint configurations were selected for the experiment (Table 8.4).

Molecular Descriptor	Type	Description
Atom-pair	Binary	Hashed fingerprints implemented in the RDKit library which are inspired by atom-pair descriptors (Carhart <i>et al.</i> , 1985).
Avalon	Binary	Hashed fingerprints mainly describing atom presences, paths, bonds, rings, and hydrogen features implemented in the RDKit library (Gedeck, Bernhard and Bartels, 2006).
CDK Functional Group	Binary	SMARTS-based dictionary fingerprint developed by Inte:Ligand and implemented using RDKit, which encodes the presence of 307 different functional groups (Laggner, 2005).
ChemAxon Functional Group	Binary	SMARTS-based dictionary fingerprint implemented using RDKit, that uses a set of patterns predefined in Marvin software (ChemAxon Ltd., 2015). It encodes the presence of 110 different groups.
Chi and Kappa Descriptors	Decimal	Shape indices implemented in the RDKit library (Hall and Kier, 1991).
Dompé	Binary	Substructure-based fingerprint implemented in MOE software, that encodes the presence of 3047 patterns taken from lead molecules.
FeatMorgan	Binary Count	Pharmacophore-based fingerprint implemented in the RDKit library, that also encodes inter-distances between features and neighbour information within a defined radius (Gobbi and Poppinger, 1998).
Layered	Binary	Subgraph-hashed fingerprint implemented in the RDKit library, that encodes several layers of information such as topology, bond order, atom types, rings, ring sizes, and aromaticity.
MACCS	Binary	SMARTS-based implementation in the RDKit library of 166 public MACCS keys by MDL Information Systems.
Morgan	Binary	Extended-connectivity fingerprints implemented in the RDKit library, which encode chemical features and neighbour information within a defined radius (Rogers and Hahn, 2010).
OChem EFG+	Binary	Integrated version of the OChem EFG fingerprint implemented using RDKit, which encodes the presence of 2080 structural features (Salmina, Haider and Tetko, 2015).
Pattern	Binary	Experimental topological fingerprint implemented in the RDKit library, which uses a set of predefined generic substructure patterns (Landrum, 2016).
RDKit	Binary	Daylight-like hashed fingerprint based on molecular subgraphs (Landrum, 2016).
Torsion	Binary	Hashed fingerprints implemented in the RDKit library which are inspired by topological torsion descriptors (Nilakantan <i>et al.</i> , 1987).

Table 8.1: Selection of molecular descriptions investigated for the PoC model.

8.3.3. Data Selection

The 111,981 reaction USPD dataset described in Table 7.6, was selected for this proof of concept. Note that this set of reactions consists of a wide variety of reaction centres since the reactions have been screened to remove any duplicates based on their reaction vector representations. 50 reaction classes (at level-4) with the number of examples greater than or equal to 300 were randomly selected, then the data was pre-processed to yield a subset of starting materials and reaction classes as follows. Reactions were mapped using the Indigo Reaction Automapper node and then starting materials were extracted by retaining the reactants associated with the highest number of mapped atoms, along with their reaction classes as described in Section 8.2. Entries containing multiple reactants with the same highest number of mapped atoms were filtered out. The resulting subset is characterised by an imbalanced distribution of starting materials per class. The set is described in Table 8.2:

Dataset	Number of Starting Materials	Number of Classes	Median Number of SMs per Class
PoC USPD subset	44,222	50	572.5

Table 8.2: PoC USPD subset description.

The dataset was described by removing the InChIKey duplicates with no regard to their association with different reaction classes. This operation returned a total number of 34,264 unique molecules, thus reducing the subset by 23% in size. The filtered set was described using the RDKit Descriptor Calculation node, to produce the following descriptors: 'ExactMW', 'NumHeavyAtoms', 'NumHeteroAtoms', and 'NumRings'. Distributions were plotted as normalised histograms and reported in Figure 8.4, which shows that the property distributions covered by the PoC USPD subset are consistent with an optimal distribution of features for small-molecule drug discovery purposes (Veber *et al.*, 2002).

The level-4 reaction classes were decomposed into level-3, -2, and -1 labels and yielded 48, 37, and 11 unique classes, respectively. These progressively reduced numbers are due to the hierarchy adopted by the SHREC (see Section 5.4.6). Level-3 and -1 labels

were not investigated further because the first produced a number of classes similar to the number of level-4 classes, whereas the second yielded too few classes. Therefore, level-4 and -2 label types were selected for the preliminary screening. Their corresponding models describe problems of different complexity: A 50-label multi-label problem can be seen as 50 different binary problems; hence, a 37-label problem is much less complex.

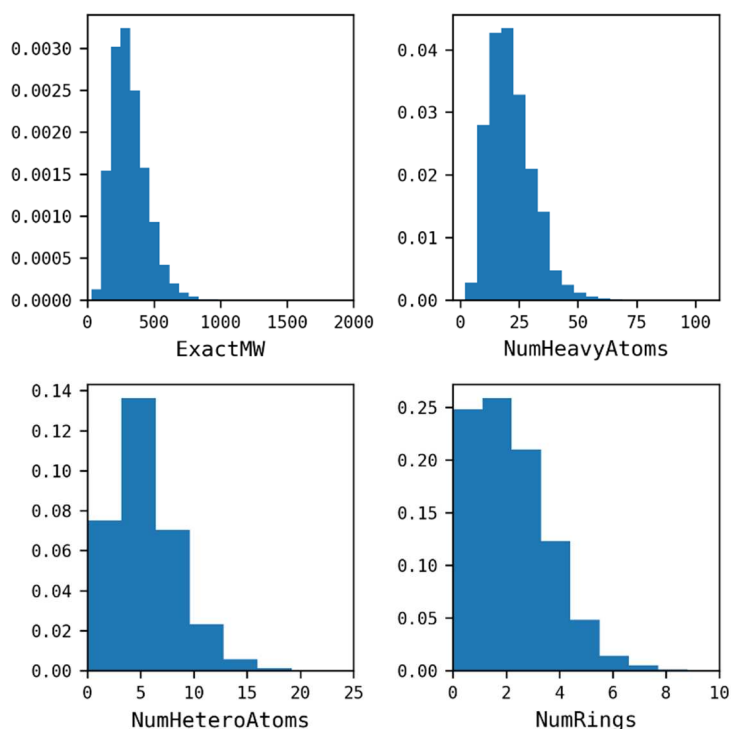


Figure 8.4: Property distributions covered by the PoC USPD subset.

The PoC USPD subset was checked for duplicate combinations of starting materials and reaction classes, according to the selected label types (i.e., level-4 and -2 labels). The two resulting subsets are described in Table 8.3:

Filtered PoC USPD Subset	Number of Starting Materials	Number of Classes	Median Number of SMs per Class
PoC USPD level-4 set	37,126	50	517
PoC USPD level-2 set	36,800	37	652

Table 8.3: Filtered PoC USPD subset descriptions.

The 44,222 molecule PoC USPD subset (Table 8.2) was reduced by 16% and 17% for level-4- and -2 labels, respectively. These results indicate a remarkable percentage of duplicate combinations of starting materials and classes, which can be explained by the

nature of the USPD set: patents often describe molecules that are combined with similar reagents to produce sets of analogues with the aim of covering certain regions of chemical space. Note that both subsets still describe imbalanced distributions of starting materials per class as reported in Figure 8.5, which shows that the main difference between the two datasets consists of a general increase in the number of examples of certain classes in the level-2 set due to the decomposition of some specific labels into bigger groups.

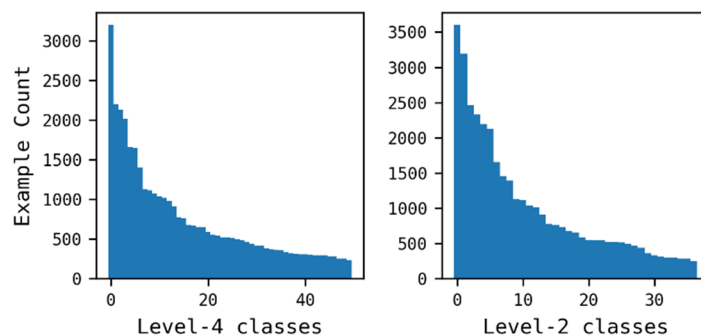


Figure 8.5: Level-4 (left) and level-2 (right) PoC USPD subsets class distributions.

The datasets were processed by encoding their starting materials as sets of molecular descriptions, which were then pivoted to associate each unique entity with multiple classes, as described in Section 8.2. The pivoted datasets are described in Table 8.4. Note that level-4 and -2 sets contain the same number of unique molecular descriptor entities; hence, they are described by the same number of rows. Preliminary information on the grouping ability of each molecular descriptor can be provided by the comparison between the PoC USPD subset filtered by InChIKeys (34,264 unique starting materials) and the pivoted datasets (Table 8.4): the mean number of unique descriptions across pivoted datasets is 32,109, which corresponds to an average reduction of 6% in the total number of rows. This result indicates that only a small amount of different structures was combined to form single descriptions during the *pivoting*.

Table 8.4 shows that descriptors characterised by high numbers of features (e.g. Avalon 2048-, 4096-, 8192-bit) often generated datasets with high numbers of unique entries, whereas descriptors with low numbers of features resulted in a smaller number of entries, for example, 13,263 unique entries for the ChemAxon fingerprint (110 bits).

Some exceptions can also be found: MACCS fingerprints (166 bits) produced a quite high number of unique entries demonstrating that high discrimination among molecular structures can be achieved with a low number of binary features, while Dompé and OChem EFG+ which encode 3047 and 2080 features, respectively, produced much lower discrimination compared to some shorter fingerprints. These results suggest that the outcome of the data pre-processing cannot be predicted *a priori* since it relies on both number and type of features, in addition to the content of the data. Consequently, training and test sets were generated for each dataset, using 80% and 20% of the data, respectively, using stratified sampling based on the number of labels associated with each entry.

Molecular Descriptor	Features	Unique Molecular Descriptions
Atom-pair	1024	34,225
Avalon	256	34,158
Avalon	640	34,199
Avalon	1024	34,200
Avalon	2048	34,203
Avalon	4096	34,210
Avalon	8192	34,212
CDK Functional Group	307	29,691
ChemAxon Functional Group	110	13,263
Chi and Kappa Descriptors	13	34,290
Dompé	3047	26,762
FeatMorgan (Radius 1) (Binary)	1024	30,238
FeatMorgan (Radius 2) (Binary)	1024	33,390
FeatMorgan (Radius 2) (Count)	1024	33,546
FeatMorgan (Radius 3) (Binary)	1024	33,514
Layered	1024	34,170
MACCS	166	33,195
Morgan (Radius 2) (Binary)	1024	34,120
OChem EFG+	2080	28,372
Pattern	1024	34,195
RDKit	1024	34,182
Torsion	1024	34,052

Table 8.4: PoC model: Molecular description datasets generated after the *pivoting*.

8.3.4. Multi-label Approaches and Classifiers

Several combinations of multi-label approaches and classifiers were selected: three scikit-multilearn (<http://scikit.ml/>) (Szymański and Kajdanowicz, 2017) Problem

Transformation (PT) (`skmultilearn.problem_transform`) methods (`BinaryRelevance`, `ClassifierChain`, `LabelPowerset`) combined with scikit-learn (Pedregosa *et al.*, 2011) combined with RF or SVM classifiers (`sklearn.ensemble.RandomForestClassifier`) (`sklearn.svm.LinearSVC`) were configured using default parameters, and one scikit-multilearn Adapted Algorithm (AA) (`skmultilearn.adapt.MLkNN`) was configured using three arbitrary neighbour levels. The default chain configuration in Classifier Chain creates label sequences alphabetically. Multi-label approaches are reviewed in detail in Section 4.5.1. Parameters are reported in Table 8.5:

Classifier	Parameters
RF	<code>n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=4, random_state=11, verbose=0, warm_start=False, class_weight=None</code>
SVM	<code>penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=11, max_iter=1000</code>
MLkNN	<code>k={3, 10, 15}</code>

Table 8.5: Classifier parameters.

8.3.5. Method

Models were created by combining the selected components as described in the tree diagram in Figure 8.6. The diagram describes an example of how models are enumerated combinatorially using a level-2 label dataset encoded using Avalon 1024-bit fingerprints, which is subsequently tested using Label Powerset combined with RF and SVM, and AA using three different configurations. The combination of these components yields 5 different models.

The theoretical number of combinations of descriptor types, classification label types and machine learning approaches is 396 (22 descriptors; 2 label types; 9 multi-label approach and classifier combinations). Rather than evaluate all possibilities, a systematic approach was taken as described in Section 8.3.6. The classifiers were trained using the training sets, then used to infer predictions on their corresponding test sets.

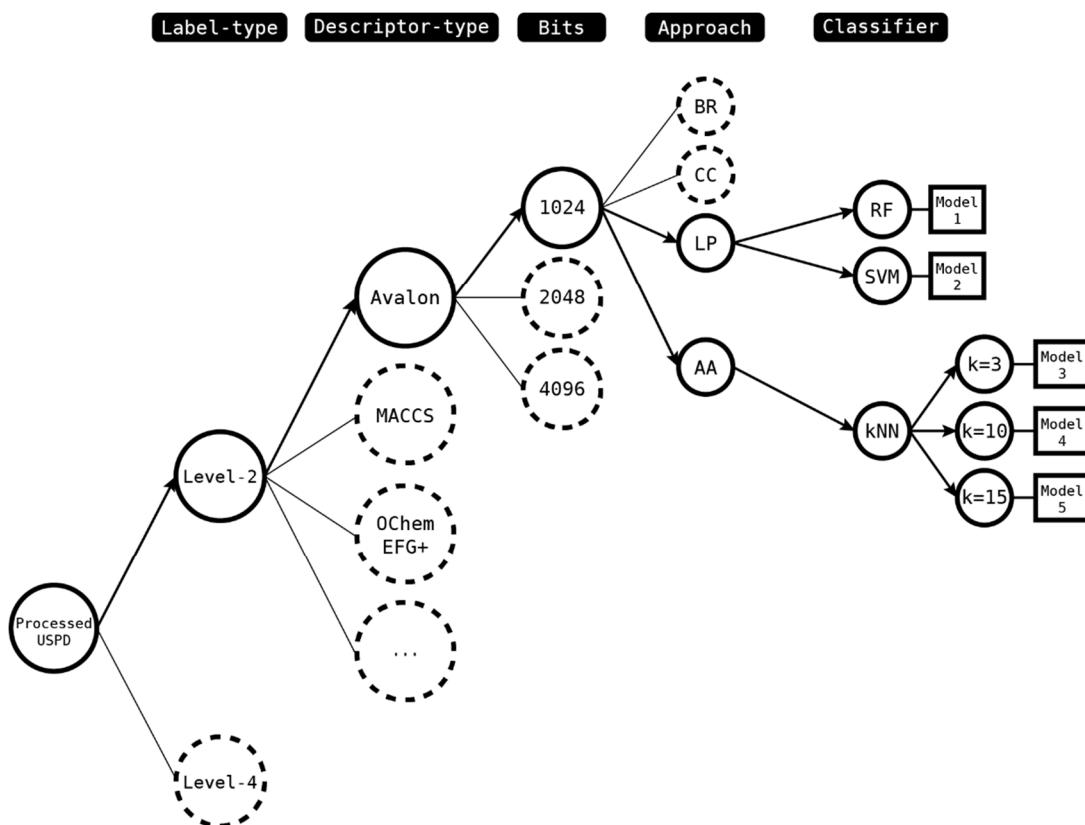


Figure 8.6: Model creation tree diagram: bolded nodes with directed edges represent an example of combinatorial enumeration, while dashed nodes with non-directed edges represent non-expanded paths.

8.3.6. Results and Discussion

True and predicted classes from the model validations were used to compute a series of metrics averaged across the classes: 0/1 Loss, Hamming Loss, Recall, Precision, and F1-score. Metrics definitions are discussed on page 247. Recall, Precision, and F1-score are expressed as *micro* averages. Note that, although the resulting models are all trained and validated using the data extracted from the PoC USPD subset, their performance can be compared only in an approximate way across different data and label types, due to the *pivoting* procedure, which yielded datasets of different number of entries (rows) and features (columns). Rather, a more accurate comparison can be done across different multi-label approaches that use the same training and test sets.

First, level-4 and -2 label datasets were compared using three PT approaches combined with RF: 44 datasets were screened as described above using Binary Relevance

(BR), Classifier Chain (CC), and Label Powerset (LP) to determine the best label-type for this problem. Their combination produced 132 models which were compared in pairs (i.e., level-4 vs. level-2 models). Results are reported in Appendix D and Recall, Precision, F1-score metrics are plotted in Figure 8.7.

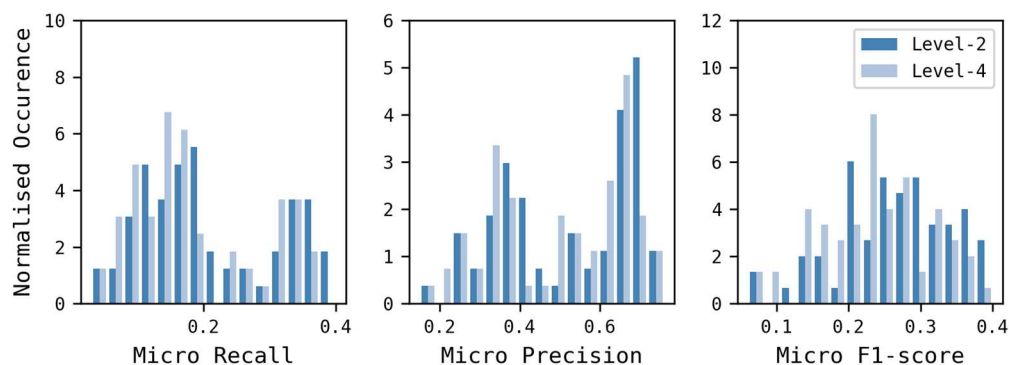


Figure 8.7: PoC model: Level-4- and level-2 label dataset comparison.

Figure 8.7 shows that level-2 labels describe a higher number of best performing models according to the F1-score. Results from Appendix D show that OChem EFG+ produced the best level-2 model (F1-score of 0.42), while Avalon 8192-bit produced the best level-4 model (F1-score of 0.38). Level-2 label models reported slightly better performance possibly due to the lower number of labels to predict and label specificity. Level-4 classes often contain detailed information on the reagent type, which is likely to be not discriminable by the starting material features, also because some reagents can actually be interchangeable in reality.

Second, PT approaches using RF were compared against AAs using the level-2 label datasets. 22 datasets were screened using BR, CC, LP, and three configurations (Table 8.5) of Multi-Label k-Nearest Neighbors (MLkNN), to determine the best multi-label approach for this problem. Their combination produced 132 models which were compared by data and approach types. Results are reported in Appendix D and Recall, Precision, F1-score are plotted in Figure 8.8, which shows clearly that PT approaches produced better performing models compared to AA, suggesting that transformation approaches are generally more suitable for reaction class recommendation.

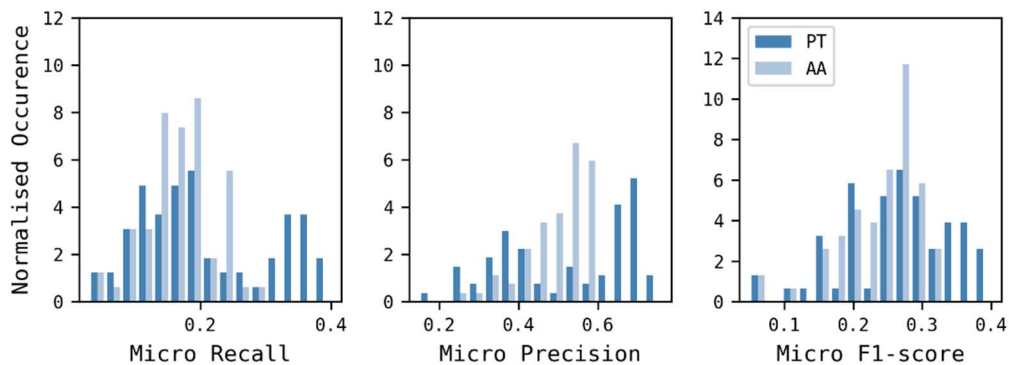


Figure 8.8: PoC model: PT and AA approaches comparison using level-2 label datasets.

Next, BR, CC, and LP approaches were assessed on the level-2 datasets using both RF and SVM classifiers. The 22 datasets, three PT methods and two classifier methods resulted in 132 models. Detailed results are reported in Appendix D. The transformation approaches were compared first without discriminating the RF and SVM classifiers. Average Recall, Precision and F1-scores are plotted in Figure 8.9 and summarised in Table 8.6:

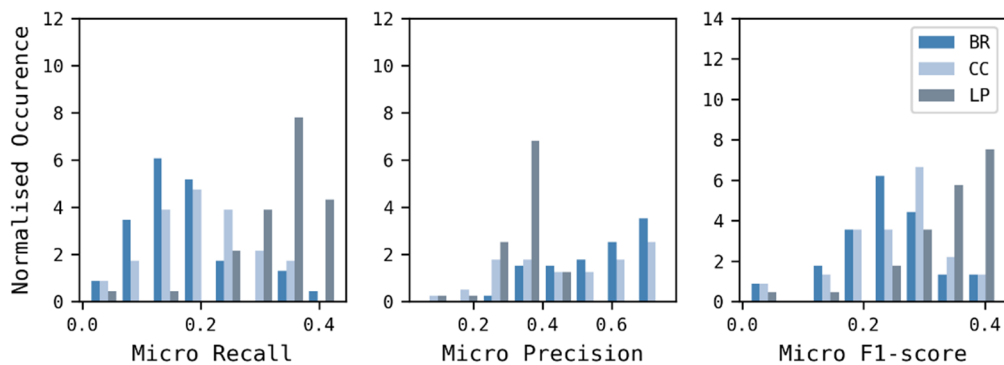


Figure 8.9: PoC model: Performance metrics comparison of PT approaches using RF and SVM and level-2 label datasets.

Method	<i>Micro Recall</i>			<i>Micro Precision</i>			<i>Micro F1-score</i>		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
BR	0.01	0.17	0.38	0.27	0.57	0.76	0.01	0.24	0.37
CC	0.03	0.20	0.37	0.07	0.48	0.76	0.06	0.26	0.37
LP	0.04	0.33	0.43	0.04	0.33	0.42	0.04	0.33	0.42

Table 8.6: PoC model: Performance metrics statistical analysis of PT approaches using RF and SVM and level-2 label datasets.

The F1-scores generally increased passing from BR to LP. A further inspection shows a positive and a negative trend for Recall and Precision, respectively. BR and CC reported considerably better Precision compared to LP, although their lower Recall suggest that they produced a higher number of false negatives. These trends also suggest that methods that account for *label dependence* such as CC and LP (see Section 4.5.1), tend to result in increased Recall and F1-score by sacrificing some Precision. This is because these techniques increase the chance of outputting labels that would not normally be predicted by classifiers which treat each label prediction as a separate binary problem. This characteristic is emphasized for LP due to the concatenation process that the algorithm performs on the labels. The general trend that emerges is that BR and CC yield high Precision models, while LP produces more balanced models.

0/1 Loss and Hamming Loss are plotted in Figure 8.10 and summarised in Table 8.7, which describe BR and CC as having similar metrics, characterised by lower Hamming Loss and higher 0/1 Loss, whereas LP reports the opposite trend, where for a small increase in Hamming Loss, it reports a remarkably lower 0/1 Loss.

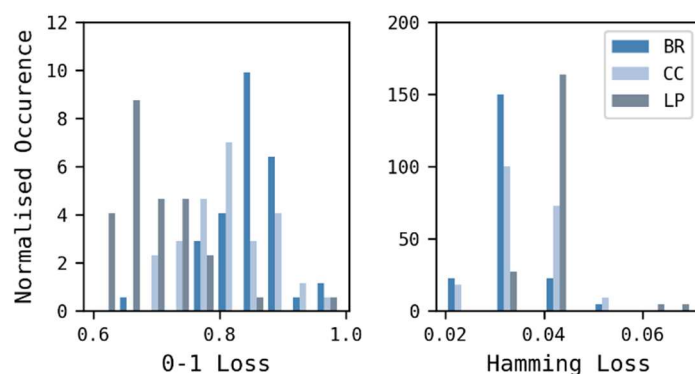


Figure 8.10: PoC model: Loss metrics comparison of PT approaches using RF and SVM and level-2 label datasets.

Method	0/1 Loss			Hamming Loss		
	Min	Mean	Max	Min	Mean	Max
BR	0.67	0.85	0.99	0.02	0.03	0.05
CC	0.68	0.81	0.97	0.02	0.03	0.05
LP	0.60	0.70	0.97	0.03	0.04	0.07

Table 8.7: PoC model: Loss metrics statistical analysis of PT approaches using RF and SVM and level-2 label datasets.

0/1 Loss and Hamming Loss are also correlated with *micro* F1-score in Figure 8.11, which shows that F1-score reports a close negative correlation to 0/1 Loss, whereas it does not correlate well to Hamming Loss, although a weak positive relationship can be detected. These results can be interpreted by considering that the models with higher F1-score are also those capable of accounting for *label dependence*: These models tend to produce more true labels since they benefit from class correlation, thus the number of entries that contain at least one error is reduced (i.e., lower 0/1 Loss), while the percentage of false labels is slightly increased (i.e., higher Hamming Loss).

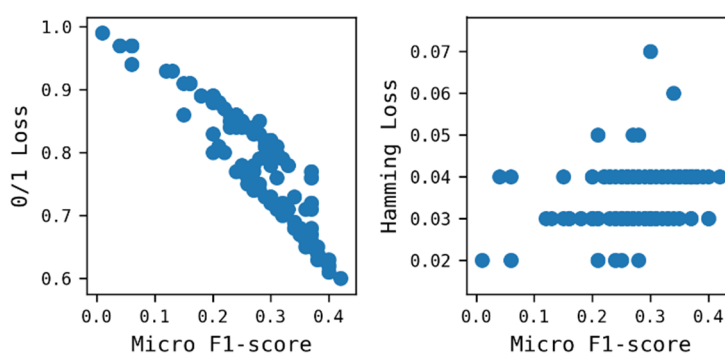


Figure 8.11: PoC model: Correlation plots between *micro* F1-score and 0/1 Loss (left) and Hamming Loss (right) for the level-2 label models.

The PT approaches were then discriminated by a classifier on the level-2 label datasets. Metrics distributions are plotted in Figure 8.12, which shows that most of the configurations worked similarly, although RF reported higher Precision and SVM reported higher Recall in some cases.

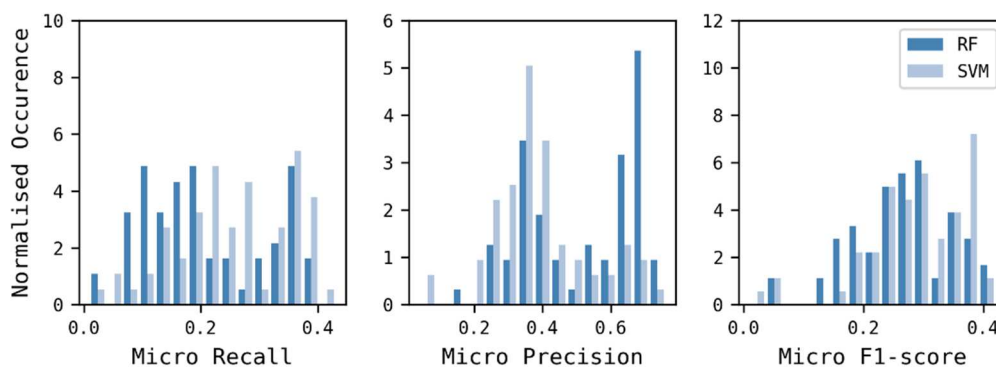


Figure 8.12: PoC model: Classifier comparison of PT approaches using level-2 label datasets.

Results were then sorted by *micro* F1-score and manually inspected to determine which models, in particular, produced the best scores. Results are reported in Table 8.8, which shows that OChem EFG+, Avalon, CDK Functional Group, FeatMorgan, and MACCS, reported the best metrics in the validation, suggesting that binary fingerprints are more effective in generalising molecular data. In addition, 14 best performing models out of 15 were obtained using LP, indicating the effectiveness of this approach in accounting for *label dependence*. The presence of PoC models associated with sufficiently good validation metrics indicates that these models were capable to determine effectively patterns across the data. These results substantiate the validity of the approach adopted to generate the data for the recommender and the configuration of the problem as a multi-label classification task.

Description	Setup	0/1 Loss	Hamming Loss	<i>Micro</i> Recall	<i>Micro</i> Precision	<i>Micro</i> F1-score
OChem EFG+	LP-SVM	0.60	0.04	0.43	0.42	0.42
Avalon 2048-bit	LP-RF	0.62	0.03	0.39	0.41	0.40
Avalon 4096-bit	LP-RF	0.62	0.03	0.40	0.40	0.40
Avalon 8192-bit	LP-RF	0.61	0.03	0.40	0.41	0.40
CDK Functional Group	LP-SVM	0.63	0.04	0.38	0.41	0.40
MACCS	LP-SVM	0.63	0.04	0.39	0.39	0.39
Avalon 2048-bit	LP-SVM	0.64	0.04	0.38	0.38	0.38
Avalon 4096-bit	LP-SVM	0.65	0.04	0.38	0.37	0.38
Avalon 640-bit	LP-SVM	0.65	0.04	0.38	0.37	0.38
Avalon 8192-bit	LP-SVM	0.63	0.04	0.39	0.38	0.38
FeatMorgan (Radius 1)	LP-SVM	0.64	0.04	0.37	0.38	0.38
FeatMorgan (Radius 2) (Binary)	LP-RF	0.64	0.04	0.37	0.38	0.38
Avalon 1024-bit	LP-RF	0.65	0.04	0.36	0.37	0.37
Avalon 1024-bit	LP-SVM	0.66	0.04	0.37	0.36	0.37
Avalon 2048-bit	BR-SVM	0.76	0.03	0.34	0.39	0.37

Table 8.8: PoC model: 15 best performing level-2 label PT models according to their *micro* F1-scores.

8.4. Final Model

8.4.1. Introduction

The PoC model provided information on how components affect the behaviour of the recommender, while condensing the potential number of configurations to test into

tens of models instead of hundreds. In this section, the recommender capabilities are extended in order to match with the 336-class reaction classification model described in Section 7.3. Ensemble methods are also introduced to overcome the limitations of PT approaches, then the best configurations are identified also according to their memory requirements.

8.4.2. Molecular Descriptors

Results from Section 8.3.6 confirm the use of binary fingerprints with a particular focus on methods capable to capture functional groups or pharmacophoric points, suggesting the selection of 5 molecular descriptor types: Avalon (1024-, 2048-, 4096-, 8192-bit), CDK Functional Group, FeatMorgan (Binary) (Radius 1) (1024-bit), FeatMorgan (Binary) (Radius 2) (1024-, 2048-bit), MACCS, and OChem EFG+ were selected as descriptors for the validation. FeatMorgan (Binary) (Radius 2) 2048-bit was also included to assess the performance of this fingerprint with a higher number of bits.

8.4.3. Data Selection

The “Balancing Tool” USPD Grants dataset described in Table 5.1 was selected due to its wide coverage of reaction classes. Note that this set has not been screened to remove duplicate reaction vectors that are associated with different starting materials. It should, therefore, be more effective for learning about the different environments in which a given reaction is feasible.

Reactions in classes falling outside of the scope of the reaction classification model were removed from the dataset leaving 336 classes, and the data was pre-processed to yield a set of starting materials and reaction classes. The dataset is described in Table 8.9 and its class composition is reported in Figure 8.13:

Dataset	Number of Starting Materials	Number of Classes	Median Number of SMs per class
Final USPD subset	1,056,836	336	799.5

Table 8.9: Final USPD subset description.

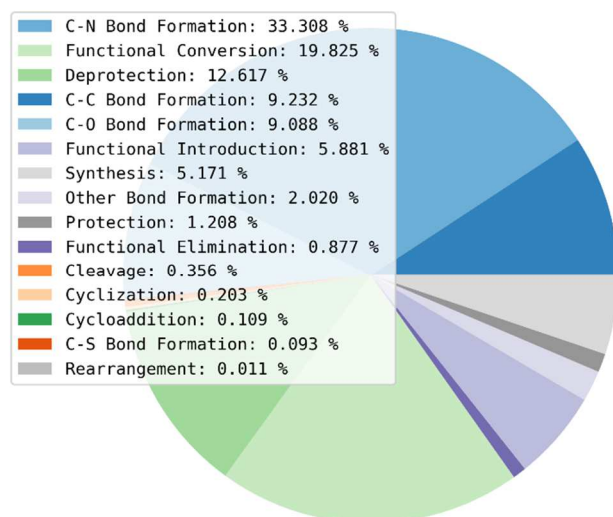


Figure 8.13: Final USPD subset level-1 class composition.

The characteristics of the data were assessed by first removing the InChIKey duplicates, with no regard for their association with different reaction classes. This returned a total of 360,477 unique molecules, representing a 66% reduction in size. This remarkable reduction indicates a higher presence of duplicates compared to the PoC data.

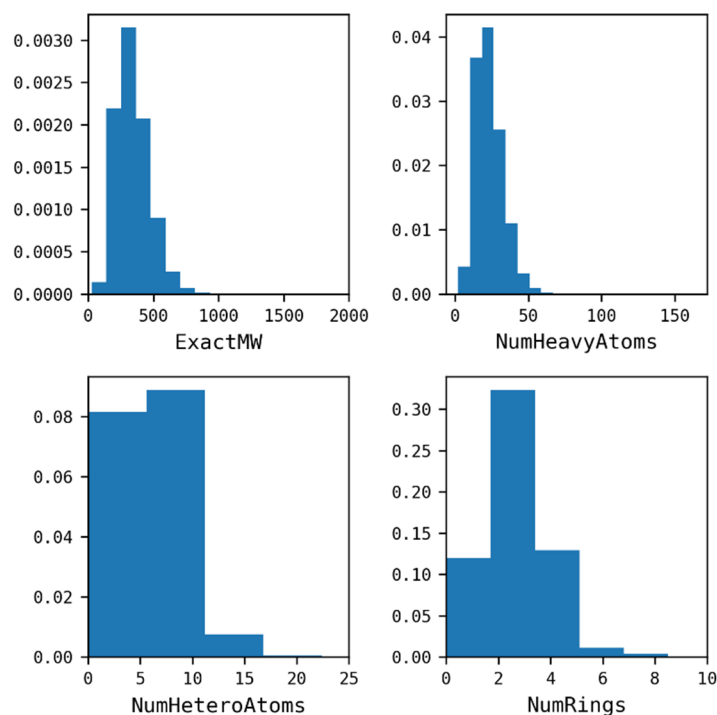


Figure 8.14: Property distribution covered by the Final USPD subset.

The property distributions of the unique molecules were plotted as normalised histograms and reported in Figure 8.14, which shows that the property distributions covered by the Final USPD subset are also consistent with an optimal distribution of features for small-molecule drug discovery purposes (Veber *et al.*, 2002).

The level-4 classes of the 1,056,836 reactions were decomposed into level-3, -2, and -1 labels yielding 319, 259, and 15 unique classes, respectively. Level-4 and -1 labels were excluded and level-3 and -2 labels were selected for the validation. The data was then checked for duplicate combinations of starting materials and reaction classes, according to the label types (i.e., level-3 and -2 labels). The two resulting subsets are described in Table 8.10.

Filtered Final USPD Subset	Number of Starting Materials	Number of Classes	Median Number of SMs per Class
Final USPD level-3 set	430,543	319	342
Final USPD level-2 set	424,138	259	290

Table 8.10: Final USPD Grants subset descriptions.

The 1,056,836 molecule Final USPD subset (Table 8.9) was reduced by 59% and 60% for level-3- and -2 labels, respectively. This substantial reduction indicates a high redundancy of starting materials associated with the same reaction class in the original set. This higher reduction compared to the PoC is due to the presence of duplicate reaction vectors (see Section 8.3.3). Distributions of reaction classes in both sets are plotted in Figure 8.15:

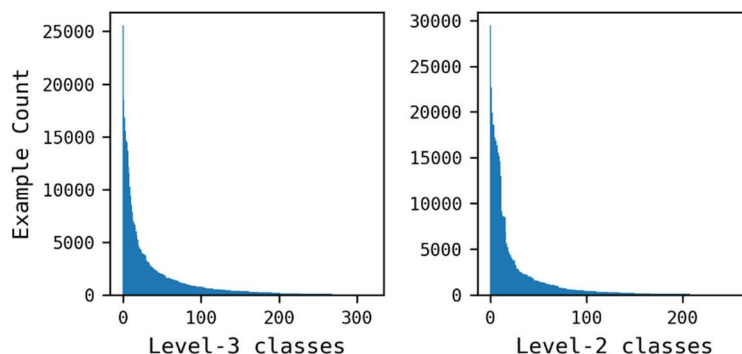


Figure 8.15: 319- (left) and 259-class (right) Final USPD subsets class distributions.

Molecular descriptors were then calculated for the molecules in the two datasets and the entries were pivoted with the reaction classes aggregated for identical descriptors sets, as shown in Figure 8.2. The pivoted datasets are described in Table 8.11:

Molecular Descriptor	Features	Unique Molecular Descriptions
Avalon	1024	358,648
Avalon	2048	358,765
Avalon	4096	359,019
Avalon	8192	359,055
CDK Functional Group	307	244,980
FeatMorgan (Radius 1) (Binary)	1024	270,049
FeatMorgan (Radius 2) (Binary)	1024	338,842
FeatMorgan (Radius 2) (Binary)	2048	338,879
MACCS	166	324,086
OCHEM EFG+	2080	241,230

Table 8.11: Final model: Molecular description datasets generated after the *pivoting*.

The mean number of unique descriptor sets across the pivoted datasets is 319,355, which corresponds to an average reduction of 11% in the total number of rows, based on 360,477 unique starting materials as a reference. This result indicates that the *pivoting* worked more effectively on these datasets compared to the PoC data. The dictionary fingerprints, such as CDK Functional Group and OCHEM EFG+, generally produced a higher data generalisation, thus condensing more structures within the same molecular description, whereas hashed fingerprints, such as Avalon and FeatMorgan, produced datasets characterised by sparser labels, i.e., less condensed. MACCS fingerprints represent an exception (see also Section 8.3.3): they are dictionary fingerprints that encode only 166 features, however, they produced discrimination almost comparable to FeatMorgan.

8.4.4. Multi-label Approaches and Classifiers

Three PT approaches and two classifiers were selected from the results in Section 8.3.6: BR, CC, and LP were combined with RF or SVM classifiers using default parameters as reported in Table 8.5. In addition, an Ensemble Method (EM) named Random k-Labelsets (RAkEL) was selected to overcome potential issues related to LP. The use of a much larger number of labels can result in the creation of a very imbalanced

distribution of examples per label-set, as well as requiring a huge amount of memory to run the algorithm. As explained in Section 4.5.1.3, RAKEL works through the construction of an ensemble of LP classifiers that are trained using smaller label-sets (i.e., combinations of labels) obtained from the random selection of k label subsets from the original label-sets. This way, the task is computationally less demanding and the label-set example distribution is less skewed. More specifically, both *disjoint* (RAkELd) and *overlapping* (RAkELo) strategies were investigated in this experiment. Both algorithms were combined with RF and SVM, and configured using default parameters as suggested in the paper by Tsoumakas *et al.* (2011) (Table 8.12).

Ensemble Method	Parameters
RAkELd	labelset_size=3
RAkELo	labelset_size=3, model_count=(number of labels multiplied by 2)

Table 8.12: RAKEL parameters.

8.4.5. Methods

Models were created following the same procedure reported for the PoC. The theoretical number of combinations of descriptor types, classification label types and machine learning approaches is 200 (10 descriptors; 2 label types; 10 multi-label approach and classifier combinations). Rather than evaluate all possibilities, a staged approach was taken as described below.

8.4.6. Results and Discussion

True and predicted classes from the model validations were used to compute a series of metrics averaged across the classes: Recall, Precision, and F1-score. Recall, Precision, and F1-score are expressed as *micro* averages. 0/1 Loss and Hamming Loss were not considered at this stage.

First, level-3 and -2 label datasets were compared using three PT approaches combined with RF, similarly to the procedure described for the PoC. Only 68 models were evaluated instead of 120 for the following reasons: OChem EFG+ (level-2 dataset) and FeatMorgan (Radius 2) 1024-bit (level-3 dataset) did not generate their corresponding SVM models for both the BR and CC approaches, possibly due to a bug

in the machine learning algorithm; LP models could not be trained due to memory issues (i.e., every model training exceeded 64 GB of RAM); Avalon 8192-bit reported a memory issue with every multi-label approach. Full results are reported in Appendix D and Recall, Precision, F1-score metrics are plotted in Figure 8.16:

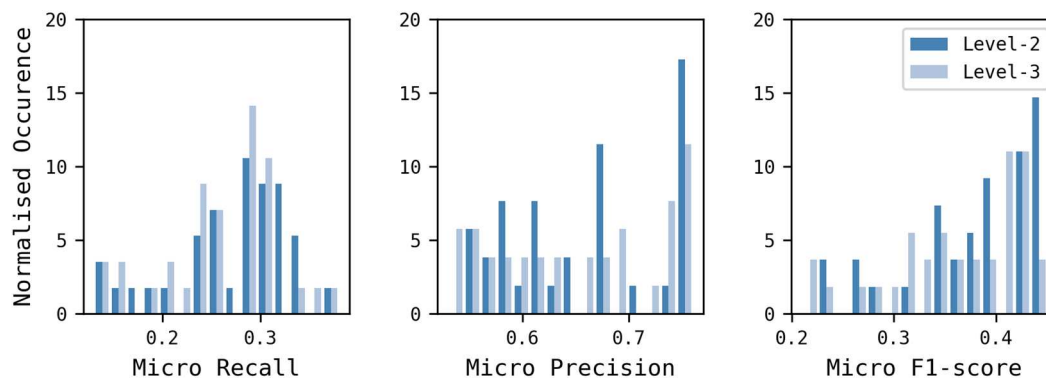


Figure 8.16: Final model: Level-3 and level-2 label dataset comparison.

Trends in Figure 8.16 are consistent with those found for the PoC. However, results from Appendix D show that in the final model validation, Avalon 2048-bit produced the best level-2 label model (F1-score of 0.45), while Avalon 4096-bit produced the best level-3 model (F1-score of 0.44), suggesting that hashed fingerprints could be more suitable than dictionary fingerprints when dealing with bigger datasets. In addition, results from the final model generally describe better metrics compared to the PoC model, suggesting that the use of a greater amount of training data can overcome the issues related to the introduction of a higher number of labels.

Maximum amounts of memory required across data-types and transformation approaches were also gathered to support the selection of the best label-type. Classifiers were not distinguished at this stage; hence the values reported in Table 8.13 represent the maximum amounts required by the most memory-consuming classifier per configuration. Results from Table 8.13 shows that level-3 models required only an average of 670 MB of extra virtual memory during their training and validation compared to level-2 models, hence suggesting the use of level-3 models since they are capable of predicting a higher number of labels. In addition, level-3 labels can be decomposed into level-2 labels, whereas the other way round cannot be done.

Molecular Descriptor	Maximum Memory Request (GB)		
	Features	Level-2 labels	Level-3 labels
Avalon	1024	14.7	15.3
Avalon	2048	25.6	26.4
Avalon	4096	48.1	48.7
CDK Functional Group	307	5.2	5.8
FeatMorgan (Binary) (Radius 1)	1024	11.1	11.6
FeatMorgan (Binary) (Radius 2)	1024	12.8	13.4
FeatMorgan (Binary) (Radius 2)	2048	24.2	24.8
MACCS	166	5.2	5.6
OChem EFG+	2080	17.0	18.3

Table 8.13: Final model: Maximum memory request per data and label type.

Second, level-3 label datasets were assessed using PT approaches and EMs combined with RF and SVM. 9 datasets (Avalon 8192-bit was excluded) were screened using BR, CC, RAKELd, and RAKELo, to determine the best multi-label approach. Only 70 of the possible 72 models were validated since FeatMorgan (Radius 2) 1024-bit did not generate its corresponding SVM models with the BR and CC approaches. Results are reported in Appendix D, and Recall, Precision, F1-score metrics, are plotted in Figure 8.17, and are summarised in Table 8.14:

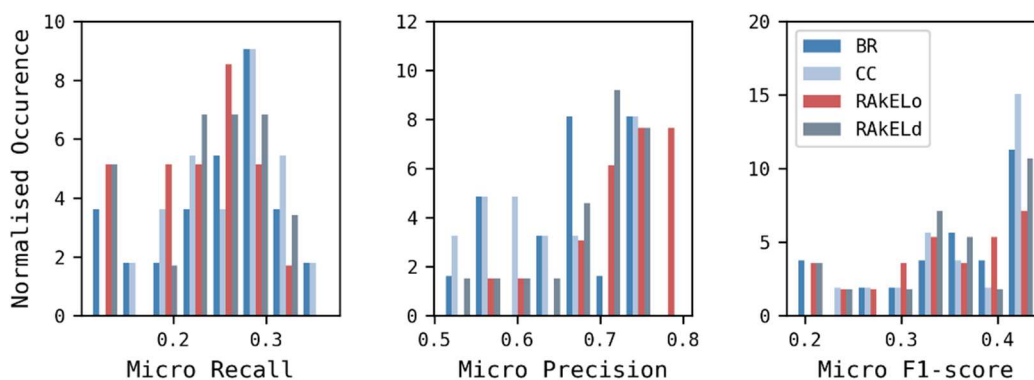


Figure 8.17: Final model: BR, CC, RAKELo, and RAKELd approaches comparison using level-4 label datasets.

Method	<i>Micro Recall</i>			<i>Micro Precision</i>			<i>Micro F1-score</i>		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
BR	0.13	0.25	0.35	0.53	0.66	0.75	0.21	0.35	0.43
CC	0.15	0.27	0.37	0.53	0.64	0.76	0.24	0.37	0.44
RAKELo	0.11	0.23	0.32	0.57	0.72	0.80	0.19	0.34	0.43
RAKELd	0.11	0.24	0.32	0.51	0.69	0.74	0.19	0.35	0.43

Table 8.14: Final model: Performance metrics statistical analysis of BR, CC, RAKELo, and RAKELd approaches using RF and SVM and level-3 label datasets.

Figure 8.17 and Table 8.14 show that PT approaches (BR and CC) reported better Recall, while EMs (RAkELo and RAkELd) reported better Precision. Table 8.14 also shows that CC generally reported better metrics compared to BR, except for Precision. The differences in performance between BR, CC, RAkELd, and RAkELo approaches were assessed statistically using one-way ANOVA (analysis of variance). No significant effect was found at the p -value <0.05 level for the four conditions, except for Precision which did show a significant effect (p -value=0.01).

PT approaches and EMs were then compared by maximum memory usage to determine the most efficient approach-type. Classifiers were not discriminated at this stage as discussed previously. Results are reported in Table 8.15:

Molecular Description	Features	Maximum Memory Request (GB)	
		BR-CC	RAkELo-RAkELd
Avalon	1024	15.3	16.6
Avalon	2048	26.4	27.6
Avalon	4096	48.7	49.8
CDK Functional Group	307	5.8	10.1
FeatMorgan (Binary) (Radius 1)	1024	11.6	16.3
FeatMorgan (Binary) (Radius 2)	1024	13.4	16.0
FeatMorgan (Binary) (Radius 2)	2048	24.8	25.3
MACCS	166	5.6	9.1
OChem EFG+	2080	18.3	21.2

Table 8.15: Final model: Maximum amounts of memory per data and approach type.

According to the results from Table 8.15, the average maximum amounts of memory for PT approaches and EMs correspond to 18.9 and 21.3 GB, respectively. Hence, EMs required an average of 2.4 GB of extra memory compared to PT approaches. Therefore, results from Figure 8.17 and Table 8.15 suggest the selection of PT approaches over EMs, and CC over BR, due to better memory efficiency and performance, respectively.

The retained configurations were then investigated by a classifier. Performance distributions are plotted in Figure 8.18, which shows that the difference between RF and SVM is more emphasised in the final model compared to the results reported in Figure 8.12 for the PoC model. In particular, RF reported better Precision in some cases,

while SVM reported better Recall for several models. In addition, F1-scores show that RF generally produced a higher number of best performing models compared to SVM.

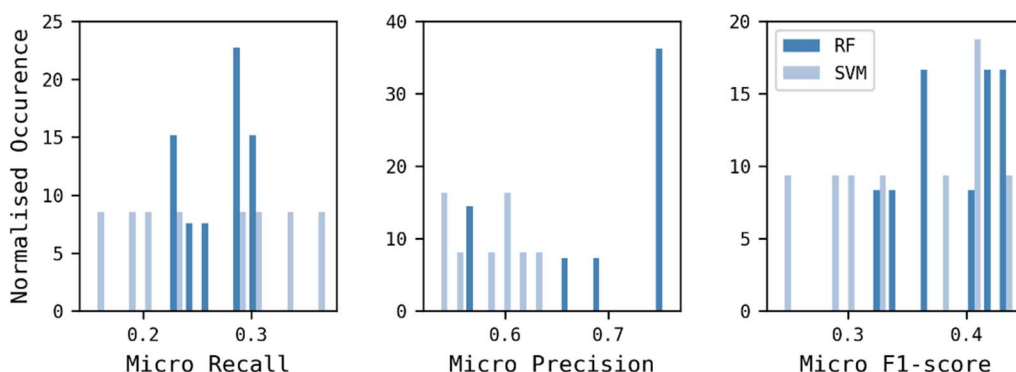


Figure 8.18: Final model: Classifier comparison for CC models using level-3 label datasets.

Results were analysed further by sorting them on their *micro* F1-scores to determine which combinations produced the best models. Results are reported in Table 8.16:

Description	Setup	0/1 Loss	<i>Micro</i> Recall	<i>Micro</i> Precision	<i>Micro</i> F1-score
Avalon 4096-bit	CC-RF	0.69	0.31	0.75	0.44
Avalon 4096-bit	CC-SVM	0.65	0.37	0.53	0.44
Avalon 2048-bit	CC-RF	0.69	0.30	0.75	0.43
Avalon 1024-bit	CC-RF	0.70	0.29	0.76	0.42
FeatMorgan 2048-bit Radius 2	CC-RF	0.69	0.29	0.75	0.42
Avalon 2048-bit	CC-SVM	0.69	0.33	0.54	0.41
FeatMorgan 1024-bit Radius 2	CC-RF	0.70	0.29	0.75	0.41
FeatMorgan 2048-bit Radius 2	CC-SVM	0.67	0.31	0.60	0.41
Avalon 1024-bit	CC-SVM	0.70	0.29	0.56	0.38
MACCS	CC-RF	0.74	0.25	0.69	0.37
OChem EFG+	CC-RF	0.73	0.26	0.57	0.36
FeatMorgan 1024-bit Radius 1	CC-RF	0.76	0.23	0.66	0.34
CDK Functional Group	CC-RF	0.76	0.23	0.57	0.33
OChem EFG+	CC-SVM	0.74	0.23	0.63	0.33
FeatMorgan 1024-bit Radius 1	CC-SVM	0.78	0.20	0.60	0.30
MACCS	CC-SVM	0.79	0.19	0.59	0.28
CDK Functional Group	CC-SVM	0.82	0.15	0.62	0.24

Table 8.16: Final model: level-3 label CC model performance metrics.

Table 8.16 confirms that RF is generally more effective than SVM, except for two cases (i.e., Avalon 2048- and 4096-bit) where SVM reported better Recall. However,

although the two classifiers did not report a large difference in performance, the multi-threading nature of RF supports the selection of this classifier.

The descriptor types associated with RF classifiers in Table 8.16 were further inspected to confirm the selection of the best model configurations. Avalon 4096-bit reported the best performance according to F1-score, and its 2048- and 1024-bit versions produced similar models, although characterised by remarkably lower memory requirements (Table 8.15). FeatMorgan Radius 2 yielded better models than Radius 1 suggesting that adding more neighbour information increases the model performance. MACCS also produced one valuable model which was surprising considering that the multi-label classification guidelines (Read, 2010) suggest using a number of features greater than the number of classes to predict. In addition, MACCS models only required a maximum of 5.6 GB of memory to be trained (Table 8.15). OChem EFG+ yielded one model with similar performance but this was 3.3 times bigger in terms of memory compared to MACCS.

8.4.7. KNIME Implementation

Reaction vector-based algorithms are already available in the KNIME Analytics Platform, thus the integration of the recommender in the same environment is desirable. The implementation of an automated workflow for reaction class recommendation combined with the structure generation algorithm is shown in Figure 8.19

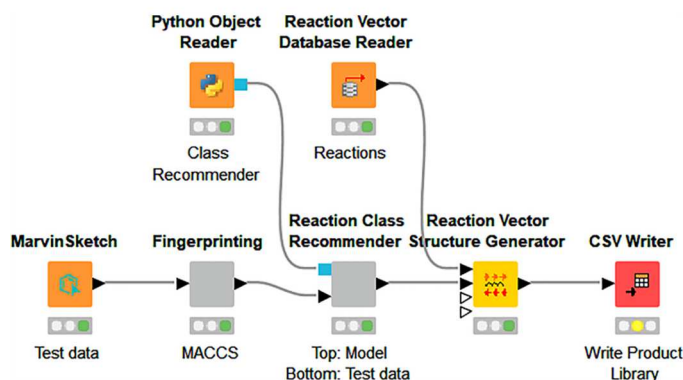


Figure 8.19: Recommended reaction vector-based design KNIME workflow.

The “Fingerprinting” node is meant to encode the molecular structures using a specified molecular descriptor (e.g. MACCS fingerprints), while “Reaction Class Recommender” is the core of the workflow, which accepts a starting material characterised by appropriate descriptors and produces a list of class suggestions that can be directly used to feed the “Reaction Vector Structure Generator” node to produce a set of recommended products. In this implementation, the model is dumped into a file which does not require any further training.

8.5. Validation of the Reaction Class Recommender

Metric-based evaluations can bring important insights for the selection of promising recommendation models but they do not demonstrate their practical use. In this section, the recommender is validated on the actual reaction class recommendations made in a retrospective experiment and then in reaction-based *de novo* design.

8.5.1. JMC 2018 Class Prediction

The recommender was applied to a set of starting materials extracted from a classified reaction dataset extracted from the Journal of Medicinal Chemistry. The recommended reaction classes were then compared with the annotated classes to quantify the performance of the model according to three conditions: if recommendations for a given starting material contained the correct class, the entry was considered as correctly classified; if the recommendations did not contain the actual class, the entry was considered as wrongly classified; if no recommendations were produced, the entry was flagged as non-recommended. The number of recommendations made for each starting material was also recorded. This experiment was also used to verify the generality of the recommender since the test set relates to a range of time that is not covered by the training data.

The procedure is reported as follows. The “Balancing Tool” JMC 2018 dataset described in Table 5.9 was classified using the Reaction Classification workflow described in Section 7.3.5. Entries associated with a credibility score lower than 0.25 were filtered out. 16,582 entries were retained (i.e., 67% of the dataset). The credibility threshold

used in this experiment was selected on the basis of the results described in Section 7.3.4.6, in order to obtain a set of highly reliable classified reactions. Classes were decomposed into level-3 labels, and duplicate SMILES associated with the same class were filtered out of the set. 11,539 entries were retained.

The level-1 class composition is reported in Figure 8.20, which shows that the content of the test set is similar to that of the training data in Figure 8.13.

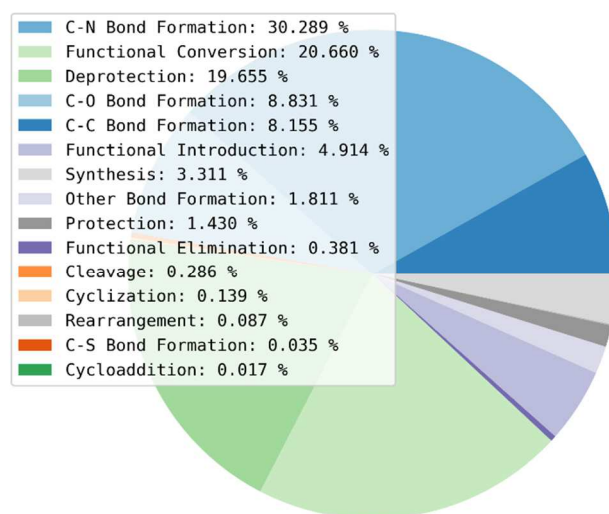


Figure 8.20: Level-1 classification of the JMC 2018 dataset.

The dataset was also described by property, as illustrated for the filtered PoC dataset in Section 8.3.3. Property distributions are reported in Figure 8.21, which also suggests broad comparability between training and test data.

Two recommenders trained using Avalon 1024-bit and MACCS fingerprints (Table 8.11) were then used to make recommendations for the starting materials and the recommendations were evaluated at different levels of the classification hierarchy (level-3, -2 and -1). The model configurations are reported in Table 8.5. Results (correct, wrong and non-recommended) are reported in Table 8.17 as percentages. The average numbers of recommendations per starting material were also determined for each model, and correspond to 1.9 (3.4 excluding the non-recommended entries) and 2.7 (4.4 excluding

the non-recommended entries) level-3 reaction classes per entry for Avalon 1024-bit and MACCS, respectively.

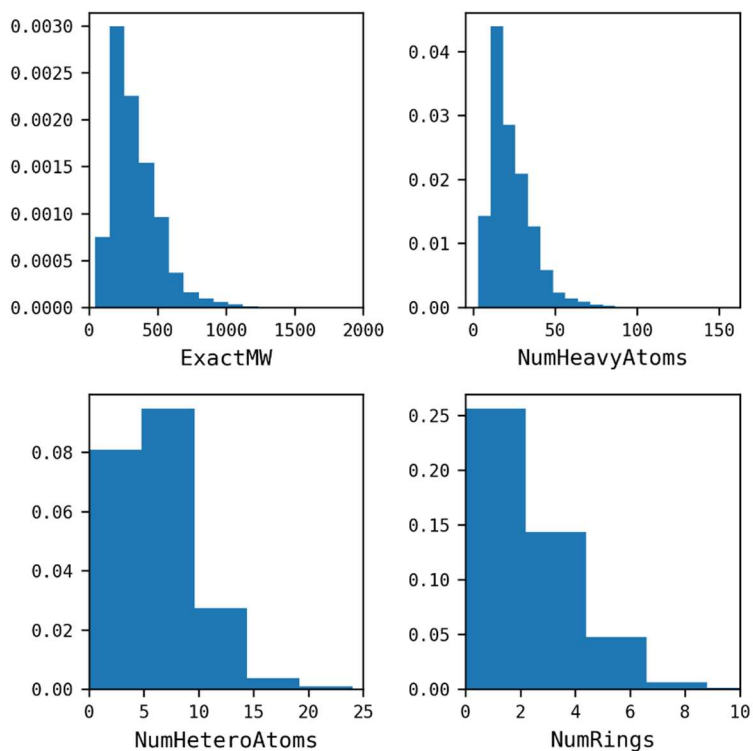


Figure 8.21: Property distribution of the starting materials extracted from the classified JMC 2018 test set.

Model	Label-type	Correct	Wrong	Non-recommended
Avalon 1024-bit	Level-3	33.5	21.9	44.6
	Level-2	34.4	21.0	44.6
	Level-1	41.9	13.5	44.6
MACCS	Level-3	37.1	23.1	39.8
	Level-2	38.0	22.2	39.8
	Level-1	45.0	15.2	39.8

Table 8.17: Performance of the Avalon 1024-bit and MACCS recommenders on the JMC 2018 dataset expressed as percentages..

Table 8.17 shows that MACCS reported a higher percentage of both correct and incorrect predictions compared to Avalon, which instead produced more non-recommended entries. This can be rationalised as follows. Avalon produced a lower compression rate in the training data preparation (Sections 8.3.3 and 8.4.3), which means that its entries are generally associated with a lower number of classes compared to those

in the MACCS datasets. In addition, Avalon has shown a higher precision compared to MACCS, which can correspond to a higher number of false negatives, thus resulting in a higher number of entries with no recommendations. Table 8.17 also describes similar trends for the two models across different levels of class information. The decomposition of the classes into more general labels did not change the percentage of non-recommended entries, yet it increased the chance of matching the correct labels due to the reduction of the total number of classes. However, note that this may not be desirable in a design scenario since the generalisation of the labels increases the number of actual reactions that would be applied to a given starting material, thus producing a higher number of products. The two models were further analysed by determining regions of separations between correct, wrong, and non-recommended entries according to the property distributions 'ExactMW' and 'NumHeavyAtoms'.

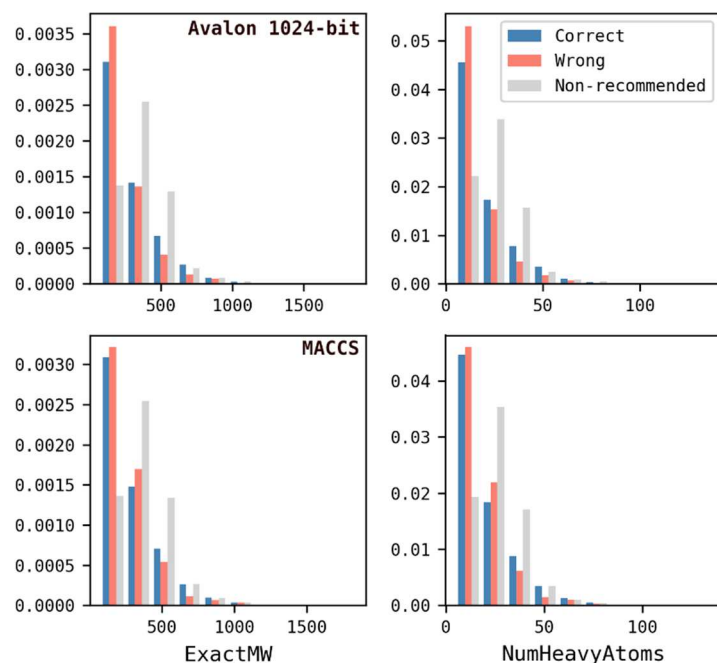


Figure 8.22: Property distributions for the starting materials coloured by correct (blue), wrong (red), and no-recommendation (grey) following application of the recommenders.

Results are reported in Figure 8.22, which shows that for both models, correct and wrong entries lie in the same ranges of values, whereas non-recommended entries increase as the molecule size increases. This result can be rationalised by comparing these

distributions with the training data properties reported in Figure 8.14. As the test entries start moving away from the property domain of the training set, the algorithm tends to do not output any recommendations for them. This is a consequence of multi-label approaches, which only output labels that really match certain algorithmic criteria (e.g. specific paths in decision trees), rather than always producing some output as multi-class classification algorithms do.

Predictions were also analysed by level-1 labels to determine whether wrong predictions and non-recommendations were more frequent for certain reaction classes. Wrongly predicted and non-recommended entry ratios are reported in Table 8.18:

Class	Examples	Wrong Prediction		Non-recommended	
		Avalon 1024-bit	MACCS	Avalon 1024-bit	MACCS
C-C Bond Formation	941	0.16	0.11	0.40	0.32
C-N Bond Formation	3495	0.13	0.11	0.50	0.47
C-O Bond Formation	1019	0.19	0.18	0.32	0.32
C-S Bond Formation	4	0.50	0.75	0.25	0.50
Cleavage	33	0.06	0.12	0.82	0.85
Cyclization	16	0.31	0.44	0.19	0.25
Cycloaddition	2	0.50	0.00	0.50	0.50
Deprotection	2268	0.05	0.03	0.52	0.42
Functional Conversion	2384	0.22	0.20	0.40	0.35
Functional Elimination	44	0.36	0.39	0.41	0.50
Functional Introduction	567	0.18	0.20	0.35	0.31
Other Bond Formation	209	0.22	0.21	0.52	0.50
Protection	165	0.37	0.42	0.27	0.26
Rearrangement	10	0.50	0.80	0.20	0.50
Synthesis	382	0.23	0.21	0.38	0.34

Table 8.18: Ratios of wrongly predicted and non-recommended entries at level-1 of the hierarchy for the Avalon 1024-bit and MACCS recommenders on the JMC 2018 set.

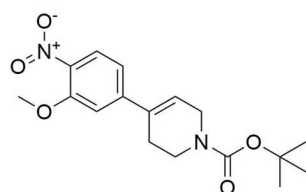
The comparison between results in Table 8.18 and percentages of reaction classes in Figure 8.13 shows consistent trends across the two models with larger differences for the classes described by fewer test examples, or that present more ambiguity, or difficulty for suggestion in absence of additional information. For example, “Deprotection” and “Functional Conversions” are similarly present in the training set (~13% and 20%, respectively) but the second reported almost 7 times as many wrong predictions (0.03 and 0.2, respectively) in the MACCS testing. This is because protecting

groups can be easily identified by fingerprints, and their presence typically involves deprotections, whereas functional groups can always be present in molecules for different purposes, even for biological interactions in the final structures. Non-recommended ratios are also consistent across the two models. Models were additionally compared by examining the intersection between their wrong predictions: Avalon 1024-bit and MACCS reported 2531 and 2661 total wrong predictions, respectively, of which 1585 are shared across the two validations (63% and 60% of the total number of wrongly predicted entries, respectively). This intersection indicates a close relationship between the two models due to the use of the same source of training data, however, the percentages of non-shared wrong predictions suggest that the two models treat some of the test data in different ways.

Wrong predictions were further analysed by manual inspection and revealed that although the true reaction classes were missing in the recommended classes, most of the entries actually received meaningful recommendations from both models. An example of an incorrect but meaningful prediction is reported in Figure 8.23.

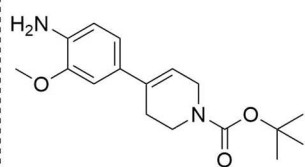
Original Reaxys Entry ID: 29929118

True class: Functional Conversion (Hydrogenation) (Alkene to alkane)



Recommendations:
[Functional Conversion (Nitro to amino)]

↓
USPD Grants
Reaction ID: US03931179-1976-1



New Product

Recommendations:
[Functional Conversion (Hydrogenation) (Alkene to alkane)]

Figure 8.23: Additional class recommendation test using the CC-RF MACCS model. The recommender did not suggest the class originally associated with the top molecule; however, the suggested transformation produces a new product for which the correct class is predicted.

These entries were transformed using the structure generation algorithm by applying their recommended classes, then new recommendations were produced on the resulting products. The new recommendations were then checked against the true classes to verify the validity of the model.

Figure 8.23 shows that the recommender did not produce the correct suggestions for the top molecule, possibly because the training data did not contain hydrogenations associated with that molecule-type. However, the product resulting from the transformation of the top molecule received the correct recommendation. This result suggests that the application of the recommender in a sequential way can still drive the algorithm towards the selection of appropriate classes to apply even if the correct suggestions are not produced in the first round.

8.5.2. DSPL Single-step de novo Design

A single-step *de novo* design experiment was carried out by integrating the recommender within the reaction vector-based structure generation framework. A set of fragments as starting materials, a set of reagents and a set of reaction vectors were selected for the construction of the design workflow. 72 fragments were selected from the DSPL screening library (Diamond Light Source, 2017) (Cox *et al.*, 2016) as a source of starting materials. Each starting material was known to reproduce one or more active compounds contained in the ExCAPE database (Sun *et al.* 2017). A set of reagents was selected from Sigma-Aldrich as a source of reagents. The 11,545 vector JMC 2008 database described in (Table 5.11) was selected as an external source of reaction vectors, that is, a set of reaction vectors that were not used in training the recommender. 56% of the entries in JMC 2008 were labelled as Unclassified.

Two control experiments were run, which consisted of: a full enumeration without the use of the recommender and including the unclassified entries from the reaction vector database (i.e., full database - “Control 1”); a full enumeration without the use of the recommender, excluding the unclassified entries from the reaction vector database (i.e., only classified reactions - “Control 2”). The experiment was then rerun using the

recommender with the recommended reaction classes acting as a filter on the reaction vectors, so that only those belonging to the recommended classes were considered for *de novo* design. The filtering was applied across three levels of hierarchy: level-3, -2, -1.

The CC-RF MACCS recommender was used to make recommendations for the selected 72 starting materials. Only 36% of the starting materials received suggestions (26 out of 72). The starting material properties were analysed to determine the separation between recommended and non-recommended entries according to the descriptors 'ExactMW' and 'NumHeavyAtoms'. Distributions are reported in Figure 8.24, which describes trends that are similar to those in Figure 8.22, which is the proportion of starting materials for which no recommendations are made, increases as the starting materials increase in their size.

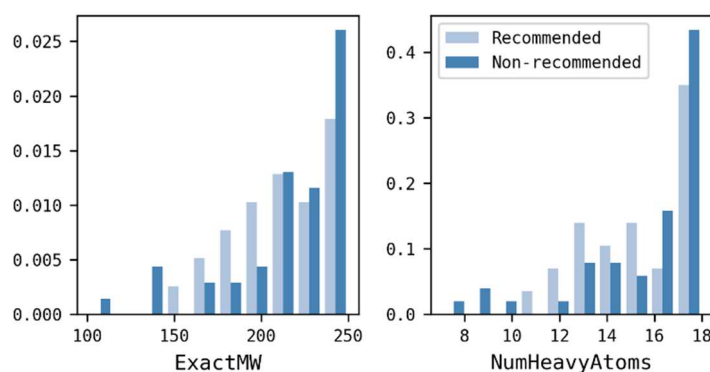


Figure 8.24: Property separation between recommended and non-recommended starting materials.

Recommendations were decomposed to yield level-3, -2, and -1 labels and statistically analysed. Results are reported in Table 8.19, which shows that moving up the hierarchy to more general classes reduces the mean number of recommendations per starting material. However, this generalisation actually increases the number of applicable reactions rather than reducing it, as discussed in Section 8.5.1. For example, the level-3 recommendations of “C-C Bond Formation (Coupling) (Suzuki)” and “C-C Bond Formation (Coupling) (Heck)”, would result in the application of reaction vectors that fall only within their reaction sub-classes (six reaction classes in total); whereas the overarching level-1 reaction class “C-C Bond Formation” includes 56 types of reaction

classes. Hence, the use of more general labels is expected to increase the number of applicable reaction vectors, and therefore the size of the product library that is generated. In particular, the numbers of applicable vectors correspond to 85, 94, 170, 238, and 357 for level-3, -2, -1, Control 2, and Control 1, pipelines respectively.

Suggested Classes per Starting Material				
Label-type	Minimum	Mean	Median	Maximum
Level-3	1	2.15	1.5	12
Level-2	1	2.08	1.5	11
Level-1	1	1.46	1	3

Table 8.19: The minimum, maximum, mean and median number of recommended classes per starting material for the different classification levels.

The design workflow was then run in five different modes (two controls and three levels of recommendations) for the 26 starting materials. Each of the five libraries was analysed as follows. The total number of unique products was determined by filtering out the InChIKey duplicates. The percentage of known active and presumed inactive compounds was determined by InChIKey comparison with ExCAPE. The proportion of actives was determined by dividing the percentage of actives by the percentage of inactives. Synthetic accessibility estimations were determined using RSynth (Chemical Computing Group ULC and ULC, 2019) and SAScore (Ertl and Schuffenhauer, 2009). In addition, the time required to enumerate each library was recorded. Results are reported in Table 8.20, where relative values are indicated in brackets:

Statistics per Library							
Mode	Number of Products	Time	Actives %	Inactives %	Active / Inactive Ratio	Mean RSynth	Mean SAScore
Level-3	43,952 (1.00)	381.22 (0.38)	0.13	0.09	1.44	0.57 (1.10)	1.69 (0.94)
Level-2	49,988 (1.14)	412.47 (0.42)	0.12	0.08	1.50	0.56 (1.08)	1.78 (0.99)
Level-1	73,741 (1.68)	585.03 (0.59)	0.11	0.08	1.38	0.54 (1.04)	1.83 (1.02)
Control 2	90,832 (1.82)	750.66 (0.76)	0.11	0.08	1.38	0.53 (1.02)	1.79 (0.99)
Control 1	141,834 (3.23)	991.65 (1.00)	0.07	0.05	1.40	0.52 (1.00)	1.80 (1.00)

Table 8.20: Library statistics for recommended and control pipelines.

Table 8.20 describes a clear trend passing from recommended to control pipelines. Level-3 and -2 recommendations produced similar numbers of products, whereas level-1, Control 2 and Control 1 pipelines produced collections that are 1.68, 1.82, and 3.23 times larger than the level-3 library, respectively. Table 8.20 also shows decreasing enumeration times as the specificity of the labels increases, and demonstrates that the use of the recommender speeds up the design process, with the level-3 recommender taking approximately half the time and one-third of the time of the Control 2 and 1 enumerations, respectively. In addition, the recommended libraries report an increasing trend in the percentages of both reproduced known active and inactive compounds compared to the control pipelines. This result can be interpreted in two different ways: On the one hand, an enrichment in the percentage of known compounds could indicate a lower tendency for novelty since the data used to train the model comes from syntheses that have already been carried out in the past, which can possibly bias the model towards the selection of a limited number of classes. On the other hand, this enrichment could represent the ability of the recommender to suggest reaction classes that are more likely to be applied in reality, thus reducing the number of structures that are actually inaccessible and/or irrelevant. Active/inactive ratios fluctuate in the same range of values across libraries, indicating that the recommender increased the known compounds with no preference for bioactivity. This hypothesis was tested by producing a random sample of the control library, which reported a coefficient of 1.76, thus indicating that this value can fluctuate widely although no actual enrichment in bioactivity is produced.

Average RSynth and SAScore across the libraries are also described in Table 8.20: The RSynth scores range between 0 and 1, where higher values mean higher accessibility; whereas, the SAScore ranges between 1 and 10, with higher values representing lower accessibility. For both scores, as the specificity of the labels increases, the average synthetic accessibility increases. Note that, although the libraries generated with the use of the recommender correspond to subpopulations of the control library, they all describe a clear shift toward higher synthetic accessibility values. An independent-samples t-test was conducted to compare RSynth and SAScore in the Control 1 ($M_{\text{RSynth}}=0.524$,

$SD_{RSynth}=0.201$, $M_{SAscore}=2.291$, $SD_{SAscore}=0.333$) and recommended libraries (Level-1 ($M_{RSynth}=0.543$, $SD_{RSynth}=0.190$, $M_{SAscore}=2.227$, $SD_{SAscore}=0.301$), Level-2 ($M_{RSynth}=0.563$, $SD_{RSynth}=0.033$, $M_{SAscore}=2.177$, $SD_{SAscore}=0.280$), Level-3 ($M_{RSynth}=0.571$, $SD_{RSynth}=0.181$, $M_{SAscore}=2.173$, $SD_{SAscore}=0.283$)). All pair-wise comparisons reported a significant difference ($p\text{-value}<0.0001$) in the scores using a confidence level of 95%. The effect sizes for these analyses were found to exceed Cohen's (1988) convention for a small effect ($d = 0.20$) except for the control-level-1 which reported Cohen's d_{RSynth} lower than 0.20. $RSynth$ and $SAscore$ values were also plotted as overlapping density plots in Figure 8.25 to show the synthetic accessibility shift.

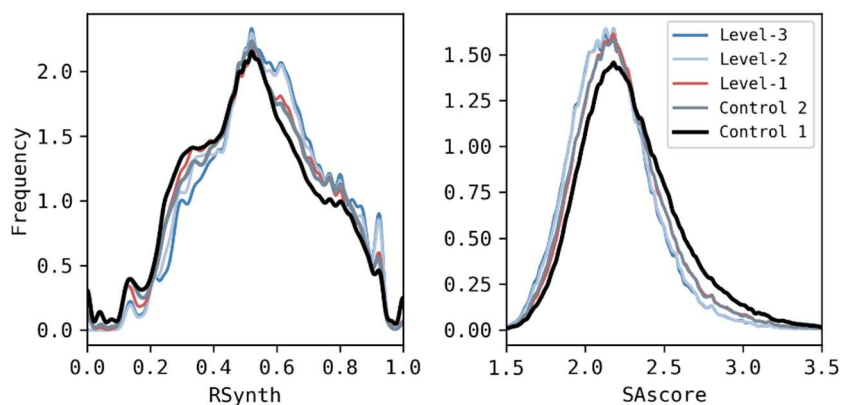


Figure 8.25: $RSynth$ and $SAscore$ distributions per library.

Product libraries were then analysed by percentages of applied reaction classes in order to determine possible effects of the recommender on the class distributions. Results are reported in Figure 8.26, where the two control pie charts are merged into one since unclassified reactions are not considered. Figure 8.26 shows consistent trends across recommended pipelines where only a limited set of level-1 classes are applied, whereas it reports a wider variety of classes in the control pipelines. More specifically, “C-O Bond Formation”, “Functional Elimination”, and “Protection” are not present at all in the recommended library pie charts indicating that these classes were not applied to any of the starting materials. Although 7 structures out of 26 presented functional groups suitable for “C-O Bond Formation” reactions, none of these structures suggested this reaction class. This can be interpreted as a lack of related examples in the training data (Figure 8.13), although a rarer class such as “Other Bond Formation” was produced for

three starting materials used in this experiment. “Functional Elimination” and “Protection” were not suggested at all possibly because of the low functionalisation and small dimensions of the starting materials, as well as due to the lack of examples in the training data.

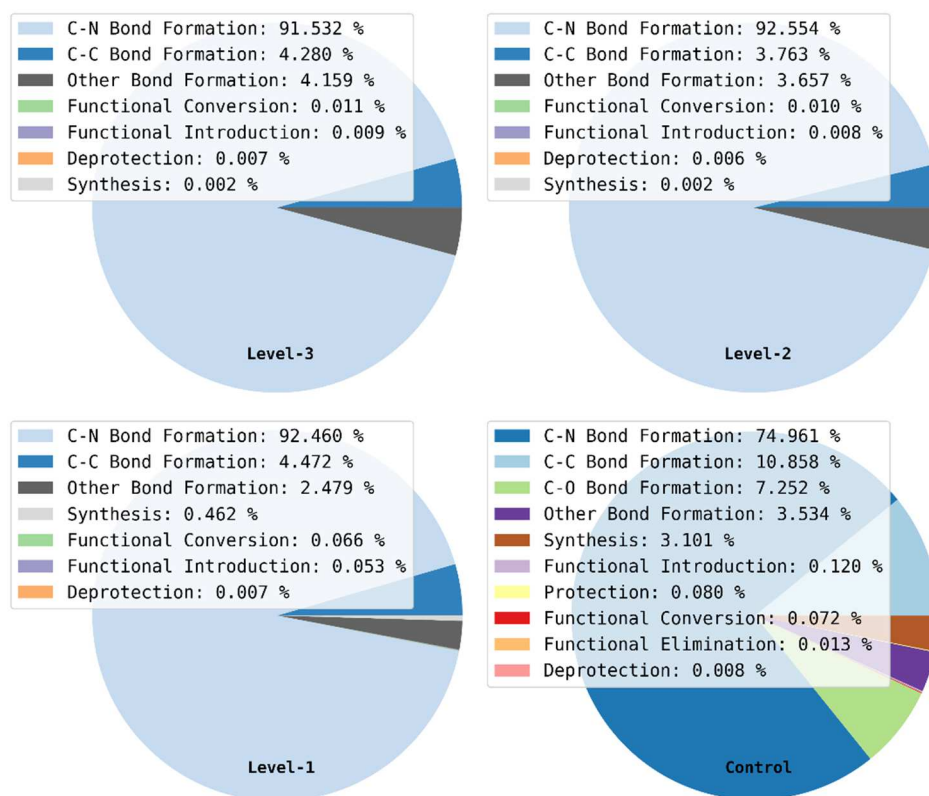


Figure 8.26: Level-1 label class distributions across libraries.

Libraries were further analysed by evaluating the identification codes of the top 25 targets hit by the reproduced active molecules in each dataset. This operation was done to determine the presence of a possible effect of the recommender on the distribution of target hits. The target information was extracted from ExCAPE. Results are reported in Figure 8.27, which shows that level-3 and level-2 distributions are identical, thus suggesting the presence of the same actives in the two datasets, whereas the other distributions report global percentage fluctuations along with a higher percentage of “Other Targets”, indicating that the application of more classes possibly expanded the target coverage. This hypothesis was verified by determining the number of unique targets hit per library which increased from 16 targets for the level-3 and level-2 libraries to 40, 42, and 44 targets for level-1, Control 2, and Control 1 libraries, respectively.

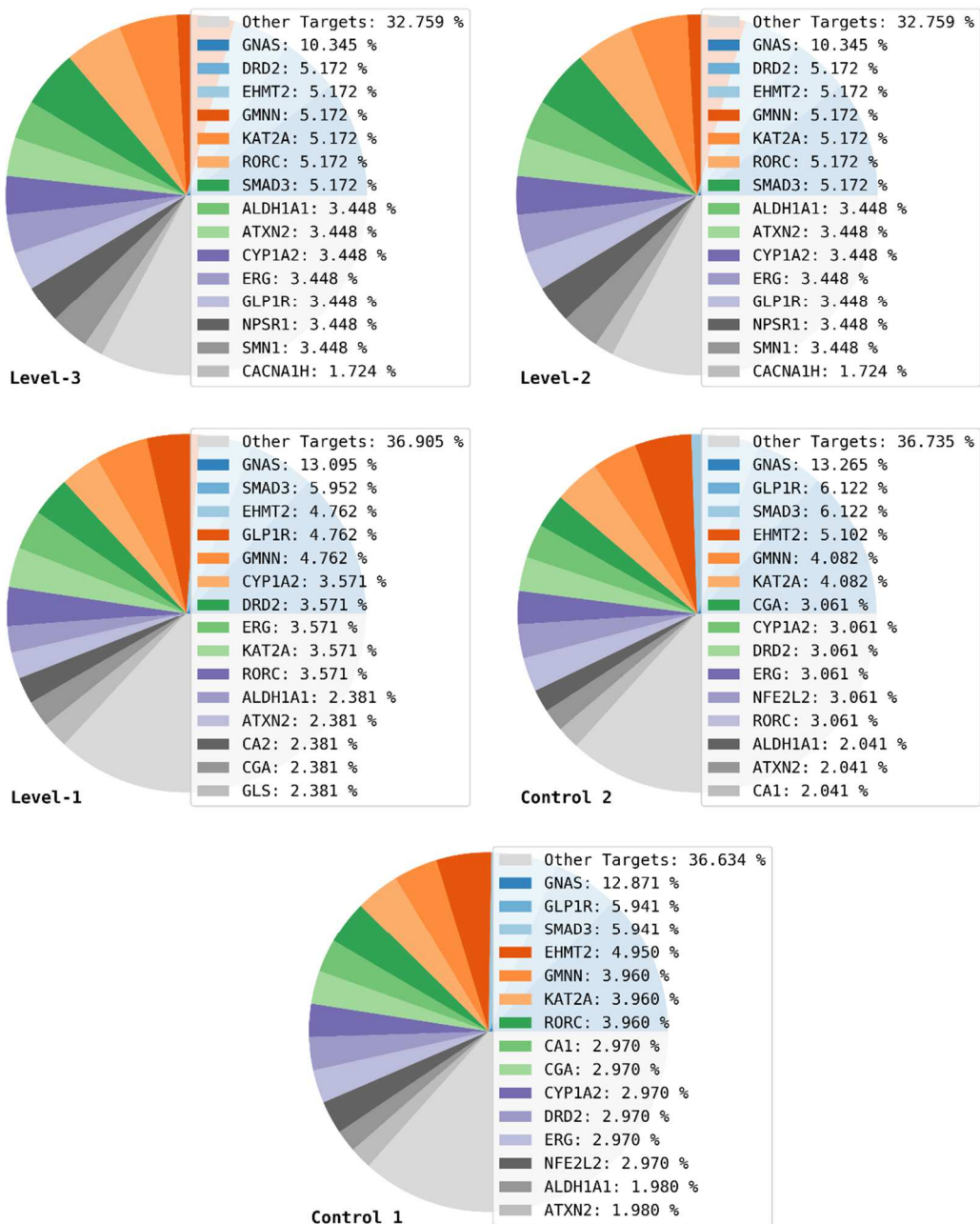


Figure 8.27: Target hits distributions across libraries.

8.5.3. Top 200 Drugs 2017 Recommended Validation

The validation reported in Section 6.3 was repeated by integrating the Avalon 1024-bit CC-RF model in the workflow in order to produce level-3 suggestions for each starting material processed during the design. Suggestions from the recommender were then used as a filter on the reaction vectors selected by the structure generator, so that only those belonging to the recommended classes were considered for *de novo* design. The

experiment was run using only the 92,530 USPD reaction vectors described in Table 5.4 as a source of reaction vectors.

Results were compared with those from the original experiment to provide a quantification of the effects of the recommender. As for the original experiment, 11 drugs failed the BRICS decomposition (see Section 6.3.5). The candidates generated from the drugs that were successfully decomposed by BRICS, were processed as reported in the original method, and the results compared with those in Table 6.2. Statistics are reported in Table 8.21, which shows that the use of the recommender led to a general decrease in the pair-wise similarities (-6% mean and -11% median) between drugs and their corresponding best compounds (i.e., compounds from the design associated with the highest similarity to the queries). This can be explained by the fact that fewer solutions are generally explored when the structure generator is combined with the recommender, hence the chances to find compounds with higher similarity to their references are also reduced. Nevertheless, according to the results reported in Section 6.3.5, the mean and median scores from the recommended design still rely in a range of values that reflect a good similarity between designed compounds and references.

Pipeline	Binary Fingerprint	Min	Max	Mean	Median
No Recommender	RDKit-ECFP4	0.19	1.00	0.62	0.60
	CDK-ECFP4	0.18	1.00	0.62	0.61
	RDKit-FCFP4	0.29	1.00	0.64	0.64
	CDK-FCFP4	0.23	1.00	0.65	0.64
Recommender	RDKit-ECFP4	0.23	1.00	0.59	0.55
	CDK-ECFP4	0.19	1.00	0.58	0.54
	RDKit-FCFP4	0.16	1.00	0.60	0.58
	CDK-FCFP4	0.21	1.00	0.61	0.58

Table 8.21: Statistics from the pair-wise similarities between queries and their corresponding best compounds from the USPD design - without and with the use of the recommender.

The comparison between the best compounds generated without and with the use of the recommender, is reported in Figure 8.28 for two cases that resulted in a notable drop in similarity. Figure 8.28 shows that the introduction of the recommender produced a drop of ~10% in similarity between best compounds and drugs, compared to the

candidates from the non-recommended design; however, the compounds from the recommended design still describe suitable group orientation and functionalities.

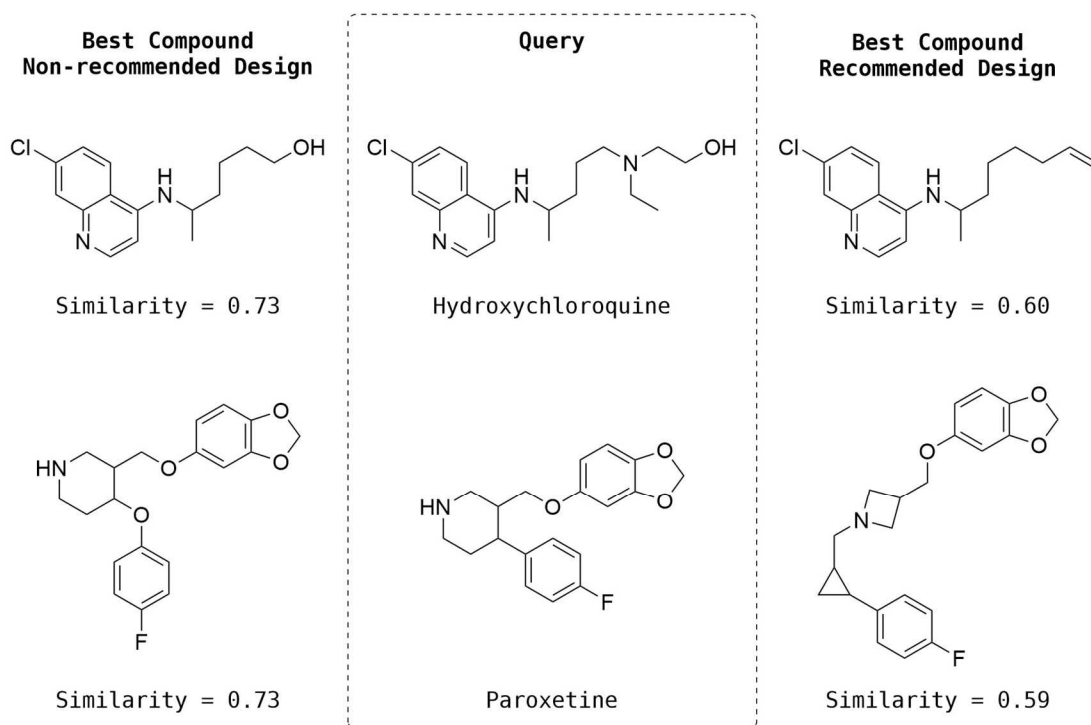


Figure 8.28: Comparison between best compounds generated without and with the use of the recommender. Designed compounds are also annotated with their similarity to their reference drugs using RDKit-ECFP4.

The recommended design also produced one example of best compound, with similarity to the query greater than that described by the candidate from the non-recommended design (Figure 8.29).

The analysis of the synthetic routes (3 steps) explained this result as a propagation of the constraints applied by the recommender at the beginning of the design. In particular, the first step involved the N-alkylation of piperidine (heterocycle), which, in the non-recommended experiment occurred by means of a “C-N Bond Formation (N-alkylation) (Chloro)” reaction, while in the recommended experiment this was not allowed since the suggestions did not contain this class.

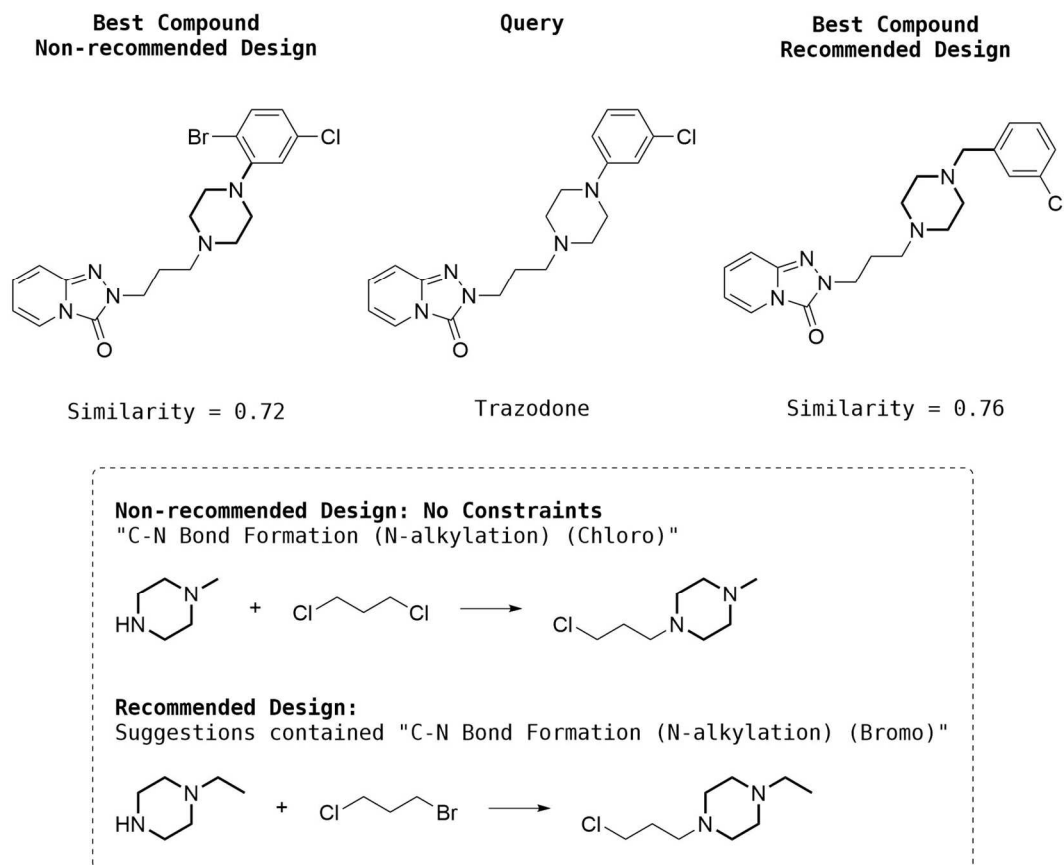


Figure 8.29: Example of best compound from the recommended design, associated with a similarity to its reference greater than that of the best compound from the non-recommended design. The reactions in the box describe the first step of the design, where the use of two different piperidines (highlighted in bold) results in the generation of different synthetic paths.

The inspection of the results also revealed that the use of the recommender led to the rediscovery of all the six drugs that were regenerated in the original experiment using the USPD vector database (Table 6.3). For these queries, the experiment was repeated to determine the differences in the number of generated products and enumeration times. These statistics are summarised in Table 8.22, which shows that the use of the recommender always led to a remarkable reduction in the number of products generated (i.e., solutions explored by the algorithm) and enumeration times. In particular, the mean reduction percentages of generated products and enumeration times correspond to 62% and 50%. These results suggest that the integration of the

recommender within the design framework can increase the efficiency of the structure generator while preserving the chance of finding relevant compounds.

Drug	Steps	Number of products generated		Enumeration times (hours)	
		Without Recommender	With Recommender	Without Recommender	With Recommender
Brimonidine	1	333,361	97,842 (-71%)	3.0	1.2 (-60%)
Glipizide	2	732,705	251,821 (-66%)	5.2	2.5 (-52%)
Glyburide	2	1,317,776	1,016,319 (-23%)	7.5	6.3 (-16%)
Levofloxacin	1	732,285	135,084 (-82%)	4.1	1.5 (-63%)
Naproxen	1	425,693	113,726 (-73%)	3.1	1.3 (-58%)
Rivaroxaban	3	1,282,308	536,212 (-58%)	7.7	3.9 (-49%)

Table 8.22: Comparison between drugs regenerated without and with the use of the recommender. Each ligand is described in the number of steps required for its regeneration, number of products generated by the algorithm and enumeration times.

8.6. Conclusions

In this chapter, the systematic development of an effective reaction class recommender has been described. Two promising models have been additionally validated in a series of experiments to further assess their performance, with special focus on *de novo* design applications. Results showed that models were capable of interpreting correctly the molecular features related to reactivity and suggest appropriate classes to apply. Statistics from the first *de novo* design experiment showed a reduction of the total numbers of generated structures and enumeration times, and a shift of the estimated synthetic accessibility of products towards higher values. The results from the second design experiment indicated that the use of the recommender led to a reduction of similarity between reference drugs and their corresponding best candidates, although this reduction did not result in a severe decrease. In addition, the results from the second design revealed that the integration of the recommender in the RENATE algorithm did not affect its ability to rediscover the drugs regenerated in the original experiment.

Chapter 9: Conclusions and Future Work

9.1. Conclusions

Reaction vector-based *de novo* design offers the advantage of accounting explicitly for the synthetic accessibility of virtually generated compounds using databases of reaction vectors that are created automatically from sets of known reactions. The work described in this thesis aimed at improving the existing reaction vector-based methods by means of additional reaction data, automated design, and machine learning.

Chapter 5 introduced the main issues associated with the use of reaction data, such as the lack of protocols for reaction standardisation and validation, and proposed a series of computational methods to address them. These methods were applied to a number of datasets to yield new standardised collections of reactions on which the entirety of this work relies. These collections were further processed to yield reaction vector databases and their corresponding reaction fingerprint datasets.

Chapter 6 demonstrated some of the benefits derived from the use of the new datasets by integrating the reaction vector structure generator within an automated ligand-based design framework referred to as RENATE. The algorithm steps consist of decomposing reference ligands into key fragments to identify similar reagents from a catalogue, then recombining reagents according to the selected scoring method.

The tool was first validated retrospectively in an experiment where a set of drugs and two reaction sets obtained from Section 5.4 were used as reference ligands and sources of transformation rules, respectively. The aim of the retrospective validation was to rediscover the reference ligands or generate similar products using similarity scoring. Results showed that, despite the constraints generally introduced by reaction-based methods, reaction vectors could produce highly similar structures to the references and also regenerate some of them.

The tool was then applied prospectively in a real case study to validate reaction vectors experimentally. The aim of the experiment was to demonstrate that reaction

vectors can be used to drive the synthesis of compounds and for effective *de novo* design. The experiment involved the design of ligands with improved brain penetration and affinity toward the biological target PARP1 using machine learning and docking as scoring methods. A number of candidates, which describe promising properties and interactions with the receptor, were submitted for synthesis to verify whether the procedures suggested by the structure generator could actually be used to support the compound preparation. A total of 7 out of 8 compounds were obtained using procedures adjusted to those proposed by the algorithm, demonstrating that reaction vectors can provide useful starting points for the preparation of selected structures. The selected compounds and their reference drugs were also evaluated on their estimated PK properties to provide evidence on the effectiveness of the reaction vector method for *de novo* drug design.

Chapter 7 describes the implementation of reaction vectors for reaction classification purposes. The work was aimed at obtaining a model that could be used to classify reaction data in order to augment the structure generator with the option of applying reactions by class. A number of supervised machine learning models were investigated using the patent sets obtained from Section 5.4 to determine an optimal setup for the model. The best performing model was combined with a Conformal Predictor to enable confidence estimation, and then used to classify two external datasets obtained from Section 5.5. Results from the experiments demonstrated that reaction classification can be used to obtain information on the composition of reaction sets and consequently to rationalise the behaviour of medicinal chemists across different work environments such as industry and academia.

Chapter 8 reported the construction of a reaction class recommendation model aimed at suggesting appropriate reaction classes to apply to molecules during the design according to their features (e.g. fingerprints). A large number of molecular descriptions, multi-label approaches, and parameters were investigated to rationalise the models' behaviours and determine a promising setup for use in *de novo* design. Two selected models were additionally validated to define their applicability domains and quantify

the effects of class recommendation on the structure generator. Results from the experiments indicated that models were capable of making associations between molecular features and reactivity, and that their integration in *de novo* design contributed to increasing the synthetic accessibility of product libraries while reducing number of solutions explored and enumeration times. In addition, results from the second design experiment showed that the integration of the model in RENATE significantly reduced the number of solutions explored by the algorithm without affecting its ability of regenerating the drugs found in the results of the original experiment.

9.2. Limitations and Future Work

The limitations of the approaches described in this thesis and the future work that could be done to address them are also reviewed. For example, a basic improvement in the reaction vector framework would be achieved by implementing a new version of the atom-pair descriptor, capable of accounting for stereochemistry. The introduction of this type of information would first support the reaction vector encoding algorithm by reducing ambiguities in atom pairs during the generation of recombination paths, and second, it would constitute the foundations of a stereospecific *de novo* design approach.

Two recent publications by Freilich and Ouellette (2019) and Jaworski and colleagues (2019) suggest potential advances for the content presented in Chapter 5. The first paper points out the presence of prophetic reactions in the patent data, questioning its suitability as a source of reaction templates for *de novo* design; hence, the use of a database of reactions from a more reliable source such as Reaxys, would increase the chance of generating structures that can be synthesised afterwards using the references in the database.

The second paper reports the benchmarking of several mapping tools including the Indigo Reaction Automapper used by the reaction standardisation workflow in Section 5.2, therefore, suggesting its replacement with a more accurate method due to its lower performance reported in the benchmark. However, the role of the mapping algorithm in the reaction standardisation workflow is limited to the identification of molecules that

describe reaction centres, hence the substitution of the mapping tool can be considered as a minor improvement of the workflow only. Some limitations can also be identified in SHREC (Section 5.4.6). The use of the patent data as a reference for the creation of the new labelling system could have introduced some bias in the current classification, hence restricting its use to certain data compositions. A potential improvement for SHREC would consist of revising the labels according to different and more diverse sources of classified reaction data.

A general trend that emerged from the experiments conducted in Chapter 6 was that RENATE produced more valuable results when applied to ligands described by smaller scaffolds and fragments that are attached linearly, hence suggesting a better suitability of the algorithm with this particular type of ligands. Some other improvements could relate to the replacement of BRICS with a more versatile algorithm due to the lack of certain fragmentation rules, and the implementation of a more sophisticated strategy for starting material/reagent role assignment.

Another substantial enhancement regards the replacement of the fingerprint-based methods used by the building block search module (see Section 6.2) with a different scoring technique, such as molecular descriptors. This is because fingerprints often do not work effectively on small query fragments resulting in low similarity scores for fragment pairs that are actually similar (Willett, 2013) (Hall *et al.*, 2017), or they can conversely yield fragments that are too similar to the queries in some circumstances. Nevertheless, the use of 2D descriptors has already demonstrated poor results (not reported here) with RENATE, hence 3D methods could be evaluated alternatively.

A final improvement of RENATE would consist of implementing it as a standalone program to support parallel computing. This is because only one query at time and a limited number of building blocks can be processed in KNIME due to the limitations imposed by the software environment. The standalone version of RENATE would perform more efficiently and could be promptly integrated in industrial schemes.

Following its computational validation, the application of RENATE to the design of new inhibitors with improved brain penetration for the target PARP1 has led to the selection and synthesis of a number of candidates. Although the pharmacokinetic properties of these compounds have been compared with those of their reference drugs using computational techniques, the experimental quantification of these properties, including the activities of the synthesised compounds, also constitutes further validation of the reaction vector method for *de novo* design.

Results from Chapter 7 suggested future improvements on the reaction classification model, such as the introduction of a more sophisticated fingerprint capable of encoding the reaction environment of certain low performing classes more effectively, and the replacement of the USPD collection with a balanced and curated training set. The integration of stereochemistry within reaction fingerprints would also enable the classification of an additional number of classes such as chiral inversions or resolutions. Furthermore, results from the classification of the external datasets suggested that reactions in pharmaceutical patents have a limited coverage of the organic reaction space. This was demonstrated by the lower percentage of reactions that could be predicted with sufficient confidence in the medicinal chemistry literature (~50%) compared to the ELN (~85%). Hence, the replacement of the training data with a more diverse source of reactions would increase the accuracy of the model.

The relatively high percentages of non-recommended entries reported in Chapter 8 also suggested a number of improvements for the reaction class recommendation model. These results evidenced a potential boundary within the domain of the applicability of the training data. The implementation of customised molecular descriptors capable of generalising reactivity features more effectively while preserving sufficient discrimination across different molecules, and a more complete source of training examples are expected to reduce the specificity of the recommender and improve its performance. Another improvement for the recommender relates to Classifier Chain models. In this particular approach, the evaluation of multiple labels during the chain construction is order

dependent so that a different ordering could account for label dependence differently; hence, the performance of models can be further increased by investigating this factor.

Metrics and Properties

Classification Metrics

Recall, Precision, and F1-score are the main classification metrics used in this thesis. These metrics are defined in Equation A. Recall determines the ratio of true positives inferred to the total number of positive instances in the test set (i.e., TP+FN), thus it provides a measure on the quantity of positive predictions without accounting for false positives, whereas Precision represents the ratio of true positives inferred to the total number of inferred positives (i.e., TP+FP), thus providing information on the quality of the positive predictions without considering false negatives. The harmonic average of Recall and Precision yields F1-score. These metrics range from 0 to 1, which indicate a completely wrong and correct classifications, respectively.

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad \text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad \text{F1-score} = \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Equation A: Definitions of Recall, Precision, and F1-score (TP, FN, and FP are true positives, false negatives, and false positives, respectively).

The Matthews Correlation Coefficient (MCC) is an additional metric evaluated in this work, which accounts for true and false labels both positives and negatives. This metric is described in Equation B. The MCC ranges from -1 to +1, which correspond to a completely wrong and correct classification, respectively.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

Equation B: Definition of the MCC (TP, FN, FP, and FN are true positives, false negatives, false positives, and false negatives respectively).

The selection Recall, Precision, F1-score, and MCC over other metrics is due to the imbalanced composition (i.e., many negative examples per class) of the datasets used in this work. In these contexts, the use of metrics such as Accuracy or ROC AUC, which account for the presence of true negatives, would result in the over estimation of the performance of individual classes (Powers, 2011).

Recall, Precision, F1-score, and MCC are binary classification metrics, hence they are calculated for each individual class, then they can be averaged using different methods to provide a global indication of how models perform on multiple classes. *Macro* averages are calculated by producing metrics for each individual class, then by averaging them with no regard to the example distribution across classes; hence, this method gives equal importance to each class. *Weighted* averages are generated by averaging across classes according to their support (number of true instances for a given class). These weights determine the relative importance of each class on the average: Classes that are more populated will have a higher effect on the score, whereas minority classes will have a lower impact. This method enables a more accurate evaluation of the performance on datasets with particular class distributions. *Micro* averages are calculated by considering all instances as they would belong to a single class. This method gives an indication of how a model performs globally without accounting explicitly for class weights.

Hamming Loss and 0/1 Loss are additional metrics used in this thesis, which are commonly used in multi-label classification. These metrics are defined in Equation C. Hamming Loss and 0/1 Loss have been introduced due to the nature of multi-label problems: In binary and multi-class classification, a given prediction can only be either correct or wrong, whereas in multi-label classification, it can be fully correct, fully wrong, or partially correct/wrong; hence, these metrics are aimed at providing additional information for the validation of multi-label models.

$$0/1 \text{ Loss} = \frac{I_E}{I} \quad \text{Hamming Loss} = \frac{FP+FN}{TP+TN+FP+FN}$$

Equation C: Definitions of 0/1 Loss (I_E and I are number of instances containing at least one error and total number of instances in the dataset, respectively) and Hamming Loss (TP , FN , FP , and FN are true positives, false negatives, false positives, and false negatives, respectively).

0/1 Loss corresponds to the fraction of entries that contain at least one error divided by the total number of entries. 0/1 Loss determines whether any false label is inferred to a given test entry: If predicted labels are all correct, the entry is classified as correct, otherwise, the entry is classified as wrong. Hamming Loss determines the ratio of false

labels to the total number of labels, thus providing a measure of the model performance that can be used in comparison with strict metrics such as 0/1 Loss, or measures such as Precision that do not account for false negatives. Hamming Loss is useful in the evaluation of models where the presence of any sort of false predictions affects severely the utility of the model (e.g. medical diagnosis). 0/1 Loss and Hamming Loss are loss functions so their optimal value is zero.

Regression Metrics

The coefficient of determination (R^2), Mean Absolute Error (MAE), and Mean Squared Error (MSE) are the regression metrics used in this thesis. These metrics are defined in Equation D. R^2 reflects the proportion of variance for a dependent variable that is explained by an independent variable, hence the strength of the correlation between these two variables. MAE and MSE measure the mean absolute and mean squared differences between true and predicted values (i.e., error) in a set of predictions, respectively.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y - \bar{y})^2} \quad MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Equation D : Definitions of R^2 , MAE, and MSE (y , \hat{y} , and \bar{y} are the actual, predicted, and mean values, respectively)

R^2 ranges from any negative real number to a maximum of 1, where negative values indicate that the model explains the data worse than a horizontal line, zero indicates that the model explains the data equally as a horizontal line, while positive values indicate that the model correlates with data. A value of 1 indicates that the model (independent variable) explains perfectly the data (dependent variable).

MAE and MSE both describe errors, hence the closer they are to zero, the better is the model. The difference between them is that MSE penalises larger errors, while MAE does not. Thus, their comparison provides information on the nature of errors generated by models.

Pharmacokinetic (PK) Properties

A list of properties used in this work is described on their use in pharmacokinetics (PK), in particular to increase the likelihood of effective BBB penetration: ‘logD’ (distribution coefficient) is the ratio of concentrations of a compound (ionized and unionized) in a mixture of two immiscible phases at equilibrium (e.g. octanol/water). ‘logD’ represents a measure of hydrophobicity which is an indicator for oral absorption and cell membrane permeation (Kwon, 2002). An optimal range for ‘logD’ is between 0 and 3 for CNS drugs (Pajouhesh and Lenz, 2005);

‘HBD’ (hydrogen bond donor count) is a direct index of the ability of a compound to form hydrogen bonding with the solvent. Compounds with high hydrogen bond forming potential are ineffective for BBB penetration (Pardridge, 2002). CNS guidelines suggest to have ‘HBD’ ≤ 3 (Travis T Wager *et al.*, 2010) (Ghose *et al.*, 2012);

‘TPSA’ (total polar surface area (\AA^2)) is the sum of surface contributions of polar atoms also including their attached hydrogens. ‘TPSA’ has been correlated with properties such as intestinal absorption or BBB penetration. CNS guidelines suggest to have ‘TPSA’ ≤ 90 (Kelder *et al.*, 1999);

‘Caco2 A-B’ (apical to basolateral (A-B) direction (nm/s)) is the speed at which a compound diffuses passively through Caco2 cell monolayers. ‘Caco2 A-B’ provides a measure of permeability across the intestinal barrier, hence reflecting the suitability for oral administration (Kwon, 2002). ‘Caco2 A-B’ values in this work were predicted by machine learning regression. Higher ‘Caco2 A-B’ values indicate higher permeability;

‘CNS MPO’ is a multi-parametric score that combines six physicochemical properties, including ‘logD’, ‘HBD’, and ‘TPSA’, to estimate the probability of a compound to show optimal properties for CNS candidates. Marketed CNS drugs reported ‘CNS MPO’ ≥ 4 , using a scale of 0-6 (Travis T. Wager *et al.*, 2010) (Wager *et al.*, 2016). Two versions of ‘CNS MPO’ have been proposed (v1/v2): The first devised using the formula from the original paper, and the second using an updated formula where ‘logP’ is discarded and ‘HBD’ is counted twice (Rankovic, 2017).

Appendix A

NameRXN Subclass	NameRXN Reaction-type	Sheffield Level [1]	Sheffield Level [2]	Sheffield Level [3]	Sheffield Level [4]
Other functional group addition	Methylation	C-C Bond Formation	Methylation		
O-containing heterocycle formation	Aldehyde Darzens reaction	C-C Bond Formation	Condensation	Carbonyl to alpha-beta-epoxy ester	Darzens
Other C-C bond formation	Knoevenagel condensation	C-C Bond Formation	Condensation	Aldol/Knoevenagel	
Other C-C bond formation	Blanc chloromethylation	C-C Bond Formation	Methylation	Blanc chloromethylation	
Other organometallic C-C bond formation	Iodo aldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Iodo
Friedel-Crafts reaction	Friedel-Crafts acylation	C-C Bond Formation	Acylation	Friedel-Crafts	
Other C-C bond formation	Aldol condensation	C-C Bond Formation	Condensation	Aldol/Knoevenagel	
Other organometallic C-C bond formation	Bromo aldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Bromo
Other C-C bond formation	Alkyne + aldehyde reaction	C-C Bond Formation	Alkyne + aldehyde		
Other C-C bond formation	Wurtz-type coupling	C-C Bond Formation	Coupling	Wurtz-Fittig	
Wittig olefination	Wittig-type olefination	C-C Bond Formation	Olefination	Wittig	
Other organometallic C-C bond formation	Iodo ketone Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Iodo
Other organometallic C-C bond formation	Bromo ketone Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Bromo
Other C-C bond formation	Ullmann-type biaryl coupling	C-C Bond Formation	Coupling	Ullmann-type	
Other C-C bond formation	Alkyne + ketone reaction	C-C Bond Formation	Alkyne + ketone		
Other C-C bond formation	Horner-Wadsworth-Emmons reaction	C-C Bond Formation	Arylketone formation	Horner-Wadsworth-Emmons	
Other C-C bond formation	Aldol addition	C-C Bond Formation	Coupling	Aldol addition	
Wittig olefination	Wittig olefination	C-C Bond Formation	Olefination	Wittig	
Other organometallic C-C bond formation	Simmons-Smith reaction	C-C Bond Formation	Cyclopropanation	Simmons-Smith	
Other organometallic C-C bond formation	Reformatsky reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Bromo

Other organometallic C-C bond formation	Decarboxylative coupling	C-C Bond Formation	Carboxylic acid + halide	Decarboxylative	
O-containing heterocycle formation	Ketone Darzens reaction	C-C Bond Formation	Condensation	Carbonyl to alpha-beta-epoxy ester	Darzens
Other C-C bond formation	Wurtz-Fittig coupling	C-C Bond Formation	Coupling	Wurtz-Fittig	
Other C-C bond formation	Houben-Hoesch reaction	C-C Bond Formation	Arylketone formation	Houben-Hoesch	
Other Pd-catalyzed reactions (Negishi,Kumada,etc.)	Kumada coupling	C-C Bond Formation	Coupling	Kumada	
Grignard reaction	Bromo Grignard reaction	C-C Bond Formation	Coupling	Grignard	Bromo
Other C-C bond formation	Cyanoalkane alkylation	C-C Bond Formation	Alkylation	Cyanoalkane	
Other C-C bond formation	Henry reaction	C-C Bond Formation	Nitroalkanes + aldehyde/ketone	Henry	
Other C-C bond formation	Perkin reaction	C-C Bond Formation	Condensation	Cinnamic acid formation	Perkin
Grignard reaction	Chloro Grignard reaction	C-C Bond Formation	Coupling	Grignard	Chloro
Grignard reaction	Iodo Grignard reaction	C-C Bond Formation	Coupling	Grignard	Iodo
Other C-C bond formation	Nitroalkane alkylation	C-C Bond Formation	Alkylation	Nitroalkane	
Other organometallic C-C bond formation	Chloro ketone Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Chloro
Other organometallic C-C bond formation	Chloro aldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Chloro
Friedel-Crafts reaction	Friedel-Crafts alkylation	C-C Bond Formation	Alkylation	Friedel-Crafts	
Other C-C bond formation	Hammick reaction	C-C Bond Formation	Coupling	Hammick	
Other C-C bond formation	Chloro Nierenstein reaction	C-C Bond Formation	Methyl insertion	Nierenstein	Chloro
Other C-C bond formation	Alkyne + formaldehyde reaction	C-C Bond Formation	Alkyne + formaldehyde		
Sonogashira reaction	Chloro Sonogashira coupling	C-C Bond Formation	Coupling	Sonogashira	Chloro
Other C-C bond formation	Claisen condensation	C-C Bond Formation	Condensation	Claisen	
Other organometallic C-C bond formation	Olefin metathesis	C-C Bond Formation	Olefin metathesis		
Other C-C bond formation	Knunyants fluoroalkylation	C-C Bond Formation	Alkylation	Knunyants	
Heck reaction	Bromo Heck reaction	C-C Bond Formation	Coupling	Heck	Bromo
Heck reaction	Chloro Heck-type reaction	C-C Bond Formation	Coupling	Heck	Chloro
Other C-C bond formation	Perkin condensation	C-C Bond Formation	Condensation	Cinnamic acid formation	Perkin
Other organometallic C-C bond formation	Cadiot-Chodkiewicz-type coupling	C-C Bond Formation	Coupling	Bisacetylene formation	Cadiot-Chodkiewicz

Other organometallic C-C bond formation	Bromo formaldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Bromo
Other Pd-catalyzed reactions (Negishi,Kumada,etc.)	Hiyama coupling	C-C Bond Formation	Coupling	Hiyama	
Other C-C bond formation	Blanc bromomethylation	C-C Bond Formation	Methylation	Blanc bromomethylation	
Other C-C bond formation	Fluoro Nierenstein reaction	C-C Bond Formation	Methyl insertion	Nierenstein	Fluoro
Sonogashira reaction	Iodo Sonogashira coupling	C-C Bond Formation	Coupling	Sonogashira	Iodo
Other organometallic C-C bond formation	Birch alkylation	C-C Bond Formation	Aromatic to alkylated cyclohexadiene	Birch	
Other organometallic C-C bond formation	Grignard Bouveault aldehyde synthesis	C-C Bond Formation	Aldehyde formation	Bouveault-Grignard	
Sonogashira reaction	Bromo Sonogashira coupling	C-C Bond Formation	Coupling	Sonogashira	Bromo
Heck reaction	Chloro Heck reaction	C-C Bond Formation	Coupling	Heck	Chloro
Other C-C bond formation	Baylis-Hillman reaction	C-C Bond Formation	Coupling	Baylis-Hillman	
Heck reaction	Iodo Heck reaction	C-C Bond Formation	Coupling	Heck	Iodo
Other organometallic C-C bond formation	Nickel Kumada coupling	C-C Bond Formation	Coupling	Kumada	
Other C-C bond formation	Wurtz coupling	C-C Bond Formation	Coupling	Wurtz-Fittig	
Friedel-Crafts reaction	Scholl reaction	C-C Bond Formation	Coupling	Scholl	
Heck reaction	Bromo Heck-type reaction	C-C Bond Formation	Coupling	Heck	Bromo
Heck reaction	Iodo Heck-type reaction	C-C Bond Formation	Coupling	Heck	Iodo
Other Pd-catalyzed reactions (Negishi,Kumada,etc.)	Negishi coupling	C-C Bond Formation	Coupling	Negishi	
Stille reaction	Bromo Stille reaction	C-C Bond Formation	Coupling	Stille	Bromo
Other organometallic C-C bond formation	McMurry coupling	C-C Bond Formation	Coupling	Carbonyl groups to alkene	McMurry
Suzuki coupling	Bromo Suzuki-type coupling	C-C Bond Formation	Coupling	Suzuki	Bromo
Other organometallic C-C bond formation	Weinreb bromo coupling	C-C Bond Formation	Amide to ketone	Weinreb Ketone reaction	
Stille reaction	Triflyloxy Stille reaction	C-C Bond Formation	Coupling	Stille	Tryflyloxy
Stille reaction	Chloro Stille reaction	C-C Bond Formation	Coupling	Stille	Chloro
Suzuki coupling	Iodo Suzuki coupling	C-C Bond Formation	Coupling	Suzuki	Iodo
Stille reaction	Iodo Stille reaction	C-C Bond Formation	Coupling	Stille	Iodo

Other organometallic C-C bond formation	Chloro formaldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Chloro
Suzuki coupling	Triflyloxy Suzuki coupling	C-C Bond Formation	Coupling	Suzuki	Triflyloxy
Other Pd-catalyzed reactions (Negishi,Kumada,etc.)	Palladium Kumada coupling	C-C Bond Formation	Coupling	Kumada	
Suzuki coupling	Iodo Suzuki-type coupling	C-C Bond Formation	Coupling	Suzuki	Iodo
Other organometallic C-C bond formation	Weinreb ketone synthesis	C-C Bond Formation	Amide to ketone	Weinreb Ketone reaction	
Suzuki coupling	Bromo Suzuki coupling	C-C Bond Formation	Coupling	Suzuki	Bromo
Other organometallic C-C bond formation	Cadiot-Chodkiewicz coupling	C-C Bond Formation	Coupling	Bisacetylene formation	Cadiot-Chodkiewicz
Other organometallic C-C bond formation	Bouveault aldehyde synthesis	C-C Bond Formation	Aldehyde formation	Bouveault	
Heck reaction	Triflyloxy Heck reaction	C-C Bond Formation	Coupling	Heck	Tryflyloxy
Sonogashira reaction	Triflyloxy Sonogashira coupling	C-C Bond Formation	Coupling	Sonogashira	Tryflyloxy
Suzuki coupling	Chloro Suzuki coupling	C-C Bond Formation	Coupling	Suzuki	Chloro
Suzuki coupling	Chloro Suzuki-type coupling	C-C Bond Formation	Coupling	Suzuki	Chloro
Other C-C bond formation	Barton-Kellogg olefination	C-C Bond Formation	Olefination	Barton-Kellogg	
Suzuki coupling	Triflyloxy Suzuki-type coupling	C-C Bond Formation	Coupling	Suzuki	Triflyloxy
Other functional group interconversion	Seyferth-Gilbert aldehyde reaction	C-C Bond Formation	Alkyne formation	Seyferth-Gilbert aldehyde	
Other C-C bond formation	Seyferth-Gilbert ketone homologation	C-C Bond Formation	Alkyne formation	Seyferth-Gilbert ketone homologation	
Other C-C bond formation	Aldehyde Hosomi-Sakurai reaction	C-C Bond Formation	Addition	Hosomi Sakurai	Aldehyde
Other organometallic C-C bond formation	Acyloin condensation	C-C Bond Formation	Coupling	Ester to alpha-hydroxyketone	
Heck reaction	Triflyloxy Heck-type reaction	C-C Bond Formation	Coupling	Heck	Tryflyloxy
Other organometallic C-C bond formation	Weinreb iodo coupling	C-C Bond Formation	Amide to ketone	Weinreb Ketone reaction	
Stille reaction	Stille-Kelly coupling	C-C Bond Formation	Coupling	Stille-Kelly	

Other organometallic C-C bond formation	Iodo formaldehyde Barbier reaction	C-C Bond Formation	Coupling	Barbier/Reformatsky	Iodo
Stille reaction	Stille reaction	C-C Bond Formation	Coupling	Stille	
Other Pd-catalyzed reactions (Negishi, Kumada, etc.)	Fukuyama coupling	C-C Bond Formation	Coupling	Fukuyama	
Other C-C bond formation	Koch reaction	C-C Bond Formation	Carboxylic acid formation	Koch	
Other organometallic C-C bond formation	Iodo Nozaki-Hiyama-Kishi reaction	C-C Bond Formation	Coupling	Nozaki-Hiyama-Kishi	
Other functional group interconversion	Ester hydrolysis	Cleavage	Hydrolysis	Ester	
Other reductions	Disulfide reduction	Cleavage	Reduction	Disulfide	
Alcohols to aldehydes	Periodate cleavage	Cleavage	Periodate		
Alkene oxidative cleavage	Alkene oxidative cleavage	Cleavage	Alkene oxidative		
Other functional group interconversion	Imine hydrolysis	Cleavage	Hydrolysis	Imine	
Other functional group interconversion	Nef reaction	Cleavage	Hydrolysis	Nitroalkane	
Alkene oxidative cleavage	Ozonolysis	Cleavage	Ozonolysis		
Other functional group interconversion	Iminium hydrolysis	Cleavage	Hydrolysis	Imine	
Other reductions	Mozingo ketone reduction	Cleavage	Reduction	Thioketal	
N-acylation to amide	N-acetylation	C-N Bond Formation	N-acetylation		
Other C-C bond formation	Strecker ketone reaction	C-N Bond Formation	Ketone + amine	Strecker	
N-acylation to amide	Amide Schotten-Baumann	C-N Bond Formation	Amide formation	Schotten-Baumann	
N-acylation to amide	Hydrazide Schotten-Baumann	C-N Bond Formation	Hydrazide formation	Schotten-Baumann	
N-acylation to amide	Carboxylic acid + hydrazine condensation	C-N Bond Formation	Condensation	Carboxylic acid + hydrazine	
N-acylation to amide	Carboxylic anhydride + amine reaction	C-N Bond Formation	Carboxylic anhydride + amine		
N-acylation to amide	Carboxylic acid + amine condensation	C-N Bond Formation	Condensation	Carboxylic acid + amine	
N-substitution with alkyl-X	Iodo N-methylation	C-N Bond Formation	N-methylation		
Heteroaryl N-alkylation	Chloro N-alkylation	C-N Bond Formation	N-alkylation	Chloro	
N-arylation with Ar-X	Fluoro N-arylation	C-N Bond Formation	N-arylation	Fluoro	

N-acylation to amide	Formic acid + amine condensation	C-N Bond Formation	Condensation	Carboxylic acid + amine	
N-acylation to amide	Carboxylic ester + amine reaction	C-N Bond Formation	Carboxylic ester + amine		
Carbamate/carbonate formation	Isocyanate + alcohol reaction	C-N Bond Formation	Carbamate formation	Isocyanate + alcohol	
N-acylation to urea	Isocyanate + amine urea coupling	C-N Bond Formation	Urea formation	Isocyanate + amine	
N-arylation with Ar-X	Chloro N-arylation	C-N Bond Formation	N-arylation	Chloro	
N-substitution with alkyl-X	N-methylation	C-N Bond Formation	N-methylation		
Heteroaryl N-alkylation	Bromo N-alkylation	C-N Bond Formation	N-alkylation	Bromo	
Reductive amination	Aldehyde reductive imination	C-N Bond Formation	Imination	Reductive	Aldehyde
N-acylation to amide	Weinreb amide synthesis	C-N Bond Formation	Amide formation	Weinreb	
Reductive amination	Ketone reductive amination	C-N Bond Formation	Amination	Reductive	Ketone
Reductive amination	Formaldehyde reductive amination	C-N Bond Formation	Amination	Reductive	
Other functional group addition	Amination	C-N Bond Formation	Amination		
N-acylation to urea	Isothiocyanate + amine thiourea coupling	C-N Bond Formation	Thiourea formation	Isothiocyanate + amine	
N-acylation to amide	Carboxylic ester + hydrazine reaction	C-N Bond Formation	Carboxylic ester + hydrazine		
Reductive amination	Ketone reductive imination	C-N Bond Formation	Imination	Reductive	Ketone
Heteroaryl N-alkylation	Mesyloxy N-alkylation	C-N Bond Formation	N-alkylation	Mesyloxy	
Other C-C bond formation	Strecker aldehyde reaction	C-N Bond Formation	Aldehyde + amine	Strecker	
N-acylation to amide	Carboxylic anhydride + sulfonamide reaction	C-N Bond Formation	Carboxylic anhydride + sulfonamide		
Reductive amination	Aldehyde reductive amination	C-N Bond Formation	Amination	Reductive	Aldehyde
N-substitution with alkyl-X	Menshutkin reaction	C-N Bond Formation	Tertiary amine to quaternary ammonium salt	Menshutkin	
Reductive amination	Alkylimino-de-oxo-bisubstitution	C-N Bond Formation	Imidation	Alkylimino-de-oxo-bisubstitution	
Heteroaryl N-alkylation	Iodo N-alkylation	C-N Bond Formation	N-alkylation	Iodo	

Reductive amination	Alcohol + amine condensation	C-N Bond Formation	Condensation	Alcohol + amine	
Amidine formation	Thioimidic ester + amine reaction	C-N Bond Formation	Guanidine formation	Thioimidic ester + amine	
Other C-C bond formation	Mannich reaction	C-N Bond Formation	Condensation	Multi-component	Mannich
Heteroaryl N-alkylation	Bromo Gabriel alkylation	C-N Bond Formation	N-alkylation	Bromo	Gabriel
N-arylation with Ar-X	Bromo N-arylation	C-N Bond Formation	N-arylation	Bromo	
Reductive amination	Eschweiler-Clarke methylation	C-N Bond Formation	Amination	Reductive	
Amidine formation	Imidic ester + amine reaction	C-N Bond Formation	Imide formation	Imidic ester + amine	
N-arylation with Ar-X	Iodo N-arylation	C-N Bond Formation	N-arylation	Iodo	
N-acylation to amide	Carboxylic anhydride + hydrazine reaction	C-N Bond Formation	Carboxylic anhydride + hydrazine		
N-arylation with Ar-X	Mesyl N-arylation	C-N Bond Formation	N-arylation	Mesyl	
N-arylation with Ar-X	Chichibabin amination	C-N Bond Formation	Amination	Chichibabin	
Reductive amination	Formaldehyde reductive imination	C-N Bond Formation	Imination	Reductive	
Heteroaryl N-alkylation	Iodo Gabriel alkylation	C-N Bond Formation	N-alkylation	Iodo	Gabriel
Heteroaryl N-alkylation	Fluoro N-alkylation	C-N Bond Formation	N-alkylation	Fluoro	
N-acylation to amide	Thioamide Schotten-Baumann	C-N Bond Formation	Thioamide formation	Schotten-Baumann	
Heteroaryl N-alkylation	Chloro Gabriel alkylation	C-N Bond Formation	N-alkylation	Chloro	Gabriel
N-acylation to amide	Carboxylic ester + sulfonamide reaction	C-N Bond Formation	Carboxylic ester + sulfonamide		
N-acylation to urea	Levy reaction	C-N Bond Formation	Urea formation	Thiocyanate + hydrazine	
N-acylation to amide	Carboxylic acid + sulfonamide condensation	C-N Bond Formation	Condensation	Carboxylic acid + sulfonamide	
N-acylation to amide	Alcohol Ritter reaction	C-N Bond Formation	Amide formation	Ritter	Alcohol
N-acylation to amide	Alkene Ritter reaction	C-N Bond Formation	Amide formation	Ritter	Alkene
Reductive amination	Dimethyl acetal reductive amination	C-N Bond Formation	Amination	Reductive	
N-arylation with Ar-X	Chloro Buchwald-Hartwig amination	C-N Bond Formation	Amination		
N-arylation with Ar-X	Triflyloxy N-arylation	C-N Bond Formation	N-arylation	Tryflyloxy	

Amidine formation	Thioimidic acid + amine reaction	C-N Bond Formation	Amidine formation	Thioimidic acid + amine	
N-arylation with Ar-X	Bromo Buchwald-Hartwig amination	C-N Bond Formation	Amination		
N-arylation with Ar-X	Triflyloxy Buchwald-Hartwig amination	C-N Bond Formation	Amination		
N-arylation with Ar-X	Iodo Buchwald-Hartwig amination	C-N Bond Formation	Amination		
N-acylation to amide	Ugi reaction	C-N Bond Formation	Condensation	Multi-component	Ugi
Other C-C bond formation	Petasis reaction	C-N Bond Formation	Multi-component	Petasis	
N-arylation with Ar-X	Chan-Lam arylamine coupling	C-N Bond Formation	Arylamine formation	Chan-Lam	
N-substitution with alkyl-X	Chan-Lam alkylamine coupling	C-N Bond Formation	Alkylamine formation	Chan-Lam	
O-sulfonylation	Sulfonic ester Schotten-Baumann	C-O Bond Formation	Sulphonic esterification	Schotten-Baumann	
O-acylation to ester	Fischer-Speier esterification	C-O Bond Formation	Esterification		
O-substitution	Williamson ether synthesis	C-O Bond Formation	Etherification	Williamson	
O-substitution	Methyl esterification	C-O Bond Formation	Esterification		
O-substitution	SNAr ether synthesis	C-O Bond Formation	Etherification	Nucleophilic aromatic substitution	
O-substitution	Ethyl esterification	C-O Bond Formation	Esterification		
O-substitution	Ullmann condensation	C-O Bond Formation	Etherification	Nucleophilic aromatic substitution	
O-acylation to ester	Esterification	C-O Bond Formation	Esterification		
O-acylation to ester	Ester Schotten-Baumann	C-O Bond Formation	Esterification	Schotten-Baumann	
O-substitution	Diazomethane esterification	C-O Bond Formation	Esterification	Diazomethane	
O-substitution	Alkene ether synthesis	C-O Bond Formation	Etherification	Alkene to ether	
O-acylation to ester	Steglich esterification	C-O Bond Formation	Esterification		
O-acylation to ester	Baeyer-Villiger oxidation	C-O Bond Formation	Esterification	Baeyer-Villiger oxidation	
O-substitution	O-methylation	C-O Bond Formation	O-methylation		
Carbamate/carbonate formation	Isothiocyanate + alcohol reaction	C-O Bond Formation	Thiocarbamic ester formation	Isothiocyanate + alcohol	
O-acylation to ester	Yamaguchi esterification	C-O Bond Formation	Esterification	Yamaguchi	
O-substitution	Chan-Lam ether coupling	C-O Bond Formation	Arylether formation	Chan-Lam	
N-acylation to amide	Passerini reaction	C-O Bond Formation	Multi-component	Passerini	

S-substitution	S-methylation	C-S Bond Formation	S-methylation		
Other acylation	Carboxylic acid + thiol condensation	C-S Bond Formation	Condensation	Carboxylic acid + thiol	
S-substitution	Migita thioether synthesis	C-S Bond Formation	Thioether formation	Migita	
N-containing heterocycle formation	Phillips benzimidazole condensation	Cyclization	Benzimidazole formation	Phillips condensation	
Other C-C bond formation	Dieckmann condensation	Cyclization	beta-keto esters formation	Dieckmann condensation	
N-containing heterocycle formation	Borsche-Drechsel carbazole synthesis	Cyclization	Tetrahydrocarbazoles formation	Borsche-Drechsel	
O-containing heterocycle formation	Oxa-Diels-Alder reaction	Cyclization	Diene + dienophile	Diels-Alder	Oxa
N-containing heterocycle formation	Pictet-Spengler reaction	Cyclization	Pictet-Spengler	beta-arylamine	
N-containing heterocycle formation	Boennemann cyclization	Cyclization	Pyridine formation	Bonnemann	
O-containing heterocycle formation	Oxa Pictet-Spengler reaction	Cyclization	Pictet-Spengler	Oxa	
Other C-C bond formation	Robinson annulation	Cyclization	Polycyclic compound	Robinson annulation	
N-containing heterocycle formation	Knorr quinoline cyclization	Cyclization	Quinoline formation	Knorr	
S-containing heterocycle formation	Dithiane Gewald reaction	Cyclization	2-amino-thiophene formation	Gewald	
N-containing heterocycle formation	Wenker synthesis	Cyclization	Aziridine formation	Wenker	
O-containing heterocycle formation	Iodolactonization	Cyclization	Iodolactonization		
Other C-C bond formation	Nazarov cyclization	Cyclization	Cyclopentenone formation	Nazarov	
N-containing heterocycle formation	Aza-Diels-Alder reaction	Cyclization	Diene + dienophile	Diels-Alder	Aza
S-containing heterocycle formation	Gewald reaction	Cyclization	2-amino-thiophene formation	Gewald	
O-containing heterocycle formation	Yamaguchi lactonization	Cyclization	Lactonization	Yamaguchi	
N-containing heterocycle formation	Pictet-Spengler cyclization	Cyclization	Pictet-Spengler	beta-arylamine	

O-containing heterocycle formation	Bromolactonization	Cyclization	Bromolactonization		
Other C-C bond formation	Diels-Alder cycloaddition	Cycloaddition	Diene + dienophile	Diels-Alder	
N-containing heterocycle formation	Azide-nitrile Huisgen cycloaddition	Cycloaddition	Azide + nitrile	Huisgen	
Other C-C bond formation	Triple bond Diels-Alder	Cycloaddition	Diene + dienophile	Diels-Alder	
N-containing heterocycle formation	Azide-alkyne Huisgen cycloaddition	Cycloaddition	Azide + terminal alkyne	Huisgen	
RCO ₂ H deprotections	CO ₂ H-Et deprotection	Deprotection	COO-Ethyl	COO-Et	
ROH deprotections	O-Ac deprotection	Deprotection	O-Acetyl	O-Ac	
ROH deprotections	O-Bn deprotection	Deprotection	O-Benzyl	O-Bn	
NH deprotections	N-Ac deprotection	Deprotection	N-Acetyl	N-Ac	
RCO ₂ H deprotections	CO ₂ H-Me deprotection	Deprotection	COO-Methyl	COO-Me	
NH deprotections	N-Bz deprotection	Deprotection	N-Benzoyl	N-Bz	
NH deprotections	N-Bn deprotection	Deprotection	N-Benzyl	N-Bn	
RSH deprotections	S-carbonyl deprotection	Deprotection	S-carbonyl		
RCO ₂ H deprotections	CO ₂ H-tBu deprotection	Deprotection	COO-t-Buthyl	COO-tBu	
ROH deprotections	O-THP deprotection	Deprotection	O-Tetrahydropyranyl	O-THP	
NH deprotections	N-Cbz deprotection	Deprotection	N-Carbobenzyloxy	N-Cbz	
NH deprotections	N-Boc deprotection	Deprotection	N-t-Butyloxycarbonyl	N-Boc	
Other deprotections	Aldehyde acetal deprotection	Deprotection	Aldehyde acetal		
NH deprotections	N-Phth deprotection	Deprotection	N-Phthalimide	N-Phth	
Other deprotections	Ketone dioxolane deprotection	Deprotection	Ketone dioxolane		
Other deprotections	Ketone ketal deprotection	Deprotection	Ketone ketal		
ROH deprotections	O-TMS deprotection	Deprotection	O-Trimethylsilyl	O-TMS	
NH deprotections	N-Benzylidene deprotection	Deprotection	N-Benzylidene		
Other deprotections	Alkyne TMS deprotection	Deprotection	Trimethylsilane	TMS	
NH deprotections	N-TFA deprotection	Deprotection	N-Trifluoroacetyl	N-TFA	
Other deprotections	Ketone dithiane deprotection	Deprotection	Ketone dithiane		
Other deprotections	Aldehyde dithiolane deprotection	Deprotection	Aldehyde dithiolane		
Other deprotections	Ketone dithiolane deprotection	Deprotection	Ketone dithiolane		

NH deprotections	N-PMB deprotection	Deprotection	N-p-Methoxybenzyl	N-PMB	
ROH deprotections	O-TBS deprotection	Deprotection	O-t-Butyldimethylsilyl	O-TBS	
ROH deprotections	O-MOM deprotection	Deprotection	O-Methoxymethyl	O-MOM	
ROH deprotections	Silyl ether deprotection	Deprotection	Silyl ether		
Other deprotections	Aldehyde dioxolane deprotection	Deprotection	Aldehyde dioxolane		
NH deprotections	N-THP deprotection	Deprotection	N-Tetrahydropyranyl	N-THP	
Other deprotections	Aldehyde dioxane deprotection	Deprotection	Aldehyde dioxane		
Other deprotections	Aldehyde dithiane deprotection	Deprotection	Aldehyde dithiane		
ROH deprotections	O-TIPS deprotection	Deprotection	O-Triisopropylsilyl	O-TIPS	
NH deprotections	N-Benzhydrylidene deprotection	Deprotection	N-Benzhydrylidene		
Other deprotections	Ketone dioxane deprotection	Deprotection	Ketone dioxane		
Nitro to amine reduction	Nitro to amino	Functional Conversion	Nitro to amino		
Alkene to alkane	Alkene hydrogenation	Functional Conversion	Hydrogenation	Alkene to alkane	
Other functional group interconversion	Chloro to bromo	Functional Conversion	Chloro to bromo		
Acid to acid chloride	Carboxylic acid to acid chloride	Functional Conversion	Carboxylic acid to acid chloride		
Other functional group interconversion	Methylsulfanyl to hydrazino	Functional Conversion	Methylsulfanyl to hydrazino		
Dehydration	Alcohol elimination	Functional Conversion	Alcohol to alkene		
Amide to amine reduction	Amide to amine reduction	Functional Conversion	Reduction	Amide to amine	
Oxidations at sulfur	Sulfanyl to sulfonyl	Functional Conversion	Sulfanyl to sulfonyl		
Other functional group interconversion	Amino to cyano	Functional Conversion	Amino to cyano		
Other functional group interconversion	Amino to chloro	Functional Conversion	Amino to chloro		
Other functional group interconversion	Chloro Kolbe nitrile synthesis	Functional Conversion	Chloro to nitrile		
Other functional group interconversion	Cyano to carbamoyl	Functional Conversion	Cyano to carbamoyl		

Other functional group interconversion	Cyano to thiocarbamoyl	Functional Conversion	Cyano to thiocarbamoyl		
Oxidations at sulfur	Sulfanyl to sulfinyl	Functional Conversion	Sulfanyl to sulfinyl		
Other oxidations	Methyl to formyl	Functional Conversion	Methyl to formyl		
Other functional group interconversion	Pyridone to chloropyridine	Functional Conversion	Pyridone to chloropyridine		
O-substitution	Hydroxy to methoxy	Functional Conversion	Hydroxy to methoxy		
Other functional group interconversion	Amino to hydroxy	Functional Conversion	Amino to hydroxy		
Acid to acid chloride	Acid to acid chloride	Functional Conversion	Carboxylic acid to acid chloride		
Cyano or imine to amine	Nitrile reduction	Functional Conversion	Reduction	Nitrile to amino	
Ketone to alcohol	Ketone to alcohol reduction	Functional Conversion	Reduction	Aldehyde/ketone to alcohol	
Alcohol to halide	Hydroxy to chloro	Functional Conversion	Hydroxy to chloro		
Other reductions	Pyridine to piperidine hydrogenation	Functional Conversion	Hydrogenation	Pyridine to piperidine	
Alcohols to aldehydes	Bromo to oxo oxidation	Functional Conversion	Oxidation	Bromo to oxo	
Other functional group interconversion	Oxo to hydroxyimino	Functional Conversion	Oxo to hydroxyimino		
Dehydration	Hydroxyiminomethyl to cyano	Functional Conversion	Hydroxyiminomethyl to cyano		
Alcohols to aldehydes	Alcohol to ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Nitrile to acid	Cyano to carboxy	Functional Conversion	Cyano to carboxy		
Other functional group interconversion	Amino to isocyanato	Functional Conversion	Amino to isocyanato		
Other functional group interconversion	Amino to isothiocyanato	Functional Conversion	Amino to isothiocyanato		
Other functional group interconversion	Bromo Kolbe nitrile synthesis	Functional Conversion	Bromo to nitrile		
Other reductions	Ketone to alkane reduction	Functional Conversion	Reduction	Aldehyde/ketone to alkane	
Other functional group interconversion	Bromo to carboxy	Functional Conversion	Bromo to carboxy		
ROH deprotections	Methoxy to hydroxy	Functional Conversion	Methoxy to hydroxy		
Alcohols to aldehydes	Jones ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Hydroxyimino to oxo	Functional Conversion	Hydroxyimino to oxo		

Other functional group interconversion	Hydroxy to amino	Functional Conversion	Hydroxy to amino		
Alcohols to acids	Alcohol to acid oxidation	Functional Conversion	Oxidation	Alcohol to acid	
Other reductions	Alkyne to alkene hydrogenation	Functional Conversion	Hydrogenation	Alkyne to alkene	
O-containing heterocycle formation	Prilezhaev epoxidation	Functional Conversion	Alkene to epoxide	Prilezhaev	
Alcohol to halide	Hydroxy to bromo	Functional Conversion	Hydroxy to bromo		
Other functional group interconversion	Bromo to iodo Finkelstein reaction	Functional Conversion	Bromo to iodo		
Other functional group interconversion	Chloro to sulfanyl	Functional Conversion	Chloro to sulfanyl		
Other functional group interconversion	Bromo to cyano	Functional Conversion	Bromo to cyano		
Other functional group interconversion	Carboxy ester to carbamoyl	Functional Conversion	Carboxy ester to carbamoyl		
Other functional group interconversion	Chlorosulfonyl to sulfamoyl	Functional Conversion	Chlorosulfonyl to sulfamoyl		
Acid to acid chloride	Sulfo to chlorosulfonyl	Functional Conversion	Sulfo to chlorosulfonyl		
Other reductions	Carboxylic acid to alcohol reduction	Functional Conversion	Reduction	Carboxylic acid to alcohol	
O-acylation to ester	Hydroxy to acetoxy	Functional Conversion	Hydroxy to acetoxy		
Alkyne to alkane	Alkyne to alkane hydrogenation	Functional Conversion	Hydrogenation	Alkyne to alkane	
Alcohols to aldehydes	Collins ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Azido to amino	Functional Conversion	Azido to amino		
Alkene oxidative cleavage	Lemieux-Johnson oxidation	Functional Conversion	Oxidation	Alkene to aldehyde/ketone	
Other functional group interconversion	Cyano to formyl	Functional Conversion	Cyano to formyl		
Other reductions	Wolff-Kishner reduction	Functional Conversion	Reduction	Aldehyde/ketone to alkane	
Other oxidations	Delepine aldehyde oxidation	Functional Conversion	Oxidation	Aldehyde to acid	
N-acylation to urea	Amino to ureido	Functional Conversion	Amino to ureido		
Alkene oxidation	Ethenyl to acetyl	Functional Conversion	Oxidation	Alkene to aldehyde/ketone	
Ester to alcohol	Ester to alcohol reduction	Functional Conversion	Reduction	Ester to alcohol	

Other functional group interconversion	Chloro to iodo Finkelstein reaction	Functional Conversion	Chloro to iodo		
Other functional group interconversion	Bromo to hydroxy	Functional Conversion	Bromo to hydroxy		
Other functional group interconversion	Chloro to fluoro	Functional Conversion	Chloro to fluoro		
Other functional group interconversion	Chloro to amino	Functional Conversion	Chloro to amino		
Other functional group interconversion	Amino to hydrazino	Functional Conversion	Amino to hydrazino		
Other functional group interconversion	Oxo to thioxo	Functional Conversion	Oxo to thioxo		
Dehydration	Carbamoyl to cyano	Functional Conversion	Carbamoyl to cyano		
Other functional group interconversion	Chloro to hydroxy	Functional Conversion	Chloro to hydroxy		
Other functional group interconversion	Chlorocarbonyl to carbamoyl	Functional Conversion	Chlorocarbonyl to carbamoyl		
N-acylation to amide	Ketone Schmidt reaction	Functional Conversion	Ketone to amide	Schmidt	
Other functional group interconversion	Carboxy to carbamoyl	Functional Conversion	Carboxy to carbamoyl		
Alcohols to aldehydes	Alcohol to aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Chloro to isothiocyanato	Functional Conversion	Chloro to isothiocyanato		
Alcohols to aldehydes	Sarett ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other reductions	Nitrosamine to hydrazine reduction	Functional Conversion	Reduction	Nitrosamine to hydrazine	
Oxidations at sulfur	Sulfinyl to sulfonyl	Functional Conversion	Sulfinyl to sulfonyl		
Other functional group interconversion	Formyl to cyano	Functional Conversion	Formyl to cyano		
Other reductions	Birch reduction	Functional Conversion	Reduction	Aromatic to cyclohexadiene	Birch
Alcohols to aldehydes	Cornforth ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Nitro to hydroxyamino	Functional Conversion	Nitro to hydroxyamino		
Ester to alcohol	Bouveault-Blanc reduction	Functional Conversion	Reduction	Ester to alcohol	

Other functional group interconversion	Chlorosulfonyl to sulfanyl	Functional Conversion	Chlorosulfonyl to sulfanyl		
Other functional group interconversion	Hydrazino to amino	Functional Conversion	Hydrazino to amino		
Other functional group interconversion	Amino to fluoro	Functional Conversion	Amino to fluoro		
Other functional group interconversion	Amino to bromo	Functional Conversion	Amino to bromo		
Other functional group interconversion	Amino to iodo	Functional Conversion	Amino to iodo		
Other functional group interconversion	Fluoro to chloro	Functional Conversion	Fluoro to chloro		
Other functional group interconversion	Amino to azido	Functional Conversion	Amino to azido		
Other functional group interconversion	Chloro to azido	Functional Conversion	Chloro to azido		
Other oxidations	Aldehyde to acid oxidation	Functional Conversion	Oxidation	Aldehyde to acid	
Other functional group interconversion	Bromo to amino	Functional Conversion	Bromo to amino		
Other functional group interconversion	Amino to sulfanyl	Functional Conversion	Amino to sulfanyl		
Other functional group interconversion	Diazo to chloro	Functional Conversion	Diazo to chloro		
Other functional group interconversion	Corey-Fuchs reaction step 1	Functional Conversion	Aldehyde to dibromoalkene		
Other functional group interconversion	Corey-Fuchs reaction step 2	Functional Conversion	Dibromoalkene to alkyne		
Other reductions	Aldehyde to alcohol reduction	Functional Conversion	Reduction	Aldehyde to alcohol	
Other functional group interconversion	Amino to chlorosulfonyl	Functional Conversion	Amino to chlorosulfonyl		
N-acylation to amide	Amino to formamido	Functional Conversion	Amino to formamido		
Salt formation	Trifluoroacetate salt formation	Functional Conversion	Hydroxy to trifluoroacetate salt		
Other functional group interconversion	Cyano to amidino	Functional Conversion	Cyano to amidino		

Other reductions	Aldehyde to alkane reduction	Functional Conversion	Reduction	Aldehyde to alkane	
Other functional group interconversion	Amino to guanidino	Functional Conversion	Amino to guanidino		
Alcohol to halide	Hydroxy to fluoro	Functional Conversion	Hydroxy to fluoro		
Other functional group interconversion	Diazonio to hydroxy	Functional Conversion	Diazo to hydroxy		
Ketone to alcohol	Meerwein-Ponndorf-Verley reduction	Functional Conversion	Reduction	Ketone to alcohol	
Other functional group interconversion	Chlorosulfonyl to sulfino	Functional Conversion	Chlorosulfonyl to sulfino		
Other functional group interconversion	Chlorosulfonyl to sulfo	Functional Conversion	Chlorosulfonyl to sulfo		
Other reductions	Pyrazine to piperazine hydrogenation	Functional Conversion	Reduction	Pyrazine to piperazine	
Other functional group interconversion	Chloro to hydrazino	Functional Conversion	Chloro to hydrazino		
Other functional group interconversion	Nitro to fluoro	Functional Conversion	Nitro to fluoro		
Other functional group interconversion	Chloro to cyano	Functional Conversion	Chloro to cyano		
Alcohols to aldehydes	Oppenauer oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Cyano or imine to amine	Secondary aldimine reduction	Functional Conversion	Reduction	Imine to amine	
Other functional group interconversion	Bromo to formyl	Functional Conversion	Bromo to formyl		
Salt formation	Lithium salt formation	Functional Conversion	Hydroxy to lithium salt		
Other functional group interconversion	Hydroxy to sulfanyl	Functional Conversion	Hydroxy to sulfanyl		
Other functional group interconversion	Chloro Grignard preparation	Functional Conversion	Chloro to MgCl	Grignard preparation	
Cyano or imine to amine	Secondary ketimine reduction	Functional Conversion	Reduction	Imine to amine	
Other functional group interconversion	Bromo to iodo	Functional Conversion	Bromo to iodo		

Other functional group interconversion	Amino to sulfo	Functional Conversion	Amino to sulfo		
Other functional group interconversion	Chloro to mesyl	Functional Conversion	Chloro to mesyl		
Alkene oxidation	Alkene oxidation	Functional Conversion	Oxidation	Alkene to aldehyde/ketone	
Other functional group interconversion	Mesyl to cyano	Functional Conversion	Mesyl to cyano		
Salt formation	Acetate salt formation	Functional Conversion	Oxidation	Ketone to carboxylic acid	
Other functional group interconversion	Fluoro to hydroxy	Functional Conversion	Fluoro to hydroxy		
Other functional group interconversion	Amino to diazonio	Functional Conversion	Amino to diazonio		
Other functional group interconversion	Chloro to methoxy	Functional Conversion	Chloro to methoxy		
Alcohols to aldehydes	Sarett aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Bromo to fluoro	Functional Conversion	Bromo to fluoro		
Other functional group interconversion	Formamido to isociano	Functional Conversion	Formamido to isociano		
Other functional group interconversion	Chloro to carboxy	Functional Conversion	Chloro to carboxy		
Other functional group interconversion	Iodo to cyano	Functional Conversion	Iodo to cyano		
Alcohol to halide	Hydroxy to iodo	Functional Conversion	Hydroxy to iodo		
Other functional group interconversion	Carboxylic acid Schmidt reaction	Functional Conversion	Carboxylic acid to amine	Schmidt	
Other functional group interconversion	Sulfoxy to hydroxy	Functional Conversion	Sulfoxy to hydroxy		
Other functional group interconversion	Bromo Grignard preparation	Functional Conversion	Bromo to MgBr	Grignard preparation	
Alcohols to aldehydes	Collins aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Carboxy to bromo	Functional Conversion	Carboxy to bromo		
Alkene oxidative cleavage	Ethenyl to formyl	Functional Conversion	Oxidation	Alkene to aldehyde/ketone	
Other reductions	Nitroso to amino reduction	Functional Conversion	Reduction	Nitroso to amino	
Oxidations at nitrogen	Amino to nitro	Functional Conversion	Amino to nitro		

Other functional group interconversion	Bromo to azido	Functional Conversion	Bromo to azido		
Other functional group interconversion	Hydroxy to mesyloxy	Functional Conversion	Hydroxy to mesyloxy		
Other functional group interconversion	Rosenmund van Braun cyanation	Functional Conversion	Bromo to cyano		
Other functional group interconversion	Chloro to iodo	Functional Conversion	Chloro to iodo		
Other functional group interconversion	Carboxy to carbonazidoyl	Functional Conversion	Carboxy to carbonazidoyl		
Other reductions	Phosphoryl deoxygenation	Functional Conversion	Reduction	Phosphoryl deoxygenation	
N-acylation to amide	Carboxy to imidazolecarbonyl	Functional Conversion	Carboxy to imidazolecarbonyl		
Other functional group interconversion	Hydroxy to azido	Functional Conversion	Hydroxy to azido		
Alcohols to aldehydes	Jones aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Isocyanato to amino	Functional Conversion	Isocyanato to amino		
Alcohols to acids	Jones acid oxidation	Functional Conversion	Oxidation	Alcohol to acid	
Other functional group interconversion	Amino to thiocyanato	Functional Conversion	Amino to thiocyanato		
Other functional group interconversion	Amino to mesylamino	Functional Conversion	Amino to mesylamino		
Other functional group interconversion	Bromo to sulfanyl	Functional Conversion	Bromo to sulfanyl		
Other functional group addition	Milas hydroxylation	Functional Conversion	Alkene to diol	Milas	
Cyano or imine to amine	Primary ketimine reduction	Functional Conversion	Reduction	Imine to amine	
Other functional group interconversion	Fluoro to amino	Functional Conversion	Fluoro to amino		
N-acylation to urea	Amino to thioureido	Functional Conversion	Amino to thioureido		
Other functional group interconversion	Chloro to sulfo	Functional Conversion	Chloro to sulfo		
Other functional group interconversion	Oxo to difluoro	Functional Conversion	Oxo to difluoro		
N-acylation to amide	Imidazolecarbonyl to amide	Functional Conversion	Imidazolecarbonyl to amide		

Other functional group interconversion	Amino to isociano	Functional Conversion	Amino to isociano		
Other functional group interconversion	Staudinger reduction	Functional Conversion	Reduction	Azide to amine	
Oxidations at sulfur	Sulfinimidoyl to sulfonimidoyl	Functional Conversion	Sulfinimidoyl to sulfonimidoyl		
Other functional group interconversion	Chlorosulfonyl to fluorosulfonyl	Functional Conversion	Chlorosulfonyl to fluorosulfonyl		
Other functional group interconversion	Mesyloxy to hydroxy	Functional Conversion	Mesyloxy to hydroxy		
O-substitution	Hydroxy to triflyloxy	Functional Conversion	Hydroxy to triflyloxy		
Other functional group interconversion	Hofmann reaction	Functional Conversion	Amide to degraded primary amine	Hofmann	
Other reductions	Clemmensen reduction	Functional Conversion	Reduction	Aldehyde/ketone to alkane	
Other functional group interconversion	Sulfanyl to chlorosulfonyl	Functional Conversion	Sulfanyl to chlorosulfonyl		
Other functional group interconversion	Bromo Gabriel synthesis	Functional Conversion	Bromo to amino	Gabriel	
Other functional group interconversion	Fluoro to cyano	Functional Conversion	Fluoro to cyano		
Other functional group interconversion	Iodo to sulfanyl	Functional Conversion	Iodo to sulfanyl		
Other functional group interconversion	Chloro Gabriel synthesis	Functional Conversion	Chloro to amino	Gabriel	
Other functional group interconversion	Iodo Kolbe nitrile synthesis	Functional Conversion	Iodo to nitrile		
N-acylation to amide	Nitro to formamido	Functional Conversion	Nitro to formamido		
Alcohols to aldehydes	Cornforth aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Iodo to hydroxy	Functional Conversion	Iodo to hydroxy		
Other functional group interconversion	Balz-Schiemann reaction	Functional Conversion	Azo to fluoro	Balz-Schiemann	
Other functional group interconversion	Iodo to azido	Functional Conversion	Iodo to azido		

Other functional group interconversion	Oxo to cyano	Functional Conversion	Oxo to cyano		
Other oxidations	Fleming-Tamao oxidation	Functional Conversion	Silyl to hydroxy	Fleming-Tamao	
Other functional group interconversion	Diazo to bromo	Functional Conversion	Diazo to bromo		
Other functional group interconversion	Corey-Fuchs reaction	Functional Conversion	Aldehyde to alkyne	Corey-Fuchs	
Other functional group interconversion	Bromo to methoxy	Functional Conversion	Bromo to methoxy		
Other functional group interconversion	Sandmeyer bromination	Functional Conversion	Amino to bromo		
Other functional group interconversion	Chlorosulfinyl to sulfinamoyl	Functional Conversion	Chlorocarbonyl to sulfinamoyl		
Other functional group interconversion	Fluoro to azido	Functional Conversion	Fluoro to azido		
Other functional group interconversion	Chloro to nitro	Functional Conversion	Chloro to nitro		
Other functional group interconversion	Fluoro to hydrazino	Functional Conversion	Fluoro to hydrazino		
O-acylation to ester	Imidazolecarbonyl to ester	Functional Conversion	Imidazolecarbonyl to ester		
Other functional group interconversion	Bromo to hydrazino	Functional Conversion	Bromo to hydrazino		
Other functional group interconversion	Bromo to thiocyanato	Functional Conversion	Bromo to thiocyanato		
Other functional group interconversion	Mesyloxy to cyano	Functional Conversion	Mesyloxy to cyano		
Other functional group interconversion	Borono to pinacolatoboranyl	Functional Conversion	Borono to pinacolatoboranyl		
N-acylation to amide	Amino to imidazolecarboxamido	Functional Conversion	Amino to imidazolecarboxamido		
Other functional group interconversion	Iodo to carboxy	Functional Conversion	Iodo to carboxy		
Other functional group interconversion	Formyl to ethynyl	Functional Conversion	Formyl to ethynyl		
Other functional group interconversion	Van Leusen reaction	Functional Conversion	Ketone to nitrile	Val Leusen	

Other functional group interconversion	Thiocyanato to sulfanyl	Functional Conversion	Thiocyanato to sulfanyl		
Alcohols to aldehydes	Oppenauer-Woodward oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Iodo to formyl	Functional Conversion	Iodo to formyl		
Other functional group interconversion	Fluoro to bromo	Functional Conversion	Fluoro to bromo		
Other functional group interconversion	Bromo to borono	Functional Conversion	Bromo to borono		
Salt formation	Sodium salt formation	Functional Conversion	Hydroxy to sodium salt		
Alcohols to aldehydes	Dess-Martin aldehyde oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Salt formation	Potassium salt formation	Functional Conversion	Hydroxy to potassium salt		
Salt separation	Potassium salt separation	Functional Conversion	Potassium salt to carboxylic acid		
Other functional group interconversion	Fluoro to methoxy	Functional Conversion	Fluoro to methoxy		
Other functional group interconversion	Nitro to hydrazino	Functional Conversion	Nitro to hydrazino		
Other functional group interconversion	Iodo to amino	Functional Conversion	Iodo to amino		
Other functional group interconversion	Bromo to mesyl	Functional Conversion	Bromo to mesyl		
Alcohols to aldehydes	Dess-Martin ketone oxidation	Functional Conversion	Oxidation	Alcohol to aldehyde/ketone	
Other functional group interconversion	Iodo to methylsulfanyl	Functional Conversion	Iodo to methylsulfanyl		
Other functional group interconversion	Bromo to pinacolatoboranyl	Functional Conversion	Bromo to pinacolatoboranyl		
Other functional group interconversion	Borono to hydroxy	Functional Conversion	Borono to hydroxy		
Ketone to alcohol	Corey-Itsuno reduction	Functional Conversion	Reduction	Ketone to alcohol	
O-acylation to ester	Hydroxy to imidazolecarbonyloxy	Functional Conversion	Hydroxy to imidazolecarbonyloxy		

Other functional group interconversion	Fluoro to sulfanyl	Functional Conversion	Fluoro to sulfanyl		
Other reductions	Diazonio to hydrazino reduction	Functional Conversion	Reduction	Diazo to hydrazino	
Other functional group interconversion	Pinacolatoboranyl to borono	Functional Conversion	Pinacolatoboranyl to borono		
Salt separation	Sodium salt separation	Functional Conversion	Sodium salt to carboxylic acid		
Other functional group interconversion	Sandmeyer iodination	Functional Conversion	Amino to iodo		
Other functional group interconversion	Chloro to thiocyanato	Functional Conversion	Chloro to thiocyanato		
Other functional group interconversion	Iodo to borono	Functional Conversion	Iodo to borono		
Other functional group interconversion	Cyano to hydroxy	Functional Conversion	Cyano to hydroxy		
Other functional group interconversion	Fluoro to iodo	Functional Conversion	Fluoro to iodo		
Other functional group interconversion	Seyferth-Gilbert-Bestmann aldehyde reaction	Functional Conversion	Reduction	Aldehyde/ketone to alkyne	Seyferth-Gilbert homologation
Other functional group interconversion	Bromo Miyaura boration	Functional Conversion	Borylation	Miyaura	
Alkene oxidation	Wacker-Tsuji oxidation	Functional Conversion	Oxidation	Alkene to aldehyde/ketone	
Other functional group interconversion	Bromo Hunsdiecker reaction	Functional Conversion	Carboxylic acid to degraded bromo	Hunsdiecker	
Other functional group interconversion	Triflyloxy Miyaura boration	Functional Conversion	Borylation	Miyaura	
Other functional group interconversion	Iodo to pinacolatoboranyl	Functional Conversion	Iodo to pinacolatoboranyl		
Other functional group interconversion	Iodo Miyaura boration	Functional Conversion	Borylation	Miyaura	
Other functional group interconversion	Triflyloxy to pinacolatoboranyl	Functional Conversion	Triflyloxy to pinacolatoboranyl		
Other functional group interconversion	Bromo Negishi preparation	Functional Conversion	Bromo to ZnBr	Negishi preparation	

Other functional group interconversion	Chloro Miyaura boration	Functional Conversion	Borylation	Miyaura	
Other functional group interconversion	Hydrazino to bromo	Functional Conversion	Hydrazino to bromo		
Other functional group interconversion	Zincke nitration	Functional Conversion	Nitration		
Other functional group interconversion	Chloro to pinacolatoboranyl	Functional Conversion	Chloro to pinacolatoboranyl		
Other functional group interconversion	Fluoro Gabriel synthesis	Functional Conversion	Fluoro to amino	Gabriel	
Ketone to alcohol	Noyori asymmetric hydrogenation	Functional Conversion	Reduction	Ketone to alcohol	
Other functional group interconversion	Iodo to mesyl	Functional Conversion	Iodo to mesyl		
Other functional group interconversion	Iodo to hydrazino	Functional Conversion	Iodo to hydrazino		
Other functional group interconversion	Pinacolatoboranyl to bromo	Functional Conversion	Pinacolatoboranyl to bromo		
Other functional group interconversion	Iodo Gabriel synthesis	Functional Conversion	Iodo to amino	Gabriel	
Other functional group interconversion	Hofmann rearrangement	Functional Conversion	Amide to degraded primary amine	Hofmann	
Salt separation	Lithium salt separation	Functional Conversion	Lithium salt to carboxylic acid		
Other functional group interconversion	Diazonio to iodo	Functional Conversion	Diazo to iodo		
Other functional group interconversion	Iodo Hunsdiecker reaction	Functional Conversion	Carboxylic acid to degraded iodo	Hunsdiecker	
Salt formation	Chloride salt formation	Functional Elimination	Dechlorination		
Other functional group interconversion	Debromination	Functional Elimination	Debromination		
Other functional group interconversion	Dechlorination	Functional Elimination	Dechlorination		
Other functional group interconversion	Deoxygenation	Functional Elimination	Deoxygenation		

Other functional group interconversion	Decarboxylation	Functional Elimination	Decarboxylation		
Other functional group interconversion	Krapcho decarboxylation	Functional Elimination	Decarboxylation	Krapcho	
Other functional group interconversion	Decarbonylation	Functional Elimination	Decarbonylation		
Salt formation	Bromide salt formation	Functional Elimination	Debromination		
Other functional group interconversion	Deiodination	Functional Elimination	Deiodination		
Other functional group interconversion	Defluorination	Functional Elimination	Defluorination		
Other functional group interconversion	Cope elimination	Functional Elimination	N-oxide to alkene and hydroxylamine	Cope	
Halogenation	Bromination	Functional Introduction	Bromination		
Oxidations at nitrogen	Nitrogen oxidation	Functional Introduction	Hydroxylation	Nitrogen to hydroxylamine	
Nitration	Nitration	Functional Introduction	Nitration		
Halogenation	Wohl-Ziegler bromination	Functional Introduction	Bromination		
Other functional group addition	Alkene dihydroxylation	Functional Introduction	Hydroxylation	Alkene dihydroxylation	
Oxidations at nitrogen	Tertiary amine oxidation	Functional Introduction	Hydroxylation	Tertiary amine to hydroxylamine	
Halogenation	Chlorination	Functional Introduction	Chlorination		
Other functional group addition	Alkene hydration	Functional Introduction	Hydroxylation	Alkene hydration	
Other functional group addition	Carboxylation	Functional Introduction	Carboxylation		
Other functional group addition	Formylation	Functional Introduction	Formylation		
Sulfonation	Chlorosulfonation	Functional Introduction	Chlorosulfonation		
Other organometallic C-C bond formation	Lithium Bouveault aldehyde synthesis	Functional Introduction	Formylation	Bouveault	
Sulfonation	Sulfonation	Functional Introduction	Sulfonylation		
Halogenation	Iodination	Functional Introduction	Iodination		
Other C-C bond formation	Vilsmeier-Haack reaction	Functional Introduction	Formylation	Vilsmeier-Haack	
Halogenation	Alkene hydrobromination	Functional Introduction	Hydrobromination		
Halogenation	Fluorination	Functional Introduction	Fluorination		

Halogenation	Alkene bromination	Functional Introduction	Bromination		
Halogenation	Alkene hydrochlorination	Functional Introduction	Hydrochlorination		
Halogenation	Alkene chlorination	Functional Introduction	Chlorination		
O-acylation to ester	Chloro alkoxycarbonylation	Functional Introduction	Alkoxycarbonylation	Chloro	
Other organometallic C-C bond formation	Hydroformylation	Functional Introduction	Hydroformylation		
Alcohol to halide	Appel bromination	Functional Introduction	Bromination		
Alcohol to halide	Appel chlorination	Functional Introduction	Chlorination		
Other C-C bond formation	Kolbe-Schmitt reaction	Functional Introduction	Carboxylation		
O-acylation to ester	Iodo alkoxycarbonylation	Functional Introduction	Alkoxycarbonylation	Iodo	
O-acylation to ester	Bromo alkoxycarbonylation	Functional Introduction	Alkoxycarbonylation	Bromo	
Halogenation	Sulfur chlorination	Functional Introduction	Chlorination	Sulphur	
Other heteroatom alkylation/arylation	Hydrostannylation	Functional Introduction	Hydrostannylation		
Other functional group interconversion	Regitz diazo transfer	Functional Introduction	Diazotisation	Regitz diazo transfer	
Other heteroatom alkylation/arylation	Bromo stannylation	Functional Introduction	Stannylation		
Other functional group addition	Reimer-Tiemann formylation	Functional Introduction	Formylation		
Halogenation	Zincke sulfur chlorination	Functional Introduction	Chlorination	Sulphur	
Other functional group addition	Upjohn dihydroxylation	Functional Introduction	Hydroxylation	Alkene dihydroxylation	
Halogenation	Alkyne to alkene chlorination	Functional Introduction	Chlorination		
Other heteroatom alkylation/arylation	Chloro stannylation	Functional Introduction	Stannylation		
Other heteroatom alkylation/arylation	Stannylation	Functional Introduction	Stannylation		
N-acylation to amide	Chloro aminocarbonylation	Functional Introduction	Aminocarbonylation	Chloro	
Other heteroatom alkylation/arylation	Iodo stannylation	Functional Introduction	Stannylation		
N-acylation to amide	Iodo aminocarbonylation	Functional Introduction	Aminocarbonylation	Iodo	
N-acylation to amide	Bromo aminocarbonylation	Functional Introduction	Aminocarbonylation	Bromo	

Halogenation	Hell-Volhard-Zelinsky halogenation	Functional Introduction	Bromination	Hell-Volhard-Zelinsky	
Sulfonation	Alkene sulfoxy addition	Functional Introduction	Sulfonylation		
N-sulfonylation	Sulfonamide Schotten-Baumann	Other Bond Formation	Sulfonamide formation	Schotten-Baumann	
S-substitution	Sulfinic acid + chloride reaction	Other Bond Formation	Sulphur dioxide compound formation	Sulfinic acid + chloride	
N-sulfonylation	Sulfonic acid + amine reaction	Other Bond Formation	N-sulfonylation	Sulfonic acid + amine	
Other heteroatom alkylation/arylation	Michaelis-Arbuzov reaction	Other Bond Formation	Phosphonate formation	Michaelis-Arbuzov	
N-sulfonylation	Sulfinamide Schotten-Baumann	Other Bond Formation	Sulfinamide formation	Schotten-Baumann	
Other acylation	Phosphonamide Schotten-Baumann	Other Bond Formation	Phosphonamide formation	Schotten-Baumann	
S-substitution	Sulfinic acid + iodide reaction	Other Bond Formation	Sulphur dioxide compound formation	Sulfinic acid + iodide	
O-sulfonylation	Sulfinic ester Schotten-Baumann	Other Bond Formation	Sulfinic ester formation	Schotten-Baumann	
S-substitution	Sulfinic acid + bromide reaction	Other Bond Formation	Sulphur dioxide compound formation	Sulfinic acid + bromide	
S-substitution	Disulfide coupling	Other Bond Formation	Disulfide formation		
S-substitution	Sulfinic acid + fluoride reaction	Other Bond Formation	Sulphur dioxide compound formation	Sulfinic acid + fluoride	
ROH protections	O-Ac protection	Protection	O-Acetyl	O-Ac	
Other protections	Ketone dioxolane protection	Protection	Ketone dioxolane		
NH protections	N-TFA protection	Protection	N-Trifluoroacetyl	N-TFA	
Other protections	Aldehyde dioxolane protection	Protection	Aldehyde dioxolane		
NH protections	N-Boc protection	Protection	N-t-Butyloxycarbonyl	N-Boc	
ROH protections	O-Bn protection	Protection	O-Benzyl	O-Bn	
RCO ₂ H protections	CO ₂ H-tBu protection	Protection	COO-t-Buthyl	COO-tBu	
Other protections	Ketone dioxane protection	Protection	Ketone dioxane		
ROH protections	O-TMS protection	Protection	O-Trimethylsilyl ether	O-TMS	
ROH protections	O-TBS protection	Protection	O-t-Butyldimethylsilyl ether	O-TBS	

Other protections	Aldehyde dithiolane protection	Protection	Aldehyde dithiolane		
NH protections	N-Bn protection	Protection	N-Benzyl	N-Bn	
ROH protections	O-MOM protection	Protection	O-Methoxymethyl ether	O-MOM	
NH protections	N-Phth protection	Protection	N-Phthalimide	N-Phth	
Other protections	Aldehyde dithiane protection	Protection	Aldehyde dithiane		
Other protections	Ketone dithiane protection	Protection	Ketone dithiane		
Other protections	Ketone dithiolane protection	Protection	Ketone dithiolane		
NH protections	N-Fmoc protection	Protection	N-Fluorenylmethyloxycarbonyl	N-Fmoc	
NH protections	N-Cbz protection	Protection	N-Carbobenzyloxy	N-Cbz	
ROH protections	O-TIPS protection	Protection	O-Triisopropylsilyl ether	O-TIPS	
Other protections	Aldehyde dioxane protection	Protection	Aldehyde dioxane		
Other functional group interconversion	Pummerer rearrangement	Rearrangement	Sulphoxide to alpha-acyloxy-thioether	Pummerer	
O-containing heterocycle formation	Johnson-Corey-Chaykovsky epoxidation	Rearrangement	Ketone to epoxide	Johnson-Corey-Chaykovsky	
Carbamate/carbonate formation	Curtius reaction	Rearrangement	Azide to isocyanate	Curtius	
Other C-C bond formation	Favorskii rearrangement	Rearrangement	alpha-halo ketone to carboxylic acid	Favorskii	
Other C-C bond formation	Ortho Fries rearrangement	Rearrangement	Phenolic ester to hydroxyaryl ketone	Fries	Ortho
N-containing heterocycle formation	Cyclic Beckmann rearrangement	Rearrangement	Oxime to amide	Cyclic	Beckmann
Other C-C bond formation	Para Fries rearrangement	Rearrangement	Phenolic ester to hydroxyaryl ketone	Fries	Para
Other functional group interconversion	Wolff rearrangement	Rearrangement	alpha-diazoketone to ketene	Wolff	
Other acylation	Acyclic Beckmann rearrangement	Rearrangement	Oxime to amide	Acyclic	Beckmann
Other functional group interconversion	Curtius rearrangement	Rearrangement	Azide to isocyanate	Curtius	

N-containing heterocycle formation	Quinazolinone synthesis	Synthesis	Quinazolinone		
N-containing heterocycle formation	6-Pyridazinone synthesis	Synthesis	6-Pyridazinone		
S-substitution	Thioether synthesis	Synthesis	Thioether		
O-substitution	Mitsunobu aryl ether synthesis	Synthesis	Aryl ether		
N-containing heterocycle formation	Paal-Knorr pyrrole synthesis	Synthesis	Pyrrole	Paal-Knorr	
N-containing heterocycle formation	Tetrazole synthesis	Synthesis	Tetrazole		
N-containing heterocycle formation	Knorr pyrazole synthesis	Synthesis	Pyrazole	Knorr	
N-containing heterocycle formation	Piperidine synthesis	Synthesis	Piperidine		
N-containing heterocycle formation	Fischer indole synthesis	Synthesis	Fisher indole		
S-containing heterocycle formation	Thiazole synthesis	Synthesis	Thiazole		
N-containing heterocycle formation	Pyrimidone synthesis	Synthesis	Pyrimidone		
O-containing heterocycle formation	1,2,4-Oxadiazole synthesis	Synthesis	1-2-4-Oxadiazole		
N-containing heterocycle formation	Benzimidazole synthesis	Synthesis	Benzimidazole		
N-containing heterocycle formation	Pyrrolidine synthesis	Synthesis	Pyrrolidine		
O-substitution	Ether synthesis	Synthesis	Ether		
S-containing heterocycle formation	1,3,4-Thiadiazole synthesis	Synthesis	1-3-4-Thiadiazole		
N-containing heterocycle formation	Dihydropyridine synthesis	Synthesis	Dihydropyridine		
N-containing heterocycle formation	1,2,4-Triazole synthesis	Synthesis	1-2-4-Triazole		
N-containing heterocycle formation	Pyrazole synthesis	Synthesis	Pyrazole		

O-containing heterocycle formation	1,2-Benzoxazole synthesis	Synthesis	1-2-Benzoxazole		
N-containing heterocycle formation	Pyrazolone synthesis	Synthesis	Pyrazolone		
N-containing heterocycle formation	Piperazine synthesis	Synthesis	Piperazine		
O-containing heterocycle formation	2-Oxazoline synthesis	Synthesis	2-Oxazoline		
O-containing heterocycle formation	1,3-Benzoxazole synthesis	Synthesis	1-3-Benzoxazole		
N-containing heterocycle formation	2,5-Pyrroledione synthesis	Synthesis	2-5-Pyrroledione		
N-containing heterocycle formation	Indazole synthesis	Synthesis	Indazole		
N-containing heterocycle formation	Bischler-Napieralski reaction	Synthesis	3-4-Dihydroisoquinolines	Bischler-Napieralski	
Other functional group interconversion	Hofmann isonitrile synthesis	Synthesis	Isonitrile	Hofmann	
O-containing heterocycle formation	3,1-Benzoxazin-4-one synthesis	Synthesis	3-1-Benzoxazin-4-one		
N-containing heterocycle formation	Imidazole synthesis	Synthesis	Imidazole		
N-containing heterocycle formation	Benzotriazole synthesis	Synthesis	Benzotriazole		
S-containing heterocycle formation	Benzothiazole synthesis	Synthesis	Benzothiazole		
N-containing heterocycle formation	Gassman indolone synthesis	Synthesis	Indolone	Gassman	
N-containing heterocycle formation	Pyrazolamine synthesis	Synthesis	Pyrazolamine		
S-containing heterocycle formation	1,2,4-Thiadiazole synthesis	Synthesis	1-2-4-Thiadiazole		
N-containing heterocycle formation	Pyazine synthesis	Synthesis	Pyazine		
N-containing heterocycle formation	Lactam synthesis	Synthesis	Lactam		

N-containing heterocycle formation	Benzimidazolone synthesis	Synthesis	Benzimidazolone		
N-containing heterocycle formation	[1,2,4]Triazolo[4,3-a]pyridine synthesis	Synthesis	[1-2-4]Triazolo[4-3-a]pyridine		
O-containing heterocycle formation	Oxazole synthesis	Synthesis	Oxazole		
O-containing heterocycle formation	Paal-Knorr furan synthesis	Synthesis	Furan	Paal-Knorr	
N-containing heterocycle formation	Pyridone synthesis	Synthesis	Pyridone		
N-containing heterocycle formation	Niementowski quinazoline synthesis	Synthesis	Quinazoline	Niementowski	
N-containing heterocycle formation	2-Pyrrolidone synthesis	Synthesis	2-Pyrrolidone		
S-containing heterocycle formation	Isothiazole synthesis	Synthesis	Isothiazole		
S-containing heterocycle formation	Thiazoline synthesis	Synthesis	Thiazoline		
N-containing heterocycle formation	Benzimidazolethione synthesis	Synthesis	Benzimidazolethione		
Other C-C bond formation	Johnson-Corey-Chaykovsky cyclopropane synthesis	Synthesis	Cyclopropane	Johnson-Corey-Chaykovsky	
O-containing heterocycle formation	1,3,4-Oxadiazole synthesis	Synthesis	1-3-4-Oxadiazole		
O-containing heterocycle formation	Dioxolane synthesis	Synthesis	Dioxolane		
N-containing heterocycle formation	2,4-Quinazolidinedione synthesis	Synthesis	2-4-Quinazolidinedione		
O-containing heterocycle formation	Morpholine synthesis	Synthesis	Morpholine		
N-containing heterocycle formation	Skraup reaction	Synthesis	Quinoline	Skraup-Doebner-Von Miller	
N-containing heterocycle formation	Pyrimidine synthesis	Synthesis	Pyrimidine		
N-containing heterocycle formation	Quinazoline synthesis	Synthesis	Quinazoline		

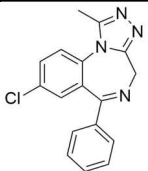
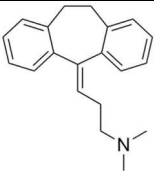
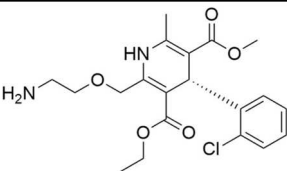
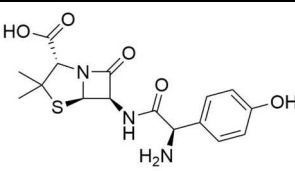
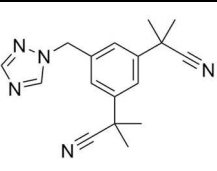
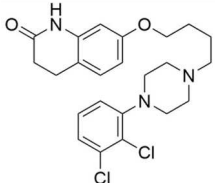
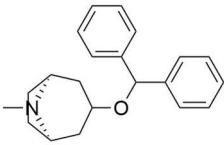
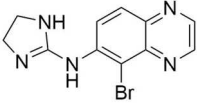
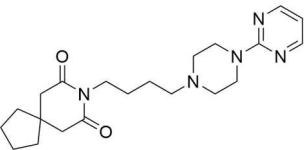
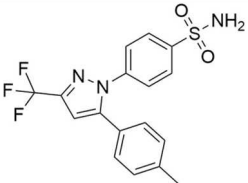
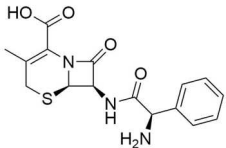
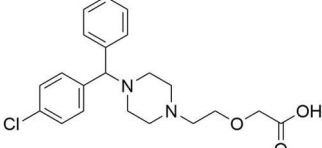
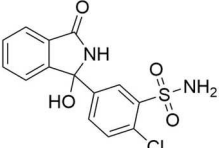
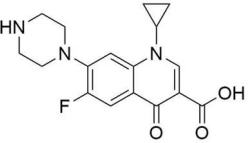
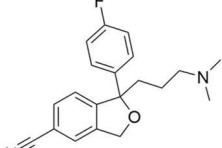
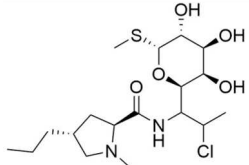
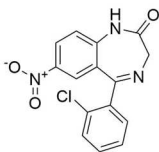
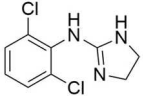
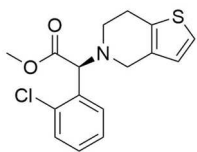
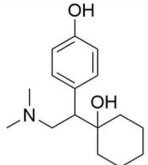
N-containing heterocycle formation	1,2,3-Triazole synthesis	Synthesis	1-2-3-Triazole		
N-containing heterocycle formation	Debus-Radziszewski imidazole synthesis	Synthesis	Imidazole	Debus-Radziszewski	
N-containing heterocycle formation	2-Thioxopyrimidin-2-one synthesis	Synthesis	2-Thioxopyrimidin-2-one		
N-containing heterocycle formation	Doebner-Miller reaction	Synthesis	Quinoline	Skraup-Doebner-Von Miller	
N-containing heterocycle formation	Conrad-Limpach quinoline synthesis	Synthesis	Quinoline	Conrad-Limpach	
S-containing heterocycle formation	Paal-Knorr thiophene synthesis	Synthesis	Thiophene	Paal-Knorr	
N-containing heterocycle formation	Pyrazolo[1,5-a]pyrimidine synthesis	Synthesis	Pyrazolo[1-5-a]pyrimidine		
N-containing heterocycle formation	[1,2,4]Triazolo[1,5-a]pyridine synthesis	Synthesis	[1-2-4]Triazolo[1-5-a]pyridine		
Other C-C bond formation	Kishner diazomethane cyclopropane synthesis	Synthesis	Cyclopropane	Kishner	
O-containing heterocycle formation	Gewald furan synthesis	Synthesis	Furan	Gewald	
N-containing heterocycle formation	Pfitzinger reaction	Synthesis	Quinoline-4-carboxylic acids	Pfitzinger	
N-containing heterocycle formation	Hydroquinazolinone synthesis	Synthesis	Hydroquinazolinone		
N-containing heterocycle formation	Knorr quinoline synthesis	Synthesis	Quinoline	Knorr	
O-containing heterocycle formation	Chromanone synthesis	Synthesis	Chromanone		
S-containing heterocycle formation	Benzothiophene synthesis	Synthesis	Benzothiophene		
N-containing heterocycle formation	Pinner pyrimidine synthesis	Synthesis	Pyrimidine	Pinner	
O-containing heterocycle formation	Isoxazole synthesis	Synthesis	Isoxazole		
N-containing heterocycle formation	Biginelli reaction	Synthesis	3-4-Dihydropyrimidin-2[1H]-ones	Biginelli	

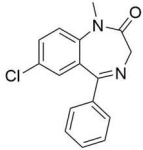
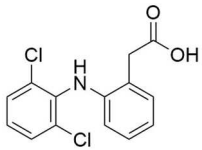
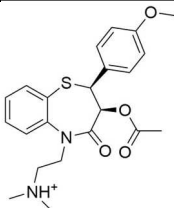
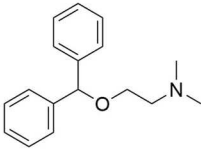
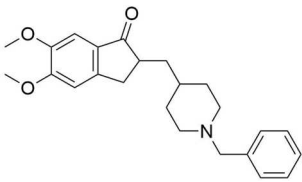
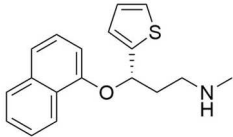
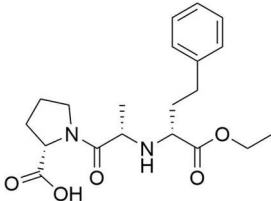
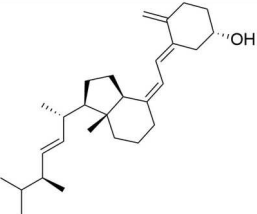
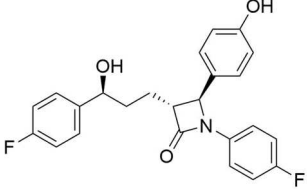
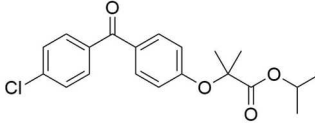
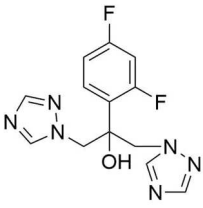
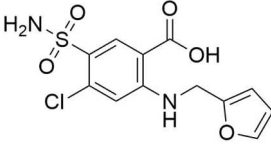
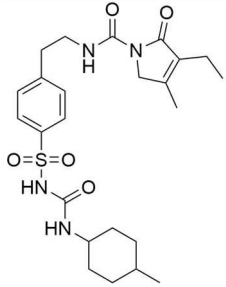
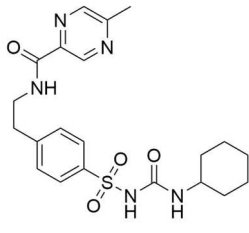
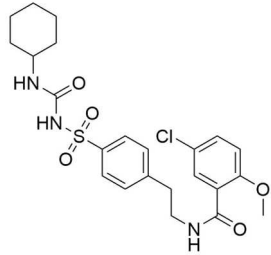
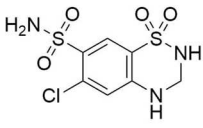
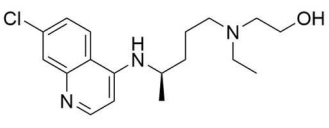
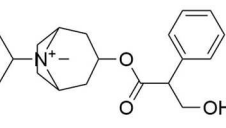
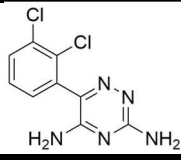
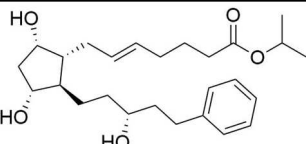
N-containing heterocycle formation	[1,2,4]Triazolo[1,5-a]pyrimidin-7-one synthesis	Synthesis	[1-2-4]Triazolo[1-5-a]pyrimidin-7-one		
N-containing heterocycle formation	Pyridine synthesis	Synthesis	Pyridine		
Other C-C bond formation	Ketene S,S-acetal synthesis	Synthesis	Ketene S-S-acetal		
S-containing heterocycle formation	Thiophene synthesis	Synthesis	Thiophene		
O-containing heterocycle formation	Van Leusen oxazole synthesis	Synthesis	Oxazole	Van Leusen	
N-containing heterocycle formation	Van Leusen imidazole synthesis	Synthesis	Imidazole	Van Leusen	
N-containing heterocycle formation	Pyridotriazole synthesis	Synthesis	Pyridotriazole		
N-containing heterocycle formation	Isoindolinone synthesis	Synthesis	Isoindolinone		
N-containing heterocycle formation	Indole synthesis	Synthesis	Indole		
N-containing heterocycle formation	Doebner reaction	Synthesis	Quinoline	Skraup-Doebner-Von Miller	
O-containing heterocycle formation	Benzofuran synthesis	Synthesis	Benzofuran		
O-containing heterocycle formation	1,3-Benzoxazol-2-one synthesis	Synthesis	1-3-Benzoxazol-2-one		
N-containing heterocycle formation	Knorr pyrrole synthesis	Synthesis	Pyrrole	Knorr	
O-containing heterocycle formation	1,3,4-Oxadiazol-2-one synthesis	Synthesis	1-3-4-Oxadiazol-2-one		
N-containing heterocycle formation	Pyrazolo[1,5-a]pyridine synthesis	Synthesis	Pyrazolo[1-5-a]pyridine		
O-containing heterocycle formation	Dihydroisoxazole synthesis	Synthesis	Dihydroisoxazole		
N-containing heterocycle formation	Friedlander quinoline synthesis	Synthesis	Quinoline	Friedlander	
N-containing heterocycle formation	Larock indole synthesis	Synthesis	Indole	Larock	

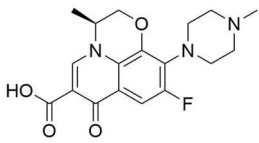
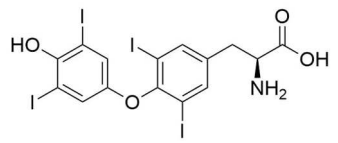
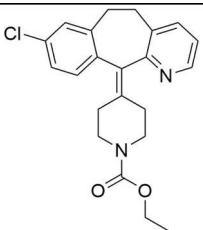
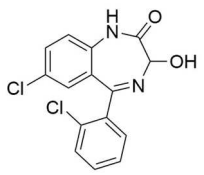
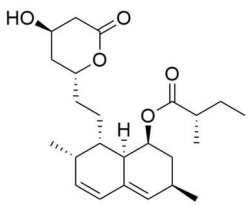
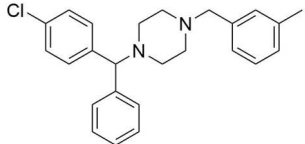
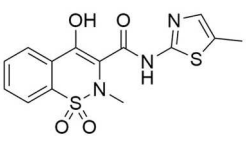
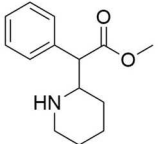
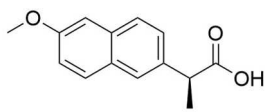
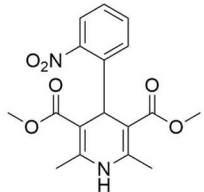
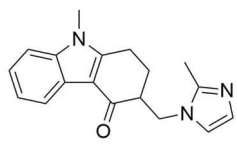
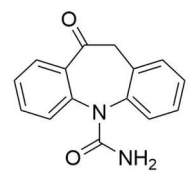
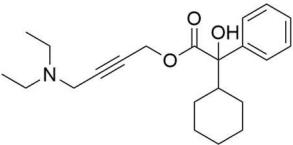
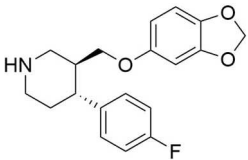
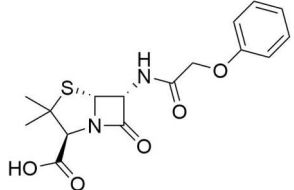
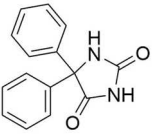
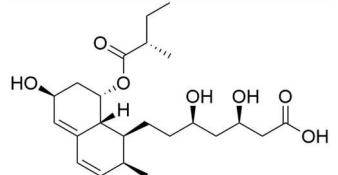
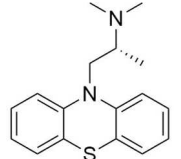
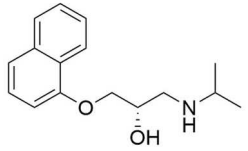
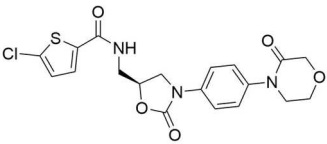
N-containing heterocycle formation	Gewald pyrrole synthesis	Synthesis	Pyrrole	Gewald	
N-containing heterocycle formation	Phthalazinone synthesis	Synthesis	Phthalazinone		
O-containing heterocycle formation	3,1-Benzoxazine-2,4-dione synthesis	Synthesis	3-1-Benzoxazine-2-4-dione		
N-containing heterocycle formation	Pechmann pyrazole synthesis	Synthesis	Pyrazole	Pechmann	
N-containing heterocycle formation	Phthalazine synthesis	Synthesis	Phthalazine		
N-containing heterocycle formation	[1,2,4]Triazolo[4,3-a]pyridin-3-one synthesis	Synthesis	[1-2-4]Triazolo[4-3-a]pyridin-3-one		
N-containing heterocycle formation	1,2,4-Triazole-3-thione synthesis	Synthesis	1-2-4-Triazole-3-thione		
N-containing heterocycle formation	Imidazo[1,2-a]pyridine synthesis	Synthesis	Imidazo[1-2-a]pyridine		
S-containing heterocycle formation	1,3-Benzothiazin-4-one synthesis	Synthesis	1-3-Benzothiazin-4-one		
O-containing heterocycle formation	1,2,4-Oxadiazol-5-one synthesis	Synthesis	1-2-4-Oxadiazol-5-one		
N-containing heterocycle formation	Niementowski quinoline synthesis	Synthesis	Quinoline	Niementowski	

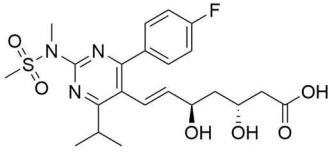
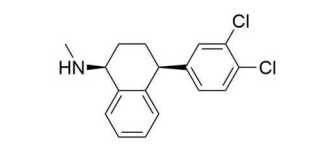
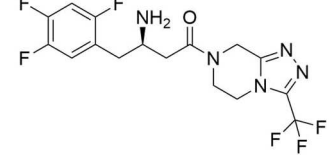
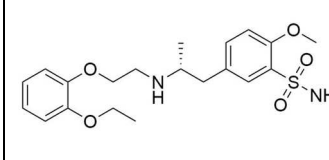
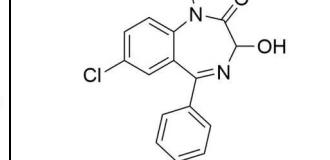
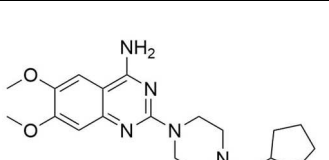
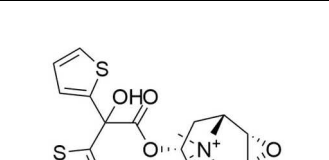
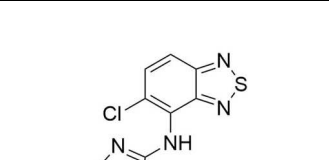
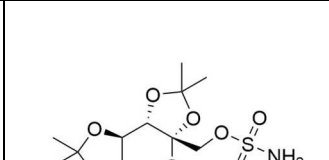
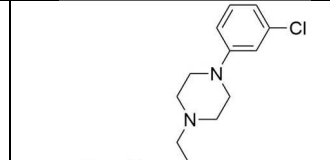
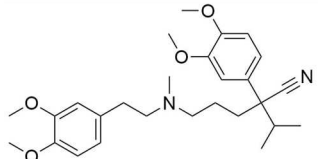
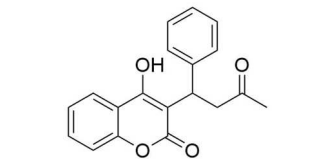
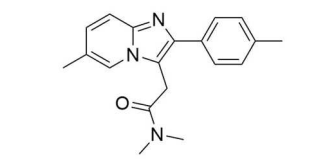
Appendix B

Top 200 Drugs 2017 Retained Structures

				
Alprazolam (*)	Amitriptyline	Amlodipine	Amoxicillin	Anastrozole
				
Aripiprazole	Bzotropine	Brimonidine	Buspirone	Celecoxib
				
Cephalexin	Cetirizine	Chlorthalidone (*)	Ciprofloxacin	Citalopram (*)
				
Clindamycin	Clonazepam (*)	Clonidine	Clopidogrel	Desvenlafaxine

				
Diazepam (*)	Diclofenac	Diltiazem	Diphenhydramine (*)	Donepezil
				
Duloxetine	Enalapril	Ergocalciferol	Ezetimibe	Fenofibrate
				
Fluconazole	Furosemide	Glimepiride	Glipizide	Glyburide
				
Hydrochlorothiazide (*)	Hydroxychloroquine	Ipratropium	Lamotrigine	Latanoprost

				
Levofloxacin	Levothyroxine	Loratadine	Lorazepam (*)	Lovastatin
				
Meclizine	Meloxicam	Methylphenidate	Naproxen	Nifedipine
				
Ondansetron	Oxcarbazepine (*)	Oxybutynin	Paroxetine	Penicillin V
				
Phenytoin	Pravastatin	Promethazine	Propranolol	Rivaroxaban

				
Rosuvastatin	Sertraline (*)	Sitagliptin	Tamsulosin	Temazepam (*)
				
Terazosin	Tiotropium	Tizanidine	Topiramate	Trazodone
				
Verapamil	Warfarin	Zolpidem		

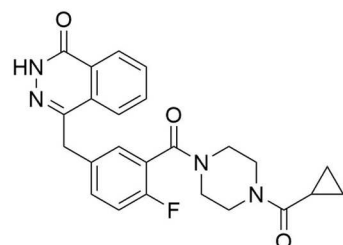
Note: The structures that have failed the BRICS decomposition are marked by a star in brackets.

Pgp-BCRP Substrate and BBB Penetration Classification Model Metrics

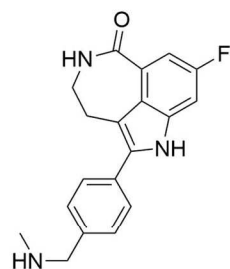
Descriptor	Pgp Substrate				BCRP Substrate				BBB Character			
	Recall	Precision	F1-score	MCC	Recall	Precision	F1-score	MCC	Recall	Precision	F1-score	MCC
Avalon (Binary)	0.70	0.70	0.70	0.40	0.73	0.74	0.72	0.43	0.93	0.94	0.93	0.80
FeatMorgan (Binary)	0.72	0.72	0.72	0.44	0.66	0.65	0.63	0.25	0.92	0.92	0.92	0.77
FeatMorgan (Count)	0.74	0.75	0.74	0.48	0.64	0.64	0.61	0.22	0.93	0.93	0.93	0.78
Morgan (Binary)	0.67	0.68	0.66	0.34	0.68	0.68	0.66	0.31	0.92	0.93	0.92	0.77
Morgan (Count)	0.69	0.71	0.69	0.40	0.69	0.71	0.67	0.35	0.93	0.93	0.92	0.77
MOE Descriptors	0.77	0.78	0.77	0.55	0.82	0.82	0.82	0.62	0.92	0.92	0.92	0.78

PARP1 Query BRICS Fragmentation

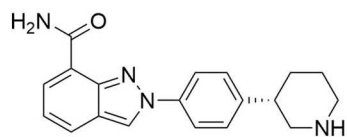
Key Fragments



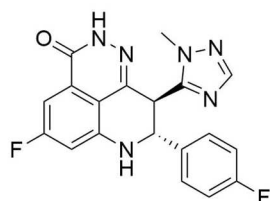
Olaparib



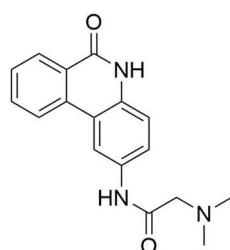
Rucaparib



Niraparib

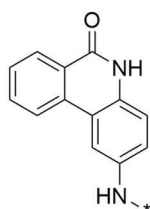
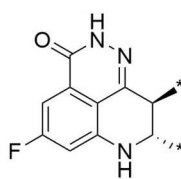
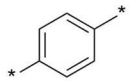
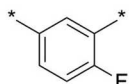


Talazoparib

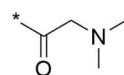
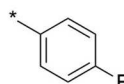
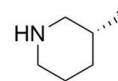
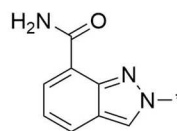
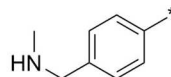
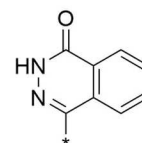
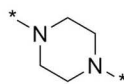


PJ34

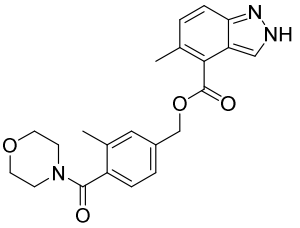
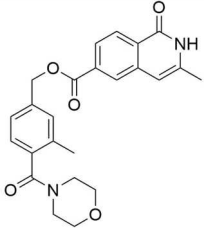
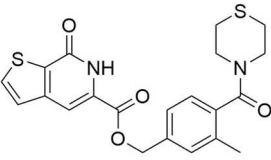
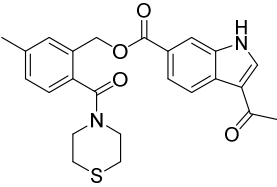
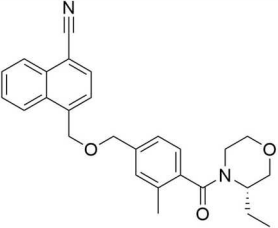
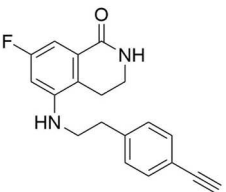
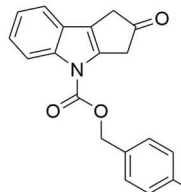
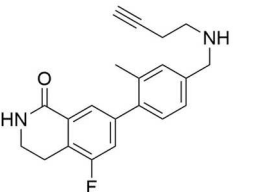
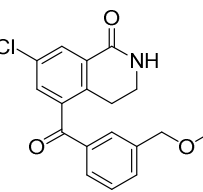
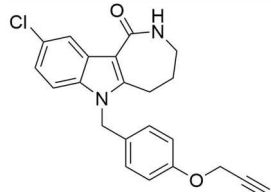
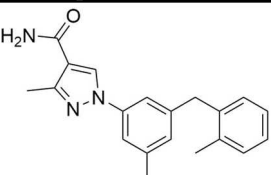
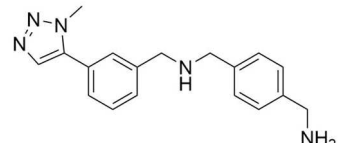
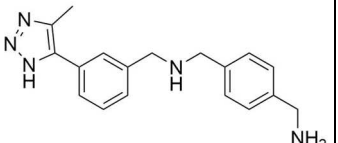
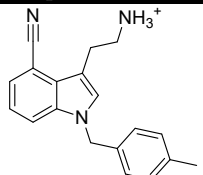
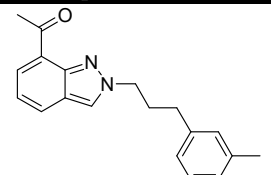
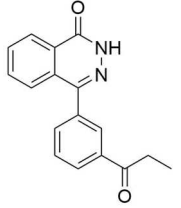
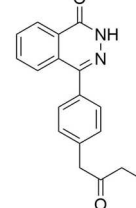
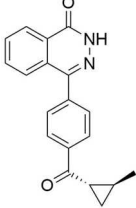
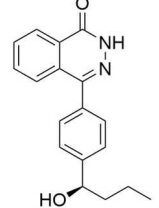
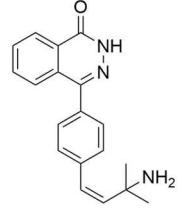
Starting Materials



Reagents



PARP1 Docking Selection - Structures

				
Olaparib - Row26	Olaparib - Row514	Olaparib - Row217	Olaparib - Row563	Olaparib - Row538
				
Rucaparib - Row312	Rucaparib - Row443	Rucaparib - Row528	Rucaparib - Row665	Rucaparib - Row600
				
Niraparib - Row113	Niraparib - Row760	Niraparib - Row760c	Niraparib - Row847	Niraparib - Row408
				
PJ34 - Row27	PJ34 - Row5	PJ34 - Row745(2)	PJ34 - Row86	PJ34 - Row304

PARP1 Docking Selection - Scores

Query	Candidate	PLP.Fitness (Mean \pm SD)	Goldscore.Fitness (Mean \pm SD)	Pose Consistency	PARP1 QSAR pIC50 (μ M)
Olaparib	Row26	88.96 \pm 1.96	61.06 \pm 2.09	10/10	0.92
Olaparib	Row514	90.33 \pm 3.28	53.76 \pm 2.78	6/10	0.79
Olaparib	Row217	84.81 \pm 6.55	59.65 \pm 6.17	4/10	0.84
Olaparib	Row563	90.09 \pm 2.68	56.73 \pm 1.89	4/10	0.78
Olaparib	Row538	93.04 \pm 3.92	45.40 \pm 12.33	6/10	0.78
Rucaparib	Row312	85.60 \pm 1.52	51.30 \pm 0.20	3/10	0.70
Rucaparib	Row443	81.01 \pm 1.08	53.80 \pm 2.77	8/10	0.64
Rucaparib	Row528	87.42 \pm 1.83	49.79 \pm 2.72	10/10	0.66
Rucaparib	Row665	74.62 \pm 0.40	33.78 \pm 6.74	3/10	0.59
Rucaparib	Row600	84.62 \pm 4.04	43.96 \pm 0.93	2/10	0.66
Niraparib	Row113	86.70 \pm 0.34	58.50 \pm 2.43	8/10	0.86
Niraparib	Row760	87.27 \pm 1.50	40.96 \pm 16.88	7/10	0.71
Niraparib	Row760c	78.50 \pm 1.65	49.94 \pm 5.91	6/10	0.49
Niraparib	Row847	88.95 \pm 0.93	60.48 \pm 4.45	6/10	0.58
Niraparib	Row408	87.39 \pm 2.45	55.45 \pm 6.59	7/10	0.78
PJ34	Row27	88.65 \pm 0.21	55.33 \pm 1.64	10/10	0.74
PJ34	Row5	85.35 \pm 1.56	57.54 \pm 1.81	9/10	0.84
PJ34	Row745(2)	80.76 \pm 1.11	61.37 \pm 1.04	7/10	0.47
PJ34	Row86	86.80 \pm 0.29	53.61 \pm 2.77	8/10	0.66
PJ34	Row304	95.14 \pm 2.18	70.46 \pm 5.42	6/10	0.56

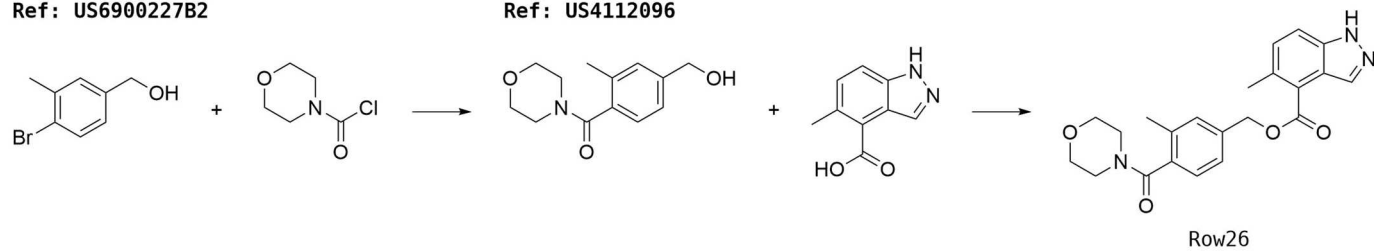
Appendix C

Olaparib - Row26

Proposed Route:

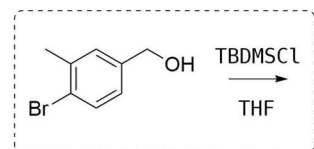
Step 1
Ref: US6900227B2

Step 2
Ref: US4112096

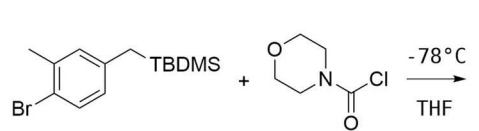


Actual Route:

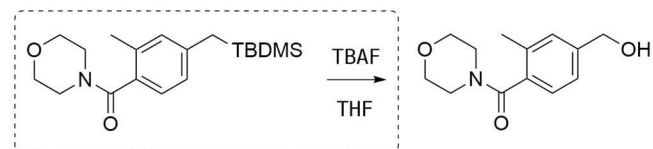
Protection



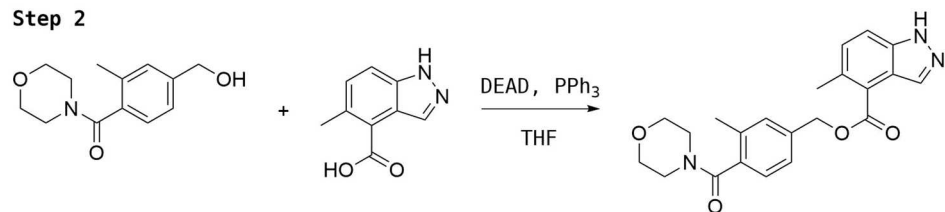
Step 1



Deprotection



Step 2

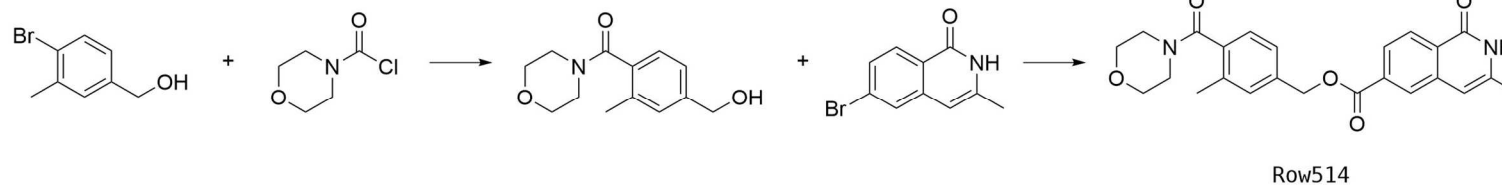


Olaparib - Row514

Proposed Route:

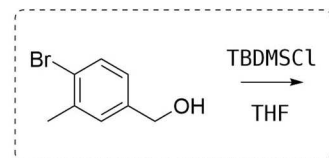
Step 1
Ref: US6900227B2

Step 2
Ref: US8476253B2

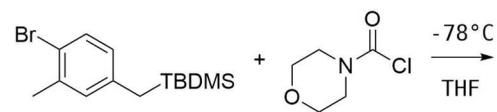


Actual Route:

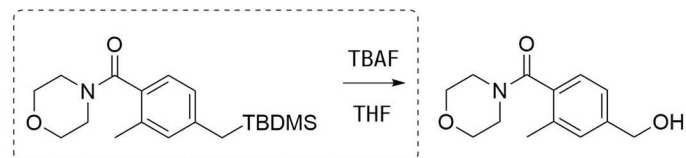
Protection



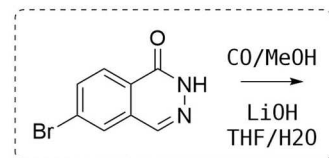
Step 1



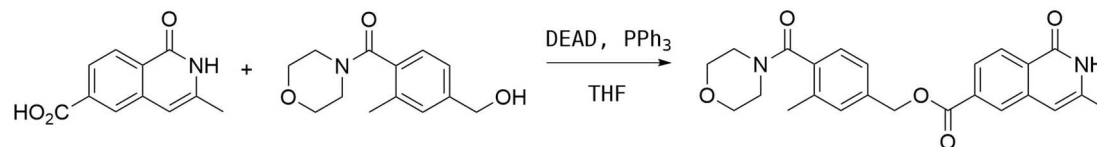
Deprotection



Group Conversion



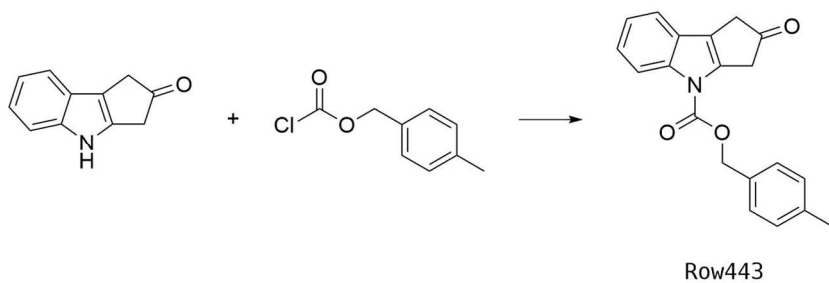
Step 2



Rucaparib - Row443

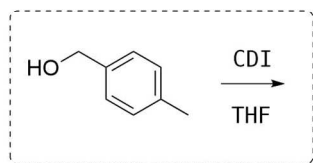
Proposed Route:

Step 1
Ref: US6469020B2

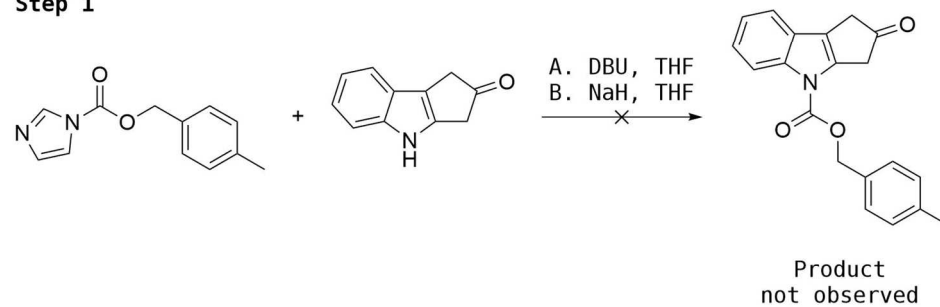


Actual Route:

Protection



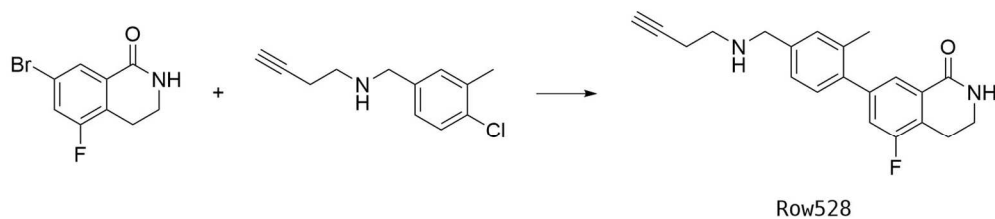
Step 1



Rucaparib - Row528

Proposed Route:

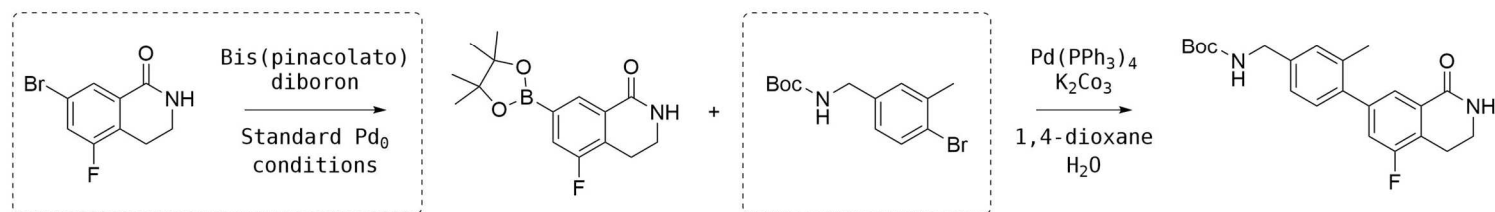
Step 1
Ref: US5380910



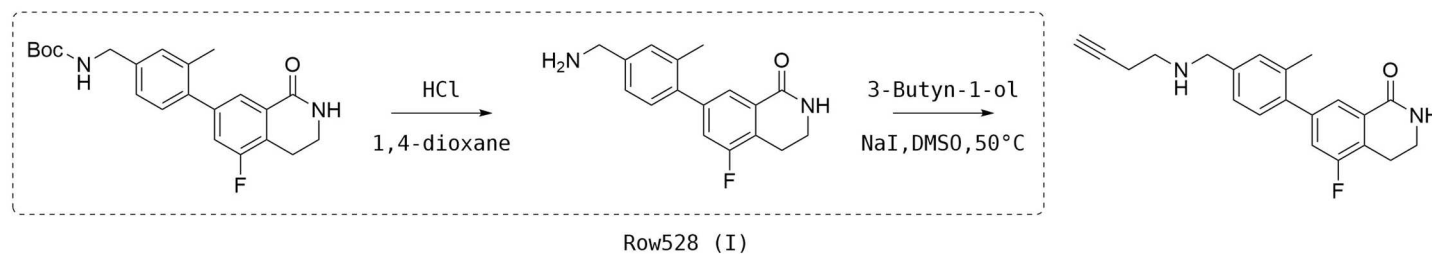
Actual Route:

Building Block Conversion

Suzuki Coupling (Alternative to Ullmann-type Coupling)



Functionalisation

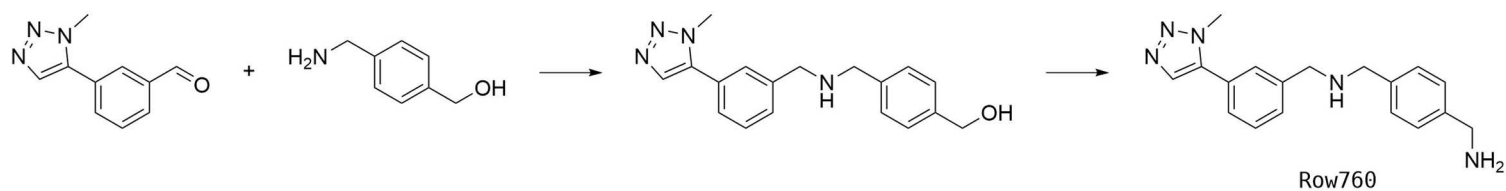


Niraparib - Row760c

Proposed Route:

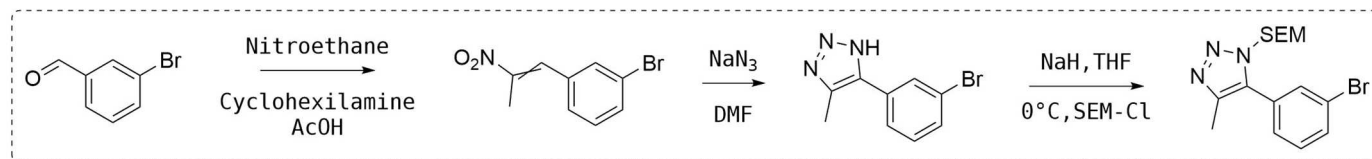
Step 1
Ref: US4442286

Step 2
Ref: US5002949

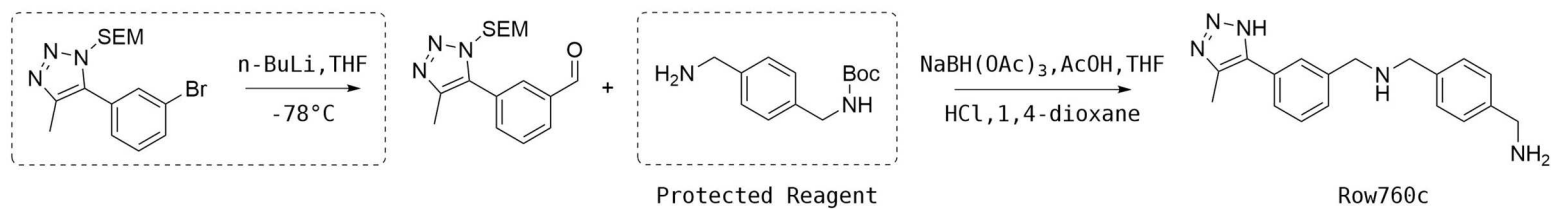


Actual Route:

Alternative Building Block Preparation



Step 1-2

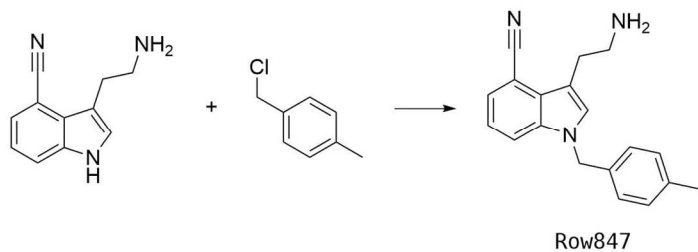


Niraparib - Row847

Proposed Route:

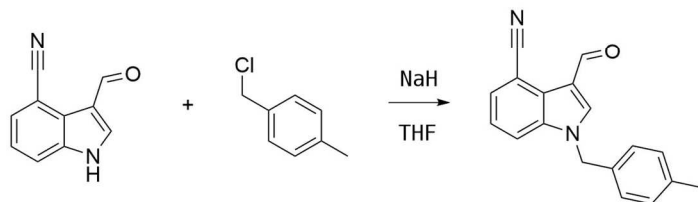
Step 1

Ref: US4104467

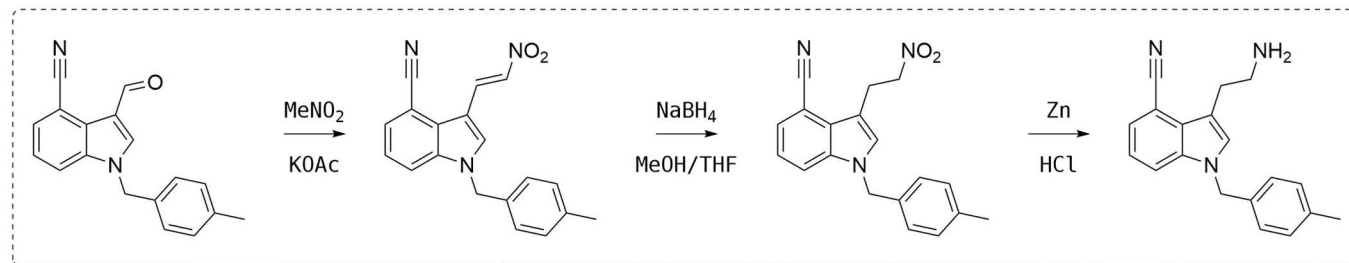


Actual Route:

Step 1



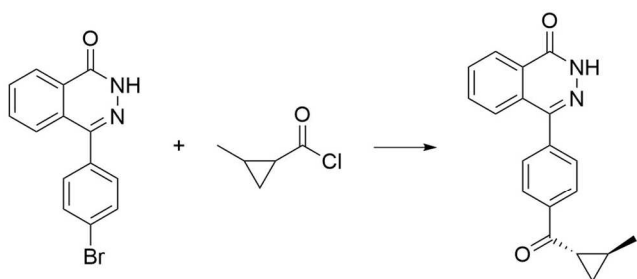
Functionalisation



PJ34 - Row745(2)

Proposed Route:

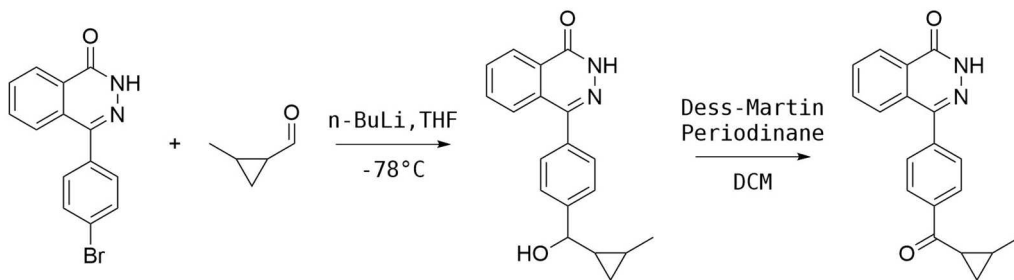
Step 1
Ref: US4028404



Row745(2)

Actual Route:

Organolithium Alternative



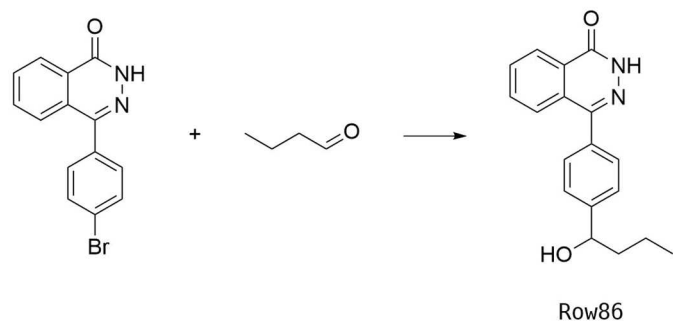
Row745(2) (I)

PJ34 - Row86

Proposed Route:

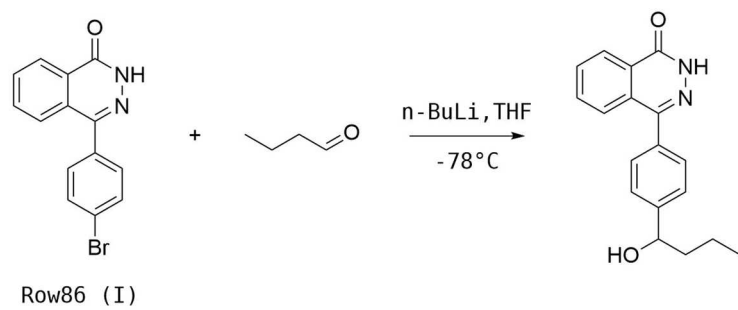
Step 1

Ref: US5156763



Actual Route:

Organolithium Alternative



Appendix D

PoC Model

Level-4 and Level-2 Label Datasets Comparison (Only PT Approaches with RF)

Molecular Representation	Approach	Level	Hamming Loss	0-1 Loss	Micro Recall	Micro Precision	Micro F1-score
Atom-pair	BR	2	0.03	0.93	0.07	0.70	0.12
Atom-pair	BR	4	0.02	0.95	0.06	0.68	0.10
Atom-pair	CC	2	0.03	0.93	0.07	0.70	0.13
Atom-pair	CC	4	0.02	0.94	0.06	0.69	0.10
Atom-pair	LP	2	0.04	0.75	0.26	0.27	0.26
Atom-pair	LP	4	0.03	0.78	0.23	0.24	0.24
Avalon (1024-bit)	BR	2	0.03	0.84	0.16	0.69	0.26
Avalon (1024-bit)	BR	4	0.02	0.86	0.15	0.66	0.25
Avalon (1024-bit)	CC	2	0.03	0.83	0.17	0.69	0.28
Avalon (1024-bit)	CC	4	0.02	0.85	0.16	0.66	0.25
Avalon (1024-bit)	LP	2	0.04	0.65	0.36	0.37	0.37
Avalon (1024-bit)	LP	4	0.03	0.66	0.35	0.36	0.36
Avalon (2048-bit)	BR	2	0.03	0.82	0.19	0.70	0.30
Avalon (2048-bit)	BR	4	0.02	0.84	0.16	0.68	0.26
Avalon (2048-bit)	CC	2	0.03	0.81	0.19	0.69	0.30
Avalon (2048-bit)	CC	4	0.02	0.84	0.16	0.68	0.27
Avalon (2048-bit)	LP	2	0.03	0.62	0.39	0.41	0.40
Avalon (2048-bit)	LP	4	0.03	0.66	0.36	0.37	0.36
Avalon (256-bit)	BR	2	0.03	0.89	0.12	0.69	0.20
Avalon (256-bit)	BR	4	0.02	0.91	0.09	0.63	0.16
Avalon (256-bit)	CC	2	0.03	0.88	0.12	0.66	0.20
Avalon (256-bit)	CC	4	0.02	0.91	0.10	0.61	0.17
Avalon (256-bit)	LP	2	0.04	0.70	0.31	0.32	0.32
Avalon (256-bit)	LP	4	0.03	0.74	0.27	0.28	0.27

Avalon (4096-bit)	BR	2	0.03	0.81	0.19	0.69	0.30
Avalon (4096-bit)	BR	4	0.02	0.84	0.17	0.66	0.27
Avalon (4096-bit)	CC	2	0.03	0.81	0.20	0.69	0.31
Avalon (4096-bit)	CC	4	0.02	0.84	0.17	0.65	0.26
Avalon (4096-bit)	LP	2	0.03	0.62	0.40	0.40	0.40
Avalon (4096-bit)	LP	4	0.03	0.65	0.36	0.37	0.37
Avalon (640-bit)	BR	2	0.03	0.85	0.15	0.66	0.25
Avalon (640-bit)	BR	4	0.02	0.87	0.13	0.65	0.22
Avalon (640-bit)	CC	2	0.03	0.84	0.16	0.68	0.26
Avalon (640-bit)	CC	4	0.02	0.86	0.14	0.65	0.23
Avalon (640-bit)	LP	2	0.04	0.66	0.35	0.36	0.36
Avalon (640-bit)	LP	4	0.03	0.69	0.32	0.33	0.33
Avalon (8192-bit)	BR	2	0.03	0.81	0.19	0.70	0.30
Avalon (8192-bit)	BR	4	0.02	0.84	0.17	0.67	0.27
Avalon (8192-bit)	CC	2	0.03	0.81	0.20	0.69	0.30
Avalon (8192-bit)	CC	4	0.02	0.84	0.16	0.67	0.26
Avalon (8192-bit)	LP	2	0.03	0.61	0.40	0.41	0.40
Avalon (8192-bit)	LP	4	0.03	0.64	0.37	0.38	0.38
CDK Functional Group	BR	2	0.03	0.84	0.18	0.54	0.27
CDK Functional Group	BR	4	0.02	0.86	0.16	0.52	0.25
CDK Functional Group	CC	2	0.03	0.80	0.21	0.52	0.30
CDK Functional Group	CC	4	0.02	0.82	0.19	0.51	0.28
CDK Functional Group	LP	2	0.04	0.68	0.37	0.36	0.37
CDK Functional Group	LP	4	0.03	0.70	0.34	0.34	0.34
ChemAxon Functional Group	BR	2	0.05	0.85	0.21	0.41	0.28
ChemAxon Functional Group	BR	4	0.04	0.88	0.18	0.36	0.24
ChemAxon Functional Group	CC	2	0.05	0.83	0.19	0.43	0.27
ChemAxon Functional Group	CC	4	0.04	0.85	0.16	0.38	0.23
ChemAxon Functional Group	LP	2	0.07	0.79	0.32	0.28	0.30
ChemAxon Functional Group	LP	4	0.05	0.82	0.30	0.26	0.28
Chi Kappa Descriptors	BR	2	0.02	0.97	0.03	0.51	0.06
Chi Kappa Descriptors	BR	4	0.02	0.97	0.03	0.51	0.07
Chi Kappa Descriptors	CC	2	0.02	0.97	0.03	0.47	0.06
Chi Kappa Descriptors	CC	4	0.02	0.96	0.04	0.48	0.07
Chi Kappa Descriptors	LP	2	0.04	0.86	0.14	0.15	0.15

Chi Kappa Descriptors	LP	4	0.04	0.86	0.15	0.16	0.15
Dompé	BR	2	0.03	0.86	0.16	0.42	0.23
Dompé	BR	4	0.03	0.87	0.15	0.43	0.22
Dompé	CC	2	0.02	0.84	0.16	0.46	0.24
Dompé	CC	4	0.02	0.85	0.16	0.47	0.24
Dompé	LP	2	0.04	0.75	0.31	0.26	0.28
Dompé	LP	4	0.04	0.74	0.31	0.27	0.29
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	2	0.03	0.86	0.15	0.57	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	4	0.02	0.89	0.13	0.55	0.20
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	2	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	4	0.02	0.86	0.14	0.55	0.23
FeatMorgan (1024-bit) (Radius 1) (Binary)	LP	2	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	LP	4	0.03	0.72	0.31	0.31	0.31
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	2	0.03	0.85	0.15	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	4	0.02	0.88	0.12	0.66	0.21
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	2	0.03	0.85	0.16	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	4	0.02	0.87	0.13	0.65	0.21
FeatMorgan (1024-bit) (Radius 2) (Binary)	LP	2	0.04	0.64	0.37	0.38	0.38
FeatMorgan (1024-bit) (Radius 2) (Binary)	LP	4	0.03	0.67	0.34	0.35	0.35
FeatMorgan (1024-bit) (Radius 2) (Count)	BR	2	0.02	0.88	0.12	0.68	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	BR	4	0.02	0.89	0.11	0.73	0.19
FeatMorgan (1024-bit) (Radius 2) (Count)	CC	2	0.02	0.88	0.12	0.67	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	CC	4	0.02	0.89	0.11	0.74	0.19
FeatMorgan (1024-bit) (Radius 2) (Count)	LP	2	0.03	0.67	0.34	0.35	0.35
FeatMorgan (1024-bit) (Radius 2) (Count)	LP	4	0.03	0.68	0.32	0.34	0.33
FeatMorgan (1024-bit) (Radius 3) (Binary)	BR	2	0.03	0.89	0.10	0.71	0.18
FeatMorgan (1024-bit) (Radius 3) (Binary)	BR	4	0.02	0.92	0.08	0.64	0.15
FeatMorgan (1024-bit) (Radius 3) (Binary)	CC	2	0.03	0.89	0.11	0.73	0.20
FeatMorgan (1024-bit) (Radius 3) (Binary)	CC	4	0.02	0.91	0.09	0.64	0.16
FeatMorgan (1024-bit) (Radius 3) (Binary)	LP	2	0.04	0.67	0.34	0.36	0.35
FeatMorgan (1024-bit) (Radius 3) (Binary)	LP	4	0.03	0.70	0.31	0.33	0.32
Layered	BR	2	0.03	0.89	0.12	0.65	0.20
Layered	BR	4	0.02	0.90	0.10	0.63	0.18
Layered	CC	2	0.03	0.88	0.13	0.66	0.21
Layered	CC	4	0.02	0.90	0.11	0.64	0.18

Layered	LP	2	0.04	0.71	0.30	0.31	0.31
Layered	LP	4	0.03	0.75	0.26	0.27	0.27
MACCS	BR	2	0.03	0.84	0.17	0.65	0.27
MACCS	BR	4	0.02	0.87	0.14	0.61	0.23
MACCS	CC	2	0.03	0.82	0.19	0.63	0.29
MACCS	CC	4	0.02	0.85	0.15	0.59	0.24
MACCS	LP	2	0.04	0.65	0.36	0.37	0.37
MACCS	LP	4	0.03	0.69	0.33	0.34	0.34
Morgan	BR	2	0.03	0.89	0.11	0.76	0.20
Morgan	BR	4	0.02	0.91	0.09	0.73	0.16
Morgan	CC	2	0.03	0.88	0.12	0.76	0.21
Morgan	CC	4	0.02	0.90	0.10	0.71	0.17
Morgan	LP	2	0.04	0.65	0.36	0.37	0.36
Morgan	LP	4	0.03	0.68	0.33	0.34	0.33
OChem EFG+	BR	2	0.03	0.80	0.22	0.53	0.31
OChem EFG+	BR	4	0.02	0.83	0.20	0.50	0.28
OChem EFG+	CC	2	0.03	0.78	0.24	0.55	0.33
OChem EFG+	CC	4	0.02	0.81	0.20	0.51	0.29
OChem EFG+	LP	2	0.04	0.67	0.37	0.36	0.37
OChem EFG+	LP	4	0.03	0.69	0.35	0.34	0.34
Pattern	BR	2	0.03	0.85	0.15	0.66	0.24
Pattern	BR	4	0.02	0.89	0.12	0.60	0.20
Pattern	CC	2	0.03	0.85	0.16	0.65	0.25
Pattern	CC	4	0.02	0.88	0.13	0.61	0.21
Pattern	LP	2	0.04	0.68	0.34	0.35	0.34
Pattern	LP	4	0.03	0.71	0.31	0.32	0.31
RDKit	BR	2	0.03	0.91	0.09	0.65	0.16
RDKit	BR	4	0.02	0.93	0.07	0.59	0.13
RDKit	CC	2	0.03	0.91	0.09	0.63	0.16
RDKit	CC	4	0.02	0.93	0.08	0.63	0.14
RDKit	LP	2	0.04	0.77	0.25	0.25	0.25
RDKit	LP	4	0.03	0.79	0.23	0.23	0.23
Torsion	BR	2	0.03	0.91	0.09	0.61	0.16
Torsion	BR	4	0.02	0.93	0.07	0.53	0.13
Torsion	CC	2	0.03	0.91	0.09	0.59	0.15

Torsion	CC	4	0.02	0.93	0.07	0.56	0.13
Torsion	LP	2	0.04	0.74	0.27	0.26	0.27
Torsion	LP	4	0.03	0.77	0.24	0.23	0.24

Problem Transformation (PT) and Adapted Algorithm (AA) Comparison (Only Level-2 Labels - RF)

Molecular Representation	Approach	Classifier	Hamming Loss	0-1 Loss	Micro Recall	Micro Precision	Micro F1-score
Atom-pair	AA	MLkNN	0.03	0.81	0.19	0.44	0.27
Atom-pair	AA	MLkNN	0.03	0.86	0.15	0.53	0.23
Atom-pair	AA	MLkNN	0.03	0.88	0.13	0.59	0.21
Atom-pair	BR	RF	0.03	0.93	0.07	0.70	0.12
Atom-pair	CC	RF	0.03	0.93	0.07	0.70	0.13
Atom-pair	LP	RF	0.04	0.75	0.26	0.27	0.26
Avalon (1024-bit)	AA	MLkNN	0.03	0.79	0.22	0.44	0.29
Avalon (1024-bit)	AA	MLkNN	0.03	0.84	0.17	0.55	0.25
Avalon (1024-bit)	AA	MLkNN	0.03	0.85	0.15	0.57	0.24
Avalon (1024-bit)	BR	RF	0.03	0.84	0.16	0.69	0.26
Avalon (1024-bit)	CC	RF	0.03	0.83	0.17	0.69	0.28
Avalon (1024-bit)	LP	RF	0.04	0.65	0.36	0.37	0.37
Avalon (2048-bit)	AA	MLkNN	0.03	0.78	0.23	0.45	0.31
Avalon (2048-bit)	AA	MLkNN	0.03	0.82	0.19	0.55	0.28
Avalon (2048-bit)	AA	MLkNN	0.03	0.85	0.16	0.58	0.25
Avalon (2048-bit)	BR	RF	0.03	0.82	0.19	0.70	0.30
Avalon (2048-bit)	CC	RF	0.03	0.81	0.19	0.69	0.30
Avalon (2048-bit)	LP	RF	0.03	0.62	0.39	0.41	0.40
Avalon (256-bit)	AA	MLkNN	0.03	0.79	0.23	0.46	0.30
Avalon (256-bit)	AA	MLkNN	0.03	0.85	0.16	0.55	0.25
Avalon (256-bit)	AA	MLkNN	0.03	0.86	0.15	0.57	0.24
Avalon (256-bit)	BR	RF	0.03	0.89	0.12	0.69	0.20
Avalon (256-bit)	CC	RF	0.03	0.88	0.12	0.66	0.20
Avalon (256-bit)	LP	RF	0.04	0.70	0.31	0.32	0.32
Avalon (4096-bit)	AA	MLkNN	0.03	0.79	0.23	0.44	0.30
Avalon (4096-bit)	AA	MLkNN	0.03	0.82	0.18	0.53	0.27
Avalon (4096-bit)	AA	MLkNN	0.03	0.84	0.16	0.57	0.26

Avalon (4096-bit)	BR	RF	0.03	0.81	0.19	0.69	0.30
Avalon (4096-bit)	CC	RF	0.03	0.81	0.20	0.69	0.31
Avalon (4096-bit)	LP	RF	0.03	0.62	0.40	0.40	0.40
Avalon (640-bit)	AA	MLkNN	0.03	0.78	0.23	0.45	0.30
Avalon (640-bit)	AA	MLkNN	0.03	0.84	0.17	0.56	0.26
Avalon (640-bit)	AA	MLkNN	0.03	0.85	0.16	0.58	0.25
Avalon (640-bit)	BR	RF	0.03	0.85	0.15	0.66	0.25
Avalon (640-bit)	CC	RF	0.03	0.84	0.16	0.68	0.26
Avalon (640-bit)	LP	RF	0.04	0.66	0.35	0.36	0.36
Avalon (8192-bit)	AA	MLkNN	0.03	0.78	0.23	0.44	0.30
Avalon (8192-bit)	AA	MLkNN	0.03	0.82	0.19	0.55	0.29
Avalon (8192-bit)	AA	MLkNN	0.03	0.84	0.17	0.58	0.26
Avalon (8192-bit)	BR	RF	0.03	0.81	0.19	0.70	0.30
Avalon (8192-bit)	CC	RF	0.03	0.81	0.20	0.69	0.30
Avalon (8192-bit)	LP	RF	0.03	0.61	0.40	0.41	0.40
CDK Functional Group	AA	MLkNN	0.03	0.81	0.23	0.41	0.30
CDK Functional Group	AA	MLkNN	0.03	0.84	0.18	0.52	0.26
CDK Functional Group	AA	MLkNN	0.03	0.85	0.17	0.53	0.26
CDK Functional Group	BR	RF	0.03	0.84	0.18	0.54	0.27
CDK Functional Group	CC	RF	0.03	0.80	0.21	0.52	0.30
CDK Functional Group	LP	RF	0.04	0.68	0.37	0.36	0.37
ChemAxon Functional Group	AA	MLkNN	0.06	0.87	0.28	0.34	0.31
ChemAxon Functional Group	AA	MLkNN	0.04	0.88	0.16	0.48	0.23
ChemAxon Functional Group	AA	MLkNN	0.04	0.90	0.13	0.51	0.21
ChemAxon Functional Group	BR	RF	0.05	0.85	0.21	0.41	0.28
ChemAxon Functional Group	CC	RF	0.05	0.83	0.19	0.43	0.27
ChemAxon Functional Group	LP	RF	0.07	0.79	0.32	0.28	0.30
Chi Kappa Descriptors	AA	MLkNN	0.02	0.93	0.07	0.26	0.11
Chi Kappa Descriptors	AA	MLkNN	0.02	0.97	0.04	0.51	0.07
Chi Kappa Descriptors	AA	MLkNN	0.02	0.97	0.03	0.50	0.05
Chi Kappa Descriptors	BR	RF	0.02	0.97	0.03	0.51	0.06
Chi Kappa Descriptors	CC	RF	0.02	0.97	0.03	0.47	0.06
Chi Kappa Descriptors	LP	RF	0.04	0.86	0.14	0.15	0.15
Dompé	AA	MLkNN	0.03	0.86	0.20	0.31	0.24
Dompé	AA	MLkNN	0.02	0.89	0.11	0.50	0.18

Dompé	AA	MLkNN	0.02	0.89	0.11	0.52	0.18
Dompé	BR	RF	0.03	0.86	0.16	0.42	0.23
Dompé	CC	RF	0.02	0.84	0.16	0.46	0.24
Dompé	LP	RF	0.04	0.75	0.31	0.26	0.28
FeatMorgan (1024-bit) (Radius 1) (Binary)	AA	MLkNN	0.03	0.80	0.24	0.41	0.31
FeatMorgan (1024-bit) (Radius 1) (Binary)	AA	MLkNN	0.03	0.84	0.18	0.51	0.27
FeatMorgan (1024-bit) (Radius 1) (Binary)	AA	MLkNN	0.03	0.85	0.16	0.53	0.25
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	RF	0.03	0.86	0.15	0.57	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	RF	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	LP	RF	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 2) (Binary)	AA	MLkNN	0.03	0.79	0.23	0.40	0.29
FeatMorgan (1024-bit) (Radius 2) (Binary)	AA	MLkNN	0.03	0.84	0.17	0.52	0.26
FeatMorgan (1024-bit) (Radius 2) (Binary)	AA	MLkNN	0.03	0.86	0.15	0.55	0.23
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	RF	0.03	0.85	0.15	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	RF	0.03	0.85	0.16	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	LP	RF	0.04	0.64	0.37	0.38	0.38
FeatMorgan (1024-bit) (Radius 2) (Count)	AA	MLkNN	0.02	0.86	0.15	0.39	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	AA	MLkNN	0.02	0.91	0.10	0.50	0.16
FeatMorgan (1024-bit) (Radius 2) (Count)	AA	MLkNN	0.02	0.91	0.09	0.53	0.16
FeatMorgan (1024-bit) (Radius 2) (Count)	BR	RF	0.02	0.88	0.12	0.68	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	CC	RF	0.02	0.88	0.12	0.67	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	LP	RF	0.03	0.67	0.34	0.35	0.35
FeatMorgan (1024-bit) (Radius 3) (Binary)	AA	MLkNN	0.03	0.82	0.21	0.35	0.26
FeatMorgan (1024-bit) (Radius 3) (Binary)	AA	MLkNN	0.03	0.88	0.13	0.48	0.20
FeatMorgan (1024-bit) (Radius 3) (Binary)	AA	MLkNN	0.03	0.89	0.11	0.53	0.18
FeatMorgan (1024-bit) (Radius 3) (Binary)	BR	RF	0.03	0.89	0.10	0.71	0.18
FeatMorgan (1024-bit) (Radius 3) (Binary)	CC	RF	0.03	0.89	0.11	0.73	0.20
FeatMorgan (1024-bit) (Radius 3) (Binary)	LP	RF	0.04	0.67	0.34	0.36	0.35
Layered	AA	MLkNN	0.03	0.81	0.20	0.44	0.28
Layered	AA	MLkNN	0.03	0.86	0.15	0.54	0.23
Layered	AA	MLkNN	0.03	0.88	0.12	0.58	0.20
Layered	BR	RF	0.03	0.89	0.12	0.65	0.20
Layered	CC	RF	0.03	0.88	0.13	0.66	0.21
Layered	LP	RF	0.04	0.71	0.30	0.31	0.31
MACCS	AA	MLkNN	0.03	0.79	0.23	0.45	0.30

MACCS	AA	MLkNN	0.03	0.83	0.18	0.56	0.27
MACCS	AA	MLkNN	0.03	0.83	0.17	0.59	0.27
MACCS	BR	RF	0.03	0.84	0.17	0.65	0.27
MACCS	CC	RF	0.03	0.82	0.19	0.63	0.29
MACCS	LP	RF	0.04	0.65	0.36	0.37	0.37
Morgan	AA	MLkNN	0.03	0.81	0.22	0.38	0.28
Morgan	AA	MLkNN	0.03	0.85	0.15	0.54	0.24
Morgan	AA	MLkNN	0.03	0.86	0.14	0.56	0.23
Morgan	BR	RF	0.03	0.89	0.11	0.76	0.20
Morgan	CC	RF	0.03	0.88	0.12	0.76	0.21
Morgan	LP	RF	0.04	0.65	0.36	0.37	0.36
OChem EFG+	AA	MLkNN	0.03	0.79	0.26	0.40	0.31
OChem EFG+	AA	MLkNN	0.03	0.82	0.19	0.56	0.28
OChem EFG+	AA	MLkNN	0.03	0.83	0.18	0.59	0.28
OChem EFG+	BR	RF	0.03	0.80	0.22	0.53	0.31
OChem EFG+	CC	RF	0.03	0.78	0.24	0.55	0.33
OChem EFG+	LP	RF	0.04	0.67	0.37	0.36	0.37
Pattern	AA	MLkNN	0.03	0.83	0.18	0.41	0.25
Pattern	AA	MLkNN	0.03	0.88	0.13	0.54	0.2
Pattern	AA	MLkNN	0.03	0.90	0.10	0.56	0.18
Pattern	BR	RF	0.03	0.85	0.15	0.66	0.24
Pattern	CC	RF	0.03	0.85	0.16	0.65	0.25
Pattern	LP	RF	0.04	0.68	0.34	0.35	0.34
RDKit	AA	MLkNN	0.03	0.82	0.19	0.40	0.26
RDKit	AA	MLkNN	0.03	0.88	0.13	0.50	0.20
RDKit	AA	MLkNN	0.03	0.91	0.09	0.58	0.16
RDKit	BR	RF	0.03	0.91	0.09	0.65	0.16
RDKit	CC	RF	0.03	0.91	0.09	0.63	0.16
RDKit	LP	RF	0.04	0.77	0.25	0.25	0.25
Torsion	AA	MLkNN	0.03	0.84	0.18	0.34	0.23
Torsion	AA	MLkNN	0.03	0.89	0.11	0.50	0.18
Torsion	AA	MLkNN	0.03	0.91	0.09	0.55	0.16
Torsion	BR	RF	0.03	0.91	0.09	0.61	0.16
Torsion	CC	RF	0.03	0.91	0.09	0.59	0.15
Torsion	LP	RF	0.04	0.74	0.27	0.26	0.27

Problem Transformation (PT) Approaches Comparison (Only Level-2 Labels - RF and SVM)

Molecular Representation	Approach	Classifier	Hamming Loss	0-1 Loss	Micro Recall	Micro Precision	Micro F1-score
Atom-pair	BR	RF	0.03	0.93	0.07	0.70	0.12
Atom-pair	BR	SVM	0.03	0.88	0.14	0.38	0.21
Atom-pair	CC	RF	0.03	0.93	0.07	0.70	0.13
Atom-pair	CC	SVM	0.04	0.80	0.22	0.23	0.22
Atom-pair	LP	RF	0.04	0.75	0.26	0.27	0.26
Atom-pair	LP	SVM	0.04	0.74	0.29	0.28	0.28
Avalon (1024-bit)	BR	RF	0.03	0.84	0.16	0.69	0.26
Avalon (1024-bit)	BR	SVM	0.03	0.79	0.25	0.44	0.32
Avalon (1024-bit)	CC	RF	0.03	0.83	0.17	0.69	0.28
Avalon (1024-bit)	CC	SVM	0.04	0.71	0.32	0.34	0.33
Avalon (1024-bit)	LP	RF	0.04	0.65	0.36	0.37	0.37
Avalon (1024-bit)	LP	SVM	0.04	0.66	0.37	0.36	0.37
Avalon (2048-bit)	BR	RF	0.03	0.82	0.19	0.70	0.30
Avalon (2048-bit)	BR	SVM	0.03	0.76	0.34	0.39	0.37
Avalon (2048-bit)	CC	RF	0.03	0.81	0.19	0.69	0.30
Avalon (2048-bit)	CC	SVM	0.04	0.71	0.36	0.36	0.36
Avalon (2048-bit)	LP	RF	0.03	0.62	0.39	0.41	0.40
Avalon (2048-bit)	LP	SVM	0.04	0.64	0.38	0.38	0.38
Avalon (256-bit)	BR	RF	0.03	0.89	0.12	0.69	0.20
Avalon (256-bit)	BR	SVM	0.03	0.91	0.09	0.67	0.16
Avalon (256-bit)	CC	RF	0.03	0.88	0.12	0.66	0.20
Avalon (256-bit)	CC	SVM	0.04	0.80	0.20	0.21	0.20
Avalon (256-bit)	LP	RF	0.04	0.70	0.31	0.32	0.32
Avalon (256-bit)	LP	SVM	0.04	0.69	0.35	0.33	0.34
Avalon (4096-bit)	BR	RF	0.03	0.81	0.19	0.69	0.30
Avalon (4096-bit)	BR	SVM	0.04	0.77	0.37	0.36	0.37
Avalon (4096-bit)	CC	RF	0.03	0.81	0.20	0.69	0.31
Avalon (4096-bit)	CC	SVM	0.04	0.72	0.37	0.36	0.37
Avalon (4096-bit)	LP	RF	0.03	0.62	0.40	0.40	0.40
Avalon (4096-bit)	LP	SVM	0.04	0.65	0.38	0.37	0.38
Avalon (640-bit)	BR	RF	0.03	0.85	0.15	0.66	0.25

Avalon (640-bit)	BR	SVM	0.03	0.81	0.20	0.53	0.29
Avalon (640-bit)	CC	RF	0.03	0.84	0.16	0.68	0.26
Avalon (640-bit)	CC	SVM	0.04	0.72	0.29	0.30	0.30
Avalon (640-bit)	LP	RF	0.04	0.66	0.35	0.36	0.36
Avalon (640-bit)	LP	SVM	0.04	0.65	0.38	0.37	0.38
Avalon (8192-bit)	BR	RF	0.03	0.81	0.19	0.70	0.30
Avalon (8192-bit)	BR	SVM	0.04	0.76	0.38	0.37	0.37
Avalon (8192-bit)	CC	RF	0.03	0.81	0.20	0.69	0.30
Avalon (8192-bit)	CC	SVM	0.04	0.72	0.37	0.36	0.37
Avalon (8192-bit)	LP	RF	0.03	0.61	0.40	0.41	0.40
Avalon (8192-bit)	LP	SVM	0.04	0.63	0.39	0.38	0.38
CDK Functional Group	BR	RF	0.03	0.84	0.18	0.54	0.27
CDK Functional Group	BR	SVM	0.03	0.85	0.14	0.66	0.24
CDK Functional Group	CC	RF	0.03	0.80	0.21	0.52	0.30
CDK Functional Group	CC	SVM	0.03	0.76	0.23	0.45	0.31
CDK Functional Group	LP	RF	0.04	0.68	0.37	0.36	0.37
CDK Functional Group	LP	SVM	0.04	0.63	0.38	0.41	0.40
ChemAxon Functional Group	BR	RF	0.05	0.85	0.21	0.41	0.28
ChemAxon Functional Group	BR	SVM	0.04	0.85	0.14	0.69	0.23
ChemAxon Functional Group	CC	RF	0.05	0.83	0.19	0.43	0.27
ChemAxon Functional Group	CC	SVM	0.04	0.84	0.14	0.65	0.23
ChemAxon Functional Group	LP	RF	0.07	0.79	0.32	0.28	0.30
ChemAxon Functional Group	LP	SVM	0.06	0.73	0.34	0.34	0.34
Chi Kappa Descriptors	BR	RF	0.02	0.97	0.03	0.51	0.06
Chi Kappa Descriptors	BR	SVM	0.02	0.99	0.01	0.66	0.01
Chi Kappa Descriptors	CC	RF	0.02	0.97	0.03	0.47	0.06
Chi Kappa Descriptors	CC	SVM	0.04	0.94	0.06	0.07	0.06
Chi Kappa Descriptors	LP	RF	0.04	0.86	0.14	0.15	0.15
Chi Kappa Descriptors	LP	SVM	0.04	0.97	0.04	0.04	0.04
Dompé	BR	RF	0.03	0.86	0.16	0.42	0.23
Dompé	BR	SVM	0.02	0.84	0.16	0.59	0.25
Dompé	CC	RF	0.02	0.84	0.16	0.46	0.24
Dompé	CC	SVM	0.02	0.79	0.20	0.50	0.28
Dompé	LP	RF	0.04	0.75	0.31	0.26	0.28
Dompé	LP	SVM	0.03	0.69	0.34	0.34	0.34

FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	RF	0.03	0.86	0.15	0.57	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	SVM	0.03	0.85	0.15	0.67	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	RF	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	SVM	0.03	0.78	0.22	0.46	0.30
FeatMorgan (1024-bit) (Radius 1) (Binary)	LP	RF	0.04	0.68	0.35	0.35	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	LP	SVM	0.04	0.64	0.37	0.38	0.38
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	RF	0.03	0.85	0.15	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	SVM	0.03	0.81	0.21	0.49	0.29
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	RF	0.03	0.85	0.16	0.67	0.25
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	SVM	0.04	0.72	0.29	0.34	0.32
FeatMorgan (1024-bit) (Radius 2) (Binary)	LP	RF	0.04	0.64	0.37	0.38	0.38
FeatMorgan (1024-bit) (Radius 2) (Binary)	LP	SVM	0.04	0.66	0.37	0.36	0.36
FeatMorgan (1024-bit) (Radius 2) (Count)	BR	RF	0.02	0.88	0.12	0.68	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	BR	SVM	0.02	0.86	0.17	0.42	0.24
FeatMorgan (1024-bit) (Radius 2) (Count)	CC	RF	0.02	0.88	0.12	0.67	0.21
FeatMorgan (1024-bit) (Radius 2) (Count)	CC	SVM	0.03	0.78	0.24	0.29	0.27
FeatMorgan (1024-bit) (Radius 2) (Count)	LP	RF	0.03	0.67	0.34	0.35	0.35
FeatMorgan (1024-bit) (Radius 2) (Count)	LP	SVM	0.03	0.74	0.29	0.26	0.27
FeatMorgan (1024-bit) (Radius 3) (Binary)	BR	RF	0.03	0.89	0.10	0.71	0.18
FeatMorgan (1024-bit) (Radius 3) (Binary)	BR	SVM	0.03	0.85	0.18	0.40	0.25
FeatMorgan (1024-bit) (Radius 3) (Binary)	CC	RF	0.03	0.89	0.11	0.73	0.20
FeatMorgan (1024-bit) (Radius 3) (Binary)	CC	SVM	0.04	0.77	0.25	0.28	0.27
FeatMorgan (1024-bit) (Radius 3) (Binary)	LP	RF	0.04	0.67	0.34	0.36	0.35
FeatMorgan (1024-bit) (Radius 3) (Binary)	LP	SVM	0.04	0.70	0.32	0.32	0.32
Layered	BR	RF	0.03	0.89	0.12	0.65	0.20
Layered	BR	SVM	0.03	0.84	0.20	0.38	0.26
Layered	CC	RF	0.03	0.88	0.13	0.66	0.21
Layered	CC	SVM	0.04	0.77	0.27	0.26	0.26
Layered	LP	RF	0.04	0.71	0.30	0.31	0.31
Layered	LP	SVM	0.04	0.76	0.26	0.26	0.26
MACCS	BR	RF	0.03	0.84	0.17	0.65	0.27
MACCS	BR	SVM	0.03	0.88	0.11	0.73	0.20
MACCS	CC	RF	0.03	0.82	0.19	0.63	0.29
MACCS	CC	SVM	0.04	0.77	0.23	0.25	0.24
MACCS	LP	RF	0.04	0.65	0.36	0.37	0.37

MACCS	LP	SVM	0.04	0.63	0.39	0.39	0.39
Morgan	BR	RF	0.03	0.89	0.11	0.76	0.20
Morgan	BR	SVM	0.03	0.81	0.22	0.46	0.29
Morgan	CC	RF	0.03	0.88	0.12	0.76	0.21
Morgan	CC	SVM	0.04	0.73	0.29	0.31	0.30
Morgan	LP	RF	0.04	0.65	0.36	0.37	0.36
Morgan	LP	SVM	0.04	0.67	0.35	0.35	0.35
OChem EFG+	BR	RF	0.03	0.80	0.22	0.53	0.31
OChem EFG+	BR	SVM	0.03	0.78	0.22	0.64	0.33
OChem EFG+	CC	RF	0.03	0.78	0.24	0.55	0.33
OChem EFG+	CC	SVM	0.03	0.71	0.29	0.54	0.37
OChem EFG+	LP	RF	0.04	0.67	0.37	0.36	0.37
OChem EFG+	LP	SVM	0.04	0.60	0.43	0.42	0.42
Pattern	BR	RF	0.03	0.85	0.15	0.66	0.24
Pattern	BR	SVM	0.04	0.67	0.35	0.35	0.35
Pattern	CC	RF	0.03	0.85	0.16	0.65	0.25
Pattern	CC	SVM	0.03	0.82	0.21	0.49	0.29
Pattern	LP	RF	0.04	0.68	0.34	0.35	0.34
Pattern	LP	SVM	0.04	0.73	0.28	0.31	0.30
RDKit	BR	RF	0.03	0.91	0.09	0.65	0.16
RDKit	BR	SVM	0.04	0.87	0.18	0.27	0.22
RDKit	CC	RF	0.03	0.91	0.09	0.63	0.16
RDKit	CC	SVM	0.05	0.81	0.22	0.21	0.21
RDKit	LP	RF	0.04	0.77	0.25	0.25	0.25
RDKit	LP	SVM	0.04	0.78	0.25	0.24	0.25
Torsion	BR	RF	0.03	0.91	0.09	0.61	0.16
Torsion	BR	SVM	0.03	0.89	0.11	0.58	0.18
Torsion	CC	RF	0.03	0.91	0.09	0.59	0.15
Torsion	CC	SVM	0.04	0.83	0.17	0.24	0.20
Torsion	LP	RF	0.04	0.74	0.27	0.26	0.27
Torsion	LP	SVM	0.04	0.73	0.30	0.28	0.29

Final Model

Level-3 and Level-2 Label Datasets Comparison (Only PT Approaches with RF and SVM)

Molecular Representation	Level	Approach	Classifier	Micro Recall	Micro Precision	Micro F1-score
Avalon (1024-bit)	2	BR	RF	0.30	0.75	0.43
Avalon (1024-bit)	2	BR	SVM	0.24	0.63	0.35
Avalon (1024-bit)	2	CC	RF	0.30	0.75	0.43
Avalon (1024-bit)	2	CC	SVM	0.28	0.56	0.38
Avalon (1024-bit)	3	BR	RF	0.29	0.75	0.41
Avalon (1024-bit)	3	CC	RF	0.29	0.76	0.42
Avalon (1024-bit)	3	BR	SVM	0.24	0.62	0.35
Avalon (1024-bit)	3	CC	SVM	0.29	0.56	0.38
Avalon (2048-bit)	2	BR	RF	0.32	0.75	0.44
Avalon (2048-bit)	2	BR	SVM	0.29	0.62	0.40
Avalon (2048-bit)	2	CC	RF	0.32	0.75	0.45
Avalon (2048-bit)	2	CC	SVM	0.34	0.57	0.42
Avalon (2048-bit)	3	BR	RF	0.30	0.74	0.43
Avalon (2048-bit)	3	CC	RF	0.30	0.75	0.43
Avalon (2048-bit)	3	BR	SVM	0.31	0.58	0.40
Avalon (2048-bit)	3	CC	SVM	0.33	0.54	0.41
Avalon (4096-bit)	2	BR	RF	0.33	0.75	0.45
Avalon (4096-bit)	2	BR	SVM	0.34	0.59	0.44
Avalon (4096-bit)	2	CC	RF	0.32	0.75	0.45
Avalon (4096-bit)	2	CC	SVM	0.38	0.56	0.45
Avalon (4096-bit)	3	BR	RF	0.31	0.74	0.43
Avalon (4096-bit)	3	CC	RF	0.31	0.75	0.44
Avalon (4096-bit)	3	BR	SVM	0.35	0.55	0.43
Avalon (4096-bit)	3	CC	SVM	0.37	0.53	0.44
CDK Functional Group	2	BR	RF	0.25	0.57	0.35
CDK Functional Group	2	BR	SVM	0.14	0.68	0.23
CDK Functional Group	2	CC	RF	0.26	0.58	0.36
CDK Functional Group	2	CC	SVM	0.16	0.61	0.26

CDK Functional Group	3	BR	RF	0.23	0.55	0.32
CDK Functional Group	3	CC	RF	0.23	0.57	0.33
CDK Functional Group	3	BR	SVM	0.13	0.69	0.21
CDK Functional Group	3	CC	SVM	0.15	0.62	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	2	BR	RF	0.24	0.64	0.35
FeatMorgan (1024-bit) (Radius 1) (Binary)	2	BR	SVM	0.17	0.67	0.27
FeatMorgan (1024-bit) (Radius 1) (Binary)	2	CC	RF	0.25	0.65	0.36
FeatMorgan (1024-bit) (Radius 1) (Binary)	2	CC	SVM	0.21	0.60	0.32
FeatMorgan (1024-bit) (Radius 1) (Binary)	3	BR	RF	0.22	0.63	0.32
FeatMorgan (1024-bit) (Radius 1) (Binary)	3	CC	RF	0.23	0.66	0.34
FeatMorgan (1024-bit) (Radius 1) (Binary)	3	BR	SVM	0.16	0.68	0.26
FeatMorgan (1024-bit) (Radius 1) (Binary)	3	CC	SVM	0.20	0.60	0.30
FeatMorgan (1024-bit) (Radius 2) (Binary)	2	BR	RF	0.30	0.74	0.43
FeatMorgan (1024-bit) (Radius 2) (Binary)	2	BR	SVM	0.24	0.68	0.35
FeatMorgan (1024-bit) (Radius 2) (Binary)	2	CC	RF	0.31	0.75	0.44
FeatMorgan (1024-bit) (Radius 2) (Binary)	2	CC	SVM	0.28	0.61	0.39
FeatMorgan (1024-bit) (Radius 2) (Binary)	3	BR	RF	0.28	0.74	0.41
FeatMorgan (1024-bit) (Radius 2) (Binary)	3	CC	RF	0.29	0.75	0.41
FeatMorgan (2048-bit) (Radius 2) (Binary)	2	BR	RF	0.31	0.75	0.43
FeatMorgan (2048-bit) (Radius 2) (Binary)	2	BR	SVM	0.28	0.68	0.40
FeatMorgan (2048-bit) (Radius 2) (Binary)	2	CC	RF	0.32	0.75	0.45
FeatMorgan (2048-bit) (Radius 2) (Binary)	2	CC	SVM	0.32	0.62	0.42
FeatMorgan (2048-bit) (Radius 2) (Binary)	3	BR	RF	0.29	0.74	0.41
FeatMorgan (2048-bit) (Radius 2) (Binary)	3	CC	RF	0.29	0.75	0.42
FeatMorgan (2048-bit) (Radius 2) (Binary)	3	BR	SVM	0.28	0.66	0.39
FeatMorgan (2048-bit) (Radius 2) (Binary)	3	CC	SVM	0.31	0.60	0.41
MACCS	2	BR	RF	0.26	0.68	0.38
MACCS	2	BR	SVM	0.13	0.70	0.23
MACCS	2	CC	RF	0.27	0.68	0.39
MACCS	2	CC	SVM	0.19	0.58	0.28
MACCS	3	BR	RF	0.25	0.67	0.36
MACCS	3	CC	RF	0.25	0.69	0.37
MACCS	3	BR	SVM	0.13	0.72	0.22
MACCS	3	CC	SVM	0.19	0.59	0.28
OChem EFG+	2	BR	RF	0.28	0.55	0.37

OChem EFG+	2	CC	RF	0.29	0.59	0.39
OChem EFG+	3	BR	RF	0.26	0.53	0.35
OChem EFG+	3	CC	RF	0.26	0.57	0.36
OChem EFG+	3	BR	SVM	0.20	0.69	0.31
OChem EFG+	3	CC	SVM	0.23	0.63	0.33

Problem Transformation (PT) and Ensemble Method (EM) Comparison (Only Level-3 Labels - RF and SVM)

Molecular Representation	Approach	Classifier	Micro Recall	Micro Precision	Micro F1-score
Avalon (1024-bit)	BR	RF	0.29	0.75	0.41
Avalon (1024-bit)	BR	SVM	0.24	0.62	0.35
Avalon (1024-bit)	CC	RF	0.29	0.76	0.42
Avalon (1024-bit)	CC	SVM	0.29	0.56	0.38
Avalon (1024-bit)	RAkELd	RF	0.29	0.74	0.41
Avalon (1024-bit)	RAkELd	SVM	0.22	0.70	0.33
Avalon (1024-bit)	RAkELo	RF	0.27	0.80	0.40
Avalon (1024-bit)	RAkELo	SVM	0.21	0.73	0.33
Avalon (2048-bit)	BR	RF	0.30	0.74	0.43
Avalon (2048-bit)	BR	SVM	0.31	0.58	0.40
Avalon (2048-bit)	CC	RF	0.30	0.75	0.43
Avalon (2048-bit)	CC	SVM	0.33	0.54	0.41
Avalon (2048-bit)	RAkELd	RF	0.30	0.74	0.43
Avalon (2048-bit)	RAkELd	SVM	0.27	0.68	0.39
Avalon (2048-bit)	RAkELo	RF	0.28	0.79	0.42
Avalon (2048-bit)	RAkELo	SVM	0.27	0.71	0.39
Avalon (4096-bit)	BR	RF	0.31	0.74	0.43
Avalon (4096-bit)	BR	SVM	0.35	0.55	0.43
Avalon (4096-bit)	CC	RF	0.31	0.75	0.44
Avalon (4096-bit)	CC	SVM	0.37	0.53	0.44
Avalon (4096-bit)	RAkELd	RF	0.31	0.74	0.43
Avalon (4096-bit)	RAkELd	SVM	0.32	0.65	0.43
Avalon (4096-bit)	RAkELo	RF	0.29	0.79	0.42
Avalon (4096-bit)	RAkELo	SVM	0.32	0.67	0.43
CDK Functional Group	BR	RF	0.23	0.55	0.32

CDK Functional Group	BR	SVM	0.13	0.69	0.21
CDK Functional Group	CC	RF	0.23	0.57	0.33
CDK Functional Group	CC	SVM	0.15	0.62	0.24
CDK Functional Group	RAkELd	RF	0.23	0.55	0.32
CDK Functional Group	RAkELd	SVM	0.11	0.71	0.19
CDK Functional Group	RAkELo	RF	0.21	0.61	0.31
CDK Functional Group	RAkELo	SVM	0.11	0.71	0.19
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	RF	0.22	0.63	0.32
FeatMorgan (1024-bit) (Radius 1) (Binary)	BR	SVM	0.16	0.68	0.26
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	RF	0.23	0.66	0.34
FeatMorgan (1024-bit) (Radius 1) (Binary)	CC	SVM	0.20	0.60	0.30
FeatMorgan (1024-bit) (Radius 1) (Binary)	RAkELd	RF	0.23	0.61	0.33
FeatMorgan (1024-bit) (Radius 1) (Binary)	RAkELd	SVM	0.14	0.69	0.24
FeatMorgan (1024-bit) (Radius 1) (Binary)	RAkELo	RF	0.20	0.69	0.31
FeatMorgan (1024-bit) (Radius 1) (Binary)	RAkELo	SVM	0.13	0.70	0.23
FeatMorgan (1024-bit) (Radius 2) (Binary)	BR	RF	0.28	0.74	0.41
FeatMorgan (1024-bit) (Radius 2) (Binary)	CC	RF	0.29	0.75	0.41
FeatMorgan (1024-bit) (Radius 2) (Binary)	RAkELd	RF	0.29	0.74	0.41
FeatMorgan (1024-bit) (Radius 2) (Binary)	RAkELd	SVM	0.21	0.72	0.33
FeatMorgan (1024-bit) (Radius 2) (Binary)	RAkELo	RF	0.27	0.79	0.40
FeatMorgan (1024-bit) (Radius 2) (Binary)	RAkELo	SVM	0.20	0.73	0.32
FeatMorgan (2048-bit) (Radius 2) (Binary)	BR	RF	0.29	0.74	0.41
FeatMorgan (2048-bit) (Radius 2) (Binary)	BR	SVM	0.28	0.66	0.39
FeatMorgan (2048-bit) (Radius 2) (Binary)	CC	RF	0.29	0.75	0.42
FeatMorgan (2048-bit) (Radius 2) (Binary)	CC	SVM	0.31	0.60	0.41
FeatMorgan (2048-bit) (Radius 2) (Binary)	RAkELd	RF	0.30	0.72	0.42
FeatMorgan (2048-bit) (Radius 2) (Binary)	RAkELd	SVM	0.25	0.71	0.37
FeatMorgan (2048-bit) (Radius 2) (Binary)	RAkELo	RF	0.28	0.78	0.41
FeatMorgan (2048-bit) (Radius 2) (Binary)	RAkELo	SVM	0.25	0.73	0.37
MACCS	BR	RF	0.25	0.67	0.36
MACCS	BR	SVM	0.13	0.72	0.22
MACCS	CC	RF	0.25	0.69	0.37
MACCS	CC	SVM	0.19	0.59	0.28
MACCS	RAkELd	RF	0.25	0.68	0.37
MACCS	RAkELd	SVM	0.11	0.74	0.19

MACCS	RAkELo	RF	0.23	0.73	0.35
MACCS	RAkELo	SVM	0.11	0.74	0.19
OChem EFG+	BR	RF	0.26	0.53	0.35
OChem EFG+	BR	SVM	0.20	0.69	0.31
OChem EFG+	CC	RF	0.26	0.57	0.36
OChem EFG+	CC	SVM	0.23	0.63	0.33
OChem EFG+	RAkELd	RF	0.27	0.51	0.35
OChem EFG+	RAkELd	SVM	0.18	0.70	0.29
OChem EFG+	RAkELo	RF	0.24	0.57	0.34
OChem EFG+	RAkELo	SVM	0.18	0.71	0.28

Bibliography

Afzal, A. M. *et al.* (2015) 'A multi-label approach to target prediction taking ligand promiscuity into account', *Journal of Cheminformatics*. BioMed Central, 7(1), p. 24. doi: 10.1186/s13321-015-0071-9.

Agatonovic-Kustrin, S. and Morton, D. (2016) 'Data Mining in Drug Discovery and Design', in *Artificial Neural Network for Drug Design, Delivery and Disposition*. Academic Press, pp. 181–193. doi: 10.1016/B978-0-12-801559-9.00009-0.

Ahlberg, E. *et al.* (2017) 'Current application of conformal prediction in drug discovery', *Annals of Mathematics and Artificial Intelligence*. Springer International Publishing, 81(1–2), pp. 145–154. doi: 10.1007/s10472-017-9550-1.

Akutsu, T. (2004) 'Efficient extraction of mapping rules of atoms from enzymatic reaction data', in *Journal of Computational Biology*, pp. 449–462. doi: 10.1089/1066527041410337.

Alano, C. C. *et al.* (2010) 'NAD⁺ depletion is necessary and sufficient for poly(ADP-ribose) polymerase-1-mediated neuronal death.', *The Journal of neuroscience : the official journal of the Society for Neuroscience*. Society for Neuroscience, 30(8), pp. 2967–78. doi: 10.1523/JNEUROSCI.5552-09.2010.

Alpaydin, E. (2010) *Introduction to machine learning*. MIT Press. Available at: <https://dl.acm.org/citation.cfm?id=1734076> (Accessed: 2 November 2017).

Altekar, G. *et al.* (2004) 'Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference', *Bioinformatics*. Oxford University Press, 20(3), pp. 407–415.

Altman, N. S. (1992) 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', *The American Statistician*, 46(3), pp. 175–185. doi: 10.1080/00031305.1992.10475879.

Alves, R. T., Delgado, M. R. and Freitas, A. A. (2008) 'Multi-label hierarchical classification of protein functions with artificial immune systems', in *Brazilian Symposium on Bioinformatics*. Springer, pp. 1–12.

Amé, J.-C., Spenlehauer, C. and de Murcia, G. (2004) 'The PARP superfamily', *BioEssays*. John Wiley & Sons, Ltd, 26(8), pp. 882–893. doi: 10.1002/bies.20085.

Aniceto, N. *et al.* (2016) 'Simultaneous Prediction of four ATP-binding Cassette Transporters' Substrates Using Multi-label QSAR', *Molecular informatics*. Wiley Online Library, 35(10), pp. 514–528.

Apostolakis, J. *et al.* (2008) 'Automatic determination of reaction mappings and reaction center information. 2. Validation on a biochemical reaction database', *Journal of Chemical Information and Modeling*, 48(6), pp. 1190–1198. doi: 10.1021/ci700433d.

Arens, J. F. (1979) 'A formalism for the classification and design of organic reactions. I. The class of (– +)_n reactions', *Journal of the Royal Netherlands Chemical Society*. WILEY-VCH Verlag, 98(4), pp. 155–161. doi: 10.1002/recl.19790980403.

Arita, M. (2003) 'In silico atomic tracing by substrate-product relationships in Escherichia coli intermediary metabolism.', *Genome research*, 13(11), pp. 2455–66. doi: 10.1101/gr.1212003.

Arús-Pous, J. *et al.* (2019) 'Exploring the GDB-13 chemical space using deep generative models', *Journal of cheminformatics*. BioMed Central, 11(1), p. 20.

Awale, M. *et al.* (2019) 'Drug Analogs from Fragment-Based Long Short-Term Memory Generative Neural Networks', *Journal of chemical information and modeling*. ACS Publications, 59(4), pp. 1347–1356.

Baell, J. B. and Holloway, G. A. (2010) 'New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays', *Journal of Medicinal Chemistry*. American Chemical Society, 53(7), pp. 2719–2740. doi: 10.1021/jm901137j.

Balaban, A. T. (1967) 'Chemical graphs. 3. Reactions with cyclic 6-membered transition states', *Revue Roumaine de Chimie*, 12, pp. 875–902.

Barnard, J. M. and Downs, G. M. (1992) 'Clustering of chemical structures on the basis of two-dimensional similarity measures', *Journal of Chemical Information and Computer Sciences*, 32(6), pp. 644–649. doi: 10.1021/ci00010a010.

Barutcuoglu, Z., Schapire, R. E. and Troyanskaya, O. G. (2006) 'Hierarchical multi-label prediction of gene function', *Bioinformatics*. Oxford University Press, 22(7), pp. 830–836.

Bauer, J. *et al.* (1985) 'IGOR and computer assisted innovation in chemistry.', *Chimia*, 39(2), pp. 43–53.

Bawden, D. (1991) 'Classification of chemical reactions: potential, possibilities and continuing relevance', *Journal of Chemical Information and Modeling*, 31(2), pp. 212–216. doi: 10.1021/ci00002a006.

Baykoucheva, S. (2015) 'Managing research data: electronic laboratory notebooks (ELNs)', *Managing Scientific Information and Research Data*. Chandos Publishing, pp. 85–96. doi: 10.1016/B978-0-08-100195-0.00009-3.

Baylon, J. L. *et al.* (2019) 'Enhancing retrosynthetic reaction prediction with deep learning using multiscale reaction classification', *Journal of chemical information and modeling*. ACS Publications, 59(2), pp. 673–688.

Becker, E. I. (1961) 'Synthetic methods of organic chemistry. Volume 14 (Theilheimer, W.; Karger, S.)', *Journal of Chemical Education*. American Chemical Society, 38(6), p. 330. doi: 10.1021/ed038p330.1.

Besnard, J. *et al.* (2012) 'Automated design of ligands to polypharmacological profiles.', *Nature*, 492(7428), pp. 215–220. doi: 10.1038/nature11691.

Bickerton, G. R. *et al.* (2012) 'Quantifying the chemical beauty of drugs', *Nature chemistry*. Nature Publishing Group, 4(2), p. 90.

Bishop, M. . C. M. (2006) *Pattern recognition and machine learning*. Springer. Available

at: <https://dl.acm.org/citation.cfm?id=1162264> (Accessed: 2 November 2017).

Blaschke, T. *et al.* (2018) 'Application of generative autoencoder in de novo molecular design', *Molecular informatics*. Wiley Online Library, 37(1–2), p. 1700123.

Blum, L. C. and Reymond, J.-L. (2009) '970 million druglike small molecules for virtual screening in the chemical universe database GDB-13', *Journal of the American Chemical Society*. ACS Publications, 131(25), pp. 8732–8733.

Blurock, E. S. (1990) 'Computer-aided synthesis design at RISC-Linz: automatic extraction and use of reaction classes', *Journal of Chemical Information and Computer Sciences*. American Chemical Society, 30(4), pp. 505–510. doi: 10.1021/ci00068a024.

Bohacek, R. *et al.* (1999) 'Growmol, a de novo computer program, and its application to thermolysin and pepsin: results of the design and synthesis of a novel inhibitor', in *Rational drug design*. Springer, pp. 103–114.

Bohacek, R. S., McMartin, C. and Guida, W. C. (1996) 'The art and practice of structure-based drug design: A molecular modeling perspective', *Medicinal Research Reviews*. Wiley Subscription Services, Inc., A Wiley Company, 16(1), pp. 3–50. doi: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6.

Böhm, H.-J. (1992a) 'LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads', *Journal of computer-aided molecular design*. Springer, 6(6), pp. 593–606.

Böhm, H.-J. (1992b) 'The computer program LUDI: a new method for the de novo design of enzyme inhibitors', *Journal of computer-aided molecular design*. Springer, 6(1), pp. 61–78.

Boiten, J.-W., Ott, M. A. and Noordik, J. H. (1995) 'Automated Overlap Analysis of Reaction Databases', *Journal of Chemical Information and Computer Sciences*, 35(1), pp. 115–120. doi: 10.1021/ci00023a017.

Boström, J. *et al.* (2018) 'Expanding the medicinal chemistry synthetic toolbox', *Nature Reviews Drug Discovery*. Nature Publishing Group, 17(10), pp. 709–727. doi: 10.1038/nrd.2018.116.

Boutell, M. R. *et al.* (2004) 'Learning multi-label scene classification', *Pattern Recognition*. Pergamon, 37(9), pp. 1757–1771. doi: 10.1016/J.PATCOG.2004.03.009.

Breiman, L. (2001) 'Random Forests', *Machine learning*. Springer, 45(1), pp. 5–32.

Brenk, R. *et al.* (2008) 'Lessons Learnt from Assembling Screening Libraries for Drug Discovery for Neglected Diseases', *ChemMedChem*. John Wiley & Sons, Ltd, 3(3), pp. 435–444. doi: 10.1002/cmdc.200700139.

Brinker, K., Fürnkranz, J. and Hüllermeier, E. (2006) 'A Unified Model for Multilabel Classification and Ranking', *Proceedings of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence*, pp. 489–493. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.8504> (Accessed: 31 May 2019).

Brinker, K. and Neubauer, C. (2010) 'System and method for case-based multilabel classification and ranking'. Google Patents.

Broughton, H., Hunt, P. and MacKey, M. (2003) 'Methods for classifying and searching chemical reactions'. Google Patents. Available at: <https://www.google.com/patents/US20030182094>.

Brown, D. G. and Boström, J. (2016) 'Analysis of Past and Present Synthetic Methodologies on Medicinal Chemistry: Where Have All the New Reactions Gone?', *Journal of Medicinal Chemistry*, 59(10), pp. 4443–4458. doi: 10.1021/acs.jmedchem.5b01409.

Brown, F. K. (1998) 'Chemoinformatics: What is It and How does It Impact Drug Discovery?', *Annual Reports in Medicinal Chemistry*, 33, pp. 375–384.

Brown, N. *et al.* (2004) 'A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules', *Journal of chemical information and computer sciences*. ACS Publications, 44(3), pp. 1079–1087.

Brown, N. *et al.* (2018) 'Big Data in Drug Discovery', *Progress in Medicinal Chemistry*. Elsevier, 57, pp. 277–356. doi: 10.1016/BS.PMCH.2017.12.003.

Buntine, W. and Niblett, T. (1992) 'A further comparison of splitting rules for decision-tree induction', *Machine Learning*. Springer, 8(1), pp. 75–85.

Button, A. *et al.* (2019) 'Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis', *Nature Machine Intelligence*. Nature Publishing Group, 1(7), pp. 307–315.

Cafilisch, A., Miranker, A. and Karplus, M. (1993) 'Multiple copy simultaneous search and construction of ligands in binding sites: application to inhibitors of HIV-1 aspartic proteinase', *Journal of medicinal chemistry*. ACS Publications, 36(15), pp. 2142–2167.

Campbell, I. B., Macdonald, S. J. F. and Procopiou, P. A. (2018) 'Medicinal chemistry in drug discovery in big pharma: past, present and future', *Drug Discovery Today*. Elsevier Current Trends, 23(2), pp. 219–234. doi: 10.1016/J.DRUDIS.2017.10.007.

De Cao, N. and Kipf, T. (2018) 'MolGAN: An implicit generative model for small molecular graphs', *arXiv preprint arXiv:1805.11973*.

Carey, J. S. *et al.* (2006) 'Analysis of the reactions used for the preparation of drug candidate molecules', *Organic & Biomolecular Chemistry*, 4(12), p. 2337. doi: 10.1039/b602413k.

Carhart, R. E., Smith, D. H. and Venkataraghavan, R. (1985) 'Atom pairs as molecular features in structure-activity studies: definition and applications', *Journal of Chemical Information and Computer Sciences*, 25(2), pp. 64–73. doi: 10.1021/ci00046a002.

Carlsson, L., Bendtsen, C. and Ahlberg, E. (2017) *Comparing Performance of Different Inductive and Transductive Conformal Predictors Relevant to Drug Discovery*, *Proceedings of Machine Learning Research*. Conformal and Probabilistic Prediction and Applications. Available at: <http://proceedings.mlr.press/v60/carlsson17a/carlsson17a.pdf> (Accessed: 10 October 2018).

ChemAxon Ltd. (2015) *Predefined Functional Groups and Named Molecule Groups*. Available at:

<https://docs.chemaxon.com/display/docs/Predefined+Functional+Groups+and+Named+Molecule+Groups>.

Chemical Abstract Service (1988) *Reactions - CASREACT - Answers to your chemical reaction questions*. Available at: <https://www.cas.org/content/reactions>.

Chemical Computing Group ULC and ULC, C. C. G. (2019) 'Molecular Operating Environment (MOE)'.

Chen, H. *et al.* (2018) 'The rise of deep learning in drug discovery', *Drug Discovery Today*. doi: 10.1016/j.drudis.2018.01.039.

Chen, I. J., Yin, D. and MacKerell, A. D. (2002) 'Combined *ab initio*/empirical approach for optimization of Lennard-Jones parameters for polar-neutral compounds', *Journal of Computational Chemistry*. Wiley Periodicals, Inc., 23(2), pp. 199–213. doi: 10.1002/jcc.1166.

Chen, L. (2003) 'Reaction Classification and Knowledge Acquisition', in *Handbook of Chemoinformatics*. Wiley-VCH Verlag GmbH, pp. 348–390. doi: 10.1002/9783527618279.ch12.

Chen, L. and Gasteiger, J. (1996) 'Organic Reactions Classified by Neural Networks: Michael Additions, Friedel–Crafts Alkylations by Alkenes, and Related Reactions', *Angewandte Chemie International Edition in English*. Hüthig & Wepf Verlag, 35(7), pp. 763–765. doi: 10.1002/anie.199607631.

Chen, W. L., Chen, D. Z. and Taylor, K. T. (2013) 'Automatic reaction mapping and reaction center detection', *Wiley Interdisciplinary Reviews: Computational Molecular Science*. John Wiley & Sons, Inc., 3(6), pp. 560–593. doi: 10.1002/wcms.1140.

Chodosh, D. F. *et al.* (2010) 'SYNthesis LIBrary, an expert system for chemical-reaction knowledge-base management', *Recueil des Travaux Chimiques des Pays-Bas*. WILEY-VCH Verlag, 111(6), pp. 247–254. doi: 10.1002/recl.19921110602.

Chou, K.-C., Wu, Z.-C. and Xiao, X. (2011) 'iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins', *PloS one*. Public Library of Science, 6(3), p. e18258.

Churchwell, C. J. *et al.* (2004) 'The signature molecular descriptor: 3. Inverse-quantitative structure–activity relationship of ICAM-1 inhibitory peptides', *Journal of Molecular Graphics and Modelling*. Elsevier, 22(4), pp. 263–273.

Clare, A. and King, R. D. (2001) 'Knowledge Discovery in Multi-label Phenotype Data', in. Springer, Berlin, Heidelberg, pp. 42–53. doi: 10.1007/3-540-44794-6_4.

Clark, A. M., Sarker, M. and Ekins, S. (2014) 'New target prediction and visualization tools incorporating open source molecular fingerprints for TB Mobile 2.0', *Journal of cheminformatics*. Nature Publishing Group, 6(1), p. 38.

Clark, D. E. *et al.* (1995) 'PRO-LIGAND: an approach to de novo molecular design. 1. Application to the design of organic molecules.', *Journal of computer-aided molecular design*, 9(1), pp. 13–32. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7751867> (Accessed: 11 May 2017).

Clark, R. D. *et al.* (2000) 'Visualizing substructural fingerprints', *Journal of Molecular*

Graphics and Modelling. Elsevier, 18(4–5), pp. 404–411.

Cohen, J. (1988) ‘Statistical power analysis for the social sciences’. Hillsdale, NJ: Erlbaum.

Coley, C. W. *et al.* (2017) ‘Prediction of Organic Reaction Outcomes Using Machine Learning’, *ACS Central Science*. American Chemical Society, 3(5), pp. 434–443. doi: 10.1021/acscentsci.7b00064.

Cook, S. A. (1971) ‘The complexity of theorem-proving procedures’, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing. STOC '71*. New York, NY, USA: ACM, pp. 151–158. doi: 10.1145/800157.805047.

Cortes, C. and Vapnik, V. (1995) ‘Support-Vector Networks’, *Machine Learning*, 20(3), pp. 273–297. doi: 10.1023/A:1022627411411.

Cox, O. B. *et al.* (2016) ‘A poised fragment library enables rapid synthetic expansion yielding the first reported inhibitors of PHIP(2), an atypical bromodomain’, *Chemical Science*. Royal Society of Chemistry, 7(3), pp. 2322–2330. doi: 10.1039/C5SC03115J.

Crabtree, J. D. and Mehta, D. P. (2009) ‘Automated reaction mapping’, *Journal of Experimental Algorithmics*. Association for Computing Machinery (ACM), 13. doi: 10.1145/1412228.1498697.

Dai, H. *et al.* (2018) ‘Syntax-directed variational autoencoder for structured data’, *arXiv preprint arXiv:1802.08786*.

Dalby, A. *et al.* (1992) ‘Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited’, *J Chem Inf Comput Sci*, 32. doi: 10.1021/ci00007a012.

Damewood, J. R., Lemian, C. L. and Masek, B. B. (2010) ‘NovoFLAP: A ligand-based de novo design approach for the generation of medicinally relevant ideas’, *Journal of Chemical Information and Modeling*, 50(7), pp. 1296–1303. doi: 10.1021/ci100080r.

Danziger, D. J. and Dean, P. M. (1989) ‘Automated Site-Directed Drug Design: A General Algorithm for Knowledge Acquisition about Hydrogen-Bonding Regions at Protein Surfaces’, *Proceedings of the Royal Society of London. Series B, Biological Sciences*. Royal Society, pp. 101–113. doi: 10.2307/2410614.

Darwin, C. (1859) *On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. H. Milford; Oxford University Press.

Daylight Chemical Information Systems Inc. (2019) *Daylight Chemical Information Systems, Inc.* Available at: <https://www.daylight.com/> (Accessed: 4 December 2019).

Dean, P. M. *et al.* (2006) ‘SkelGen: a general tool for structure-based de novo ligand design’, *Expert opinion on drug discovery*. Taylor & Francis, 1(2), pp. 179–189.

Degen, J. *et al.* (2008) ‘On the Art of Compiling and Using “Drug-Like” Chemical Fragment Spaces’, *ChemMedChem*. John Wiley & Sons, Ltd, 3(10), pp. 1503–1507. doi: 10.1002/cmdc.200800178.

Degen, J. and Rarey, M. (2006) 'FlexNovo: Structure-Based Searching in Large Fragment Spaces', *ChemMedChem*. WILEY-VCH Verlag, 1(8), pp. 854–868. doi: 10.1002/cmdc.200500102.

DeWitte, R. S. and Shakhnovich, E. I. (1996) 'SMoG: de novo design method based on simple, fast, and accurate free energy estimates. 1. Methodology and supporting evidence', *Journal of the American Chemical Society*. ACS Publications, 118(47), pp. 11733–11744.

Dey, F. and Caflisch, A. (2008) 'Fragment-Based de Novo Ligand Design by Multiobjective Evolutionary Optimization', *Journal of Chemical Information and Modeling*, 48(3), pp. 679–690. doi: 10.1021/ci700424b.

Diamond Light Source (2017) *Diamond Fragment Libraries*. Available at: <https://www.diamond.ac.uk/Instruments/Mx/Fragment-Screening/Fragment-Libraries.html> (Accessed: 15 January 2018).

Dittmar, P. G. *et al.* (1983) 'The CAS ONLINE search system. 1. General system design and selection, generation, and use of search screens', *Journal of Chemical Information and Computer Sciences*, 23(3), pp. 93–102. doi: 10.1021/ci00039a002.

Douguet, D. *et al.* (2005) 'LEA3D: a computer-aided ligand design for structure-based drug design', *Journal of medicinal chemistry*. ACS Publications, 48(7), pp. 2457–2468.

Douguet, D., Thoreau, E. and Grassy, G. (2000) 'A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm', *Journal of Computer-Aided Molecular Design*, 14(5), pp. 449–466. doi: 10.1023/A:1008108423895.

Doveston, R. G. *et al.* (2015) 'A unified lead-oriented synthesis of over fifty molecular scaffolds', *Organic & Biomolecular Chemistry*. The Royal Society of Chemistry, 13(3), pp. 859–865. doi: 10.1039/C4OB02287D.

Downs, G. M. *et al.* (1989) 'Review of ring perception algorithms for chemical graphs', *Journal of chemical information and computer sciences*. ACS Publications, 29(3), pp. 172–187.

Dugundji, James; Ugi, I., Dugundji, J. and Ugi, I. (1973) 'An algebraic model of constitutional chemistry as a basis for chemical computer programs BT -', in *Computers in Chemistry*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 19–64. doi: 10.1007/BFb0051317.

Durmus, S. *et al.* (2015) 'Breast Cancer Resistance Protein (BCRP/ABCG2) and P-glycoprotein (P-GP/ABCB1) Restrict Oral Availability and Brain Accumulation of the PARP Inhibitor Rucaparib (AG-014699)', *Pharmaceutical Research*, 32(1), pp. 37–46. doi: 10.1007/s11095-014-1442-z.

Durrant, J. D., Amaro, R. E. and McCammon, J. A. (2009) 'AutoGrow: a novel algorithm for protein inhibitor design', *Chemical biology & drug design*. Wiley Online Library, 73(2), pp. 168–178.

Durrant, J. D., Lindert, S. and McCammon, J. A. (2013) 'AutoGrow 3.0: an improved algorithm for chemically tractable, semi-automated protein inhibitor design', *Journal of Molecular Graphics and Modelling*. Elsevier, 44, pp. 104–112.

Dyson, G. M. . L. M. F. . M. H. L. (1968) 'A Modified IUPAC-Dyson Notation System for Chemical Structures', *Inform Storage Retrieval*, (4), pp. 27–83.

Eiblmaier, J. *et al.* (2002) 'Linking reaction information from different sources in 224th National Meeting of the American Chemical Society'. Boston.

Eisen, M. B. *et al.* (1994) 'HOOK: a program for finding novel molecular architectures that satisfy the chemical and steric requirements of a macromolecule binding site', *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library, 19(3), pp. 199–221.

Ekblad, T. *et al.* (2013) 'PARP inhibitors: polypharmacology versus selective inhibition', *FEBS Journal*. John Wiley & Sons, Ltd (10.1111), 280(15), pp. 3563–3575. doi: 10.1111/febs.12298.

Eklund, M. *et al.* (2015) 'The application of conformal prediction to the drug discovery process', *Annals of Mathematics and Artificial Intelligence*. Springer International Publishing, 74(1–2), pp. 117–132. doi: 10.1007/s10472-013-9378-2.

Elisseeff, A. and Weston, J. (2001) 'A Kernel Method for Multi-Labelled Classification', *In Advances in Neural Information Processing Systems*, 14, pp. 681–687. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2423> (Accessed: 31 May 2019).

Elsevier (2009) *Reaxys® - The shortest path to chemistry research answers*.

Englert, P. and Kovács, P. (2015) 'Efficient Heuristics for Maximum Common Substructure Search', *Journal of Chemical Information and Modeling*, 55(5), pp. 941–955. doi: 10.1021/acs.jcim.5b00036.

EPAM (2017) *Indigo Toolkit*. Available at: <http://lifescience.opensource.epam.com/indigo/>.

Ertl, P. (2003) 'Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups', *Journal of chemical information and computer sciences*. ACS Publications, 43(2), pp. 374–380.

Ertl, P. and Schuffenhauer, A. (2009) 'Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions', *Journal of Cheminformatics*. Nature Publishing Group, 1(1), pp. 1–11. doi: 10.1186/1758-2946-1-8.

Favre, H. A. and Powell, W. H. (2014) *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*. The Royal Society of Chemistry. doi: 10.1039/9781849733069.

Fechner, U. and Schneider, G. (2006) 'Flux (1): A Virtual Synthesis Scheme for Fragment-Based de Novo Design', *Journal of Chemical Information and Modeling*, 46(2), pp. 699–707. doi: 10.1021/ci0503560.

Fechner, U. and Schneider, G. (2007) 'Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design', *Journal of Chemical Information and Modeling*, 47(2), pp. 656–667. doi: 10.1021/ci6005307.

Ferraris, D. V. (2010) 'Evolution of Poly(ADP-ribose) Polymerase-1 (PARP-1) Inhibitors.

From Concept to Clinic', *Journal of Medicinal Chemistry*. American Chemical Society, 53(12), pp. 4561–4584. doi: 10.1021/jm100012m.

First, E. L., Gounaris, C. E. and Floudas, C. A. (2012) 'Stereochemically consistent reaction mapping and identification of multiple reaction mechanisms through integer linear optimization', *Journal of Chemical Information and Modeling*, 52(1), pp. 84–92. doi: 10.1021/ci200351b.

Fooshee, D., Andronico, A. and Baldi, P. (2013) 'ReactionMap: An efficient atom-mapping algorithm for chemical reactions', *Journal of Chemical Information and Modeling*, 53(11), pp. 2812–2819. doi: 10.1021/ci400326p.

Freeland, R. G. *et al.* (1979) 'The Chemical Abstracts Service Chemical Registry System. II. Augmented Connectivity Molecular Formula', *Journal of Chemical Information and Computer Sciences*, 19(2), pp. 94–98. doi: 10.1021/ci60018a012.

Freilich, J. Ouellette, L. (2019) 'Science fiction: Fictitious experiments in patents', *Science*, 364 (6445), pp. 1036–1037. doi: 10.1126/SCIENCE.AAX0748.

Freund, Y., Schapire, R. and Abe, N. (1999) 'A short introduction to boosting', *Journal-Japanese Society For Artificial Intelligence*. JAPANESE SOC ARTIFICIAL INTELL, 14(771–780), p. 1612.

Fujita, S. (1986) 'Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts', *Journal of Chemical Information and Modeling*. American Chemical Society, 26(4), pp. 205–212. doi: 10.1021/ci00052a009.

Funatsu, K. and Sasaki, S. I. (1988) 'Computer-assisted organic synthesis design and reaction prediction system, "AIPHOS"', *Tetrahedron Computer Methodology*, 1(1), pp. 27–37. doi: 10.1016/0898-5529(88)90006-1.

Garcia Soriano, F. *et al.* (2001) 'Diabetic endothelial dysfunction: the role of poly(ADP-ribose) polymerase activation', *Nature Medicine*. Nature Publishing Group, 7(1), pp. 108–113. doi: 10.1038/83241.

Gaspar, H. A. *et al.* (2015) 'GTM-Based QSAR Models and Their Applicability Domains', *Molecular informatics*. Wiley Online Library, 34(6-7), pp. 348–356.

Gasteiger, J. *et al.* (1987) 'A new treatment of chemical reactivity: Development of EROS, an expert system for reaction prediction and synthesis design', in *Organic Synthesis, Reactions and Mechanisms*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 19–73. doi: 10.1007/3-540-16904-0_14.

Gasteiger, J. *et al.* (1992) 'Similarity concepts for the planning of organic reactions and syntheses', *Journal of chemical information and computer sciences*. ACS Publications, 32(6), pp. 700–712.

Gasteiger, J. (2006) 'The central role of chemoinformatics', *Chemometrics and Intelligent Laboratory Systems*. Elsevier, 82(1–2), pp. 200–209.

Gasteiger, J. (2007) 'De novo design and synthetic accessibility', *Journal of Computer-Aided Molecular Design*. Kluwer Academic Publishers, 21(6), pp. 307–309. doi: 10.1007/s10822-007-9115-1.

Gasteiger, J., Ihlenfeldt, W. D. and Röse, P. (2010) 'A collection of computer methods for synthesis design and reaction prediction', *Recueil des Travaux Chimiques des Pays-Bas*. WILEY-VCH Verlag, 111(6), pp. 270–290. doi: 10.1002/recl.19921110605.

Gedeck, P., Bernhard, R. and Bartels, C. (2006) 'QSAR – How Good Is It in Practice? Comparison of Descriptor Sets on an Unbiased Cross Section of Corporate Data Sets', *Journal of Chemical Information and Modeling*. American Chemical Society. doi: 10.1021/CI050413P.

Gehlhaar, D. K. *et al.* (1995) 'De novo design of enzyme inhibitors by Monte Carlo ligand generation', *Journal of medicinal chemistry*. ACS Publications, 38(3), pp. 466–472.

Gelernter, H., Rose, J. R. and Chen, C. (1990) 'Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning', *Journal of Chemical Information and Computer Sciences*. American Chemical Society, 30(4), pp. 492–504. doi: 10.1021/ci00068a023.

Gerald M. Maggiora, M. A. J. (1992) 'Concepts and Applications of Molecular Similarity', *Journal of Molecular Structure*. Wiley, 269(3–4), pp. 376–377. doi: 10.1016/0022-2860(92)85011-5.

Ghose, A. K. *et al.* (2012) 'Knowledge-Based, Central Nervous System (CNS) Lead Selection and Lead Optimization for CNS Drug Discovery', *ACS Chemical Neuroscience*. American Chemical Society, 3(1), pp. 50–68. doi: 10.1021/cn200100h.

Gillet, V. *et al.* (1993) 'SPROUT: A program for structure generation', *Journal of Computer-Aided Molecular Design*, 7(2), pp. 127–153. doi: 10.1007/BF00126441.

Gillet, V. J. *et al.* (1995) 'SPROUT, HIPPO and CAESA: Tools for de novo structure generation and estimation of synthetic accessibility', *Perspectives in Drug Discovery and Design*, 3(1), pp. 34–50. doi: 10.1007/BF02174466.

Gillet, V. J., Bodkin, M. J. and Hristozov, D. (2013) 'Multiobjective De Novo Design of Synthetically Accessible Compounds', in *De novo Molecular Design*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 267–285. doi: 10.1002/9783527677016.ch11.

Glen, R. C. and Payne, A. W. (1995) 'A genetic algorithm for the automated generation of molecules within constraints.', *Journal of computer-aided molecular design*, 9(2), pp. 181–202. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7608749> (Accessed: 11 May 2017).

Gobbi, A. and Poppinger, D. (1998) 'Genetic optimization of combinatorial libraries', *Biotechnology and Bioengineering*. Wiley-Blackwell, 61(1), pp. 47–54. doi: 10.1002/(SICI)1097-0290(199824)61:1<47::AID-BIT9>3.0.CO;2-Z.

Godbole, S. and Sarawagi, S. (2004) 'Discriminative Methods for Multi-labeled Classification', in Springer, Berlin, Heidelberg, pp. 22–30. doi: 10.1007/978-3-540-24775-3_5.

Gohlke, H., Hendlich, M. and Klebe, G. (2000) 'Knowledge-based scoring function to predict protein-ligand interactions', *Journal of molecular biology*. Elsevier, 295(2), pp. 337–356.

Goldberg, D. E. and Holland, J. H. (1988) 'Genetic algorithms and machine learning'.

Kluwer Academic Publishers-Plenum Publishers; Kluwer Academic Publishers ...

Gómez-Bombarelli, R. *et al.* (2018) 'Automatic chemical design using a data-driven continuous representation of molecules', *ACS central science*. ACS Publications, 4(2), pp. 268–276.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*. MIT press.

de Gooijer, M. C. *et al.* (2018) 'ABCB1 Attenuates the Brain Penetration of the PARP Inhibitor AZD2461', *Molecular Pharmaceutics*, 15(11), pp. 5236–5243. doi: 10.1021/acs.molpharmaceut.8b00742.

Grethe, G. *et al.* (2018) 'International chemical identifier for reactions (RInChI)', *Journal of Cheminformatics*. Springer, 10(1). doi: 10.1186/s13321-018-0277-8.

Grethe, G. and Moock, T. E. (1990) 'Similarity searching in REACCS. A new tool for the synthetic chemist', *Journal of Chemical Information and Modeling*, 30(4), pp. 511–520. doi: 10.1021/ci00068a025.

Guimaraes, G. L. *et al.* (2017) 'Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models', *arXiv preprint arXiv:1705.10843*.

Guo, M., Liu, Y. and Malec, J. (2004) 'A New Q-Learning Algorithm Based on the Metropolis Criterion', *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(5), pp. 2140–2143. doi: 10.1109/TSMCB.2004.832154.

Gupta, A. *et al.* (2018) 'Generative Recurrent Networks for De Novo Drug Design', *Molecular Informatics*, 37(1). doi: 10.1002/minf.201700111.

Hall, L. H. and Kier, L. B. (1991) 'The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling', in: Wiley-Blackwell, pp. 367–422. doi: 10.1002/9780470125793.ch9.

Hall, R. J., Murray, C. W. and Verdonk, M. L. (2017) 'The Fragment Network: A Chemistry Recommendation Engine Built Using a Graph Database', *Journal of medicinal chemistry*. ACS Publications, 60(14), pp. 6440–6450.

Hann, M. *et al.* (1999) 'Strategic Pooling of Compounds for High-Throughput Screening', *Journal of Chemical Information and Computer Sciences*, 39(5), pp. 897–902. doi: 10.1021/ci990423o.

Hansch, C. *et al.* (1962) 'Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients', *Nature*. Nature Publishing Group, 194(4824), p. 178.

Harrison, J. M. and Lynch, M. F. (1970) 'Computer analysis of chemical reactions for storage and retrieval', *J. Chem. Soc. C*. The Royal Society of Chemistry, (15), pp. 2082–2087. doi: 10.1039/J39700002082.

Hartenfeller, M. *et al.* (2008) 'Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization', *Chemical biology & drug design*. Wiley Online Library, 72(1), pp. 16–26.

Hartenfeller, M. *et al.* (2012) 'Dogs: Reaction-driven de novo design of bioactive compounds', *PLoS Computational Biology*, 8(2). doi: 10.1371/journal.pcbi.1002380.

Hartenfeller, M., Renner, S. and Jacoby, E. (2013) 'Reaction-Driven De Novo Design: a Keystone for Automated Design of Target Family-Oriented Libraries', in *De novo Molecular Design*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 245–266. doi: 10.1002/9783527677016.ch10.

Hartenfeller, M. and Schneider, G. (2011) 'Enabling future drug discovery by de novo design', *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5), pp. 742–759. doi: 10.1002/wcms.49.

Hazai, E. *et al.* (2013) 'Predicting substrates of the human breast cancer resistance protein using a support vector machine method', *BMC Bioinformatics*, 14(1), p. 130. doi: 10.1186/1471-2105-14-130.

Head, R. D. *et al.* (1996) 'VALIDATE: A new method for the receptor-based prediction of binding affinities of novel ligands', *Journal of the American Chemical Society*. ACS Publications, 118(16), pp. 3959–3969.

Hecht, D. and Fogel, G. B. (2009) 'A novel in silico approach to drug discovery via computational intelligence', *Journal of Chemical Information and Modeling*, 49(4), pp. 1105–1121. doi: 10.1021/ci9000647.

Heinonen, M. *et al.* (2011) 'Computing atom mappings for biochemical reactions without subgraph isomorphism.', *Journal of computational biology: a journal of computational molecular cell biology*, 18(1), pp. 43–58. doi: 10.1089/cmb.2009.0216.

Heller, S. R. *et al.* (2015) 'InChI, the IUPAC International Chemical Identifier', *Journal of Cheminformatics*, 7(1), p. 23. doi: 10.1186/s13321-015-0068-4.

Hendrickson, J. B. (1997) 'Comprehensive system for classification and nomenclature of organic reactions', *Journal of chemical information and computer sciences*. ACS Publications, 37(5), pp. 852–860.

Hendrickson, J. B. (2010) 'Systematic Signatures for Organic Reactions', *Journal of Chemical Information and Modeling*, 50(8), pp. 1319–1329. doi: 10.1021/ci1000482.

Hendrickson, J. B. and Miller, T. M. (1990) 'Reaction indexing for reaction databases', *Journal of chemical information and computer sciences*. ACS Publications, 30(4), pp. 403–408.

Hendrickson, J. B. and Sander, T. (1995) 'COGNOS: A Beilstein-Type System for Organizing Organic Reactions', *Journal of Chemical Information and Computer Sciences*, 35(2), pp. 251–260. doi: 10.1021/ci00024a015.

Herges, R. (1994) 'Coarctate transition states: the discovery of a reaction principle', *Journal of Chemical Information and Computer Sciences*, 34(1), pp. 91–102. doi: 10.1021/ci00017a011.

Herges, R. and Hoock, C. (1992) 'Reaction Planning: Computer-Aided Discovery of a Novel Elimination Reaction', *Science*, 255(5045), pp. 711 LP – 713. Available at:

<http://science.sciencemag.org/content/255/5045/711.abstract>.

Hiss, J. A., Hartenfeller, M. and Schneider, G. (2010) 'Concepts and Applications of "Natural Computing" Techniques in De Novo Drug and Peptide Design', *Current Pharmaceutical Design*, 16(15), pp. 1656–1665.

Ho, C. M. W. and Marshall, G. R. (1993) 'SPLICE: A program to assemble partial query solutions from three-dimensional database searches into novel ligands', *Journal of Computer-Aided Molecular Design*, 7(6), pp. 623–647. doi: 10.1007/BF00125322.

Ho, T. K. (1995) 'Random Decision Forests', in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. Washington, DC, USA: IEEE Computer Society (ICDAR '95), pp. 278--. Available at: <http://dl.acm.org/citation.cfm?id=844379.844681>.

Ho, T. K. (1998) 'The random subspace method for constructing decision forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), pp. 832–844. doi: 10.1109/34.709601.

Hodes, L. (1976) 'Selection of Descriptors According to Discrimination and Redundancy. Application to Chemical Structure Searching', *Journal of Chemical Information and Computer Sciences*, 16(2), pp. 88–93. doi: 10.1021/ci60006a012.

Holm, B. E. (1969) *Chemical Structure Information Handling*. Washington, D.C.: National Academies Press. doi: 10.17226/21566.

Holt, N. J. *et al.* (2012) *Psychology: The science of mind and behaviour*. McGraw-Hill Education.

Hristozov, D. *et al.* (2011) 'Validation of Reaction Vectors for de Novo Design', in Bienstock, R. J. (ed.) *Library Design, Search Methods, and Applications of Fragment-Based Drug Design*, pp. 29–43. doi: 10.1021/bk-2011-1076.ch002.

Hristozov, D., Gasteiger, J. and B. Da Costa, F. (2007) 'Multilabeled Classification Approach To Find a Plant Source for Terpenoids', *Journal of Chemical Information and Modeling*, 48(1), pp. 56–67. doi: 10.1021/ci700175m.

Hu, Q.-N. *et al.* (2012) 'Assignment of EC numbers to enzymatic reactions with reaction difference fingerprints', *PloS one*. Public Library of Science, 7(12), p. e52901.

Huang, Q., Li, L.-L. and Yang, S.-Y. (2010) 'PhDD: A new pharmacophore-based de novo design method of drug-like molecules combined with assessment of synthetic accessibility', *Journal of Molecular Graphics and Modelling*, 28(8), pp. 775–787. doi: 10.1016/j.jmgm.2010.02.002.

Ikebata, H. *et al.* (2017) 'Bayesian molecular design with a chemical language model', *Journal of computer-aided molecular design*. Springer, 31(4), pp. 379–391.

InfoChem (1974) *SPRESI*. Available at: <http://www.infochem.de/products/databases/spresi.shtml>.

InfoChem (1996) *ChemReact*. Available at: <http://www.infochem.de/products/databases/chemreact41.shtml>.

Irsoy, O. and Cardie, C. (2014) ‘Opinion mining with deep recurrent neural networks’, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 720–728.

Ishchenko, A. *et al.* (2012) ‘Structure-based design technology contour and its application to the design of renin inhibitors’, *Journal of chemical information and modeling*. ACS Publications, 52(8), pp. 2089–2097.

Ishchenko, A. V and Shakhnovich, E. I. (2002) ‘Small molecule growth 2001 (SMoG2001): An improved knowledge-based scoring function for protein–ligand interactions’, *Journal of medicinal chemistry*. ACS Publications, 45(13), pp. 2770–2780.

Jaques, N. *et al.* (2017) ‘Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control’, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 1645–1654.

Jaspers, J. E. *et al.* (2015) ‘BRCA2-Deficient Sarcomatoid Mammary Tumors Exhibit Multidrug Resistance’, *Cancer Research*, 75(4), pp. 732–741. doi: 10.1158/0008-5472.CAN-14-0839.

Jawlik, A. A. (2016) *Statistics from A to Z: Confusing Concepts Clarified*. Hoboken, NJ, USA: John Wiley & Sons, Inc. doi: 10.1002/9781119272021.

Jaworski, W. *et al.* (2019) ‘Automatic mapping of atoms across both simple and complex chemical reactions’, *Nature Communications*. Nature Publishing Group, 10(1), p. 1434. doi: 10.1038/s41467-019-09440-2.

Jochum, C., Gasteiger, J. and Ugi, I. (1980) ‘The principle of minimum chemical distance (PMCD)’, *Angewandte Chemie International Edition in English*. Wiley Online Library, 19(7), pp. 495–505.

John Wiley and Sons (1999) *e-EROS Encyclopedia of Reagents for Organic Synthesis*. Available at: <http://onlinelibrary.wiley.com/book/10.1002/047084289X>.

Johnson, M. A. and Maggiora, G. M. (1990) *Concepts and applications of molecular similarity*. Wiley. Available at: <https://www.wiley.com/engb/Concepts+and+Applications+of+Molecular+Similarity-p-9780471621751> (Accessed: 17 October 2019).

Jones, G. *et al.* (1997) ‘Development and validation of a genetic algorithm for flexible docking’, *Journal of Molecular Biology*. Academic Press, 267(3), pp. 727–748. doi: 10.1006/JMBI.1996.0897.

Kadurin, A., Nikolenko, S., *et al.* (2017) ‘druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico’, *Molecular pharmaceutics*. ACS Publications, 14(9), pp. 3098–3104.

Kadurin, A., Aliper, A., *et al.* (2017) ‘The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology’, *Oncotarget*. Impact Journals, LLC, 8(7), p. 10883.

Kang, S. and Cho, K. (2019) ‘Conditional Molecular Design with Deep Generative Models’, *Journal of Chemical Information and Modeling*, 59(1). doi: 10.1021/acs.jcim.8b00263.

Kawai, K., Fujishima, S. and Takahashi, Y. (2008) ‘Predictive Activity Profiling of Drugs by Topological-Fragment-Spectra-Based Support Vector Machines’, *Journal of Chemical Information and Modeling*. American Chemical Society, 48(6), pp. 1152–1160. doi: 10.1021/ci7004753.

Kawai, K. and Takahashi, Y. (2009) ‘Identification of the Dual Action Antihypertensive Drugs Using TFS-Based Support Vector Machines’, *Chem-Bio Informatics Journal*, 9, pp. 41–51. doi: 10.1273/cbij.9.41.

Kelder, J. *et al.* (1999) ‘Polar Molecular Surface as a Dominating Determinant for Oral Absorption and Brain Penetration of Drugs’, *Pharmaceutical Research*. Kluwer Academic Publishers-Plenum Publishers, 16(10), pp. 1514–1519. doi: 10.1023/A:1015040217741.

King, G. *et al.* (2001) *Logistic Regression in Rare Events Data*. Available at: <http://gking.harvard.edu>. (Accessed: 4 October 2018).

Kireeva, N. *et al.* (2012) ‘Generative topographic mapping (GTM): universal tool for data visualization, structure-activity modeling and dataset comparison’, *Molecular informatics*. Wiley Online Library, 31(3-4), pp. 301–312.

Kraut, H. *et al.* (2013) ‘Algorithm for reaction classification’, *Journal of Chemical Information and Modeling*, 53(11), pp. 2884–2895. doi: 10.1021/ci400442f.

Kumar, A. and Maranas, C. D. (2014) ‘CLCA: Maximum common molecular substructure queries within the MetRxn database’, *Journal of Chemical Information and Modeling*. American Chemical Society, 54(12), pp. 3417–3438. doi: 10.1021/ci5003922.

Kumar, C., P.T.V., L. and Arunachalam, A. (2019) ‘Structure based pharmacophore study to identify possible natural selective PARP-1 trapper as anti-cancer agent’, *Computational Biology and Chemistry*. Elsevier, 80, pp. 314–323. doi: 10.1016/J.COMPBIOLCHEM.2019.04.018.

Kutchukian, P. S. *et al.* (2013) ‘Construction of Drug-Like Compounds by Markov Chains’, in *De novo Molecular Design*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 311–323. doi: 10.1002/9783527677016.ch13.

Kutchukian, P. S., Lou, D. and Shakhnovich, E. I. (2009) ‘FOG: Fragment Optimized Growth Algorithm for the de Novo Generation of Molecules Occupying Druglike Chemical Space’, *Journal of Chemical Information and Modeling*, 49(7), pp. 1630–1642. doi: 10.1021/ci9000458.

Kwon, Y. (2002) *Handbook of Essential Pharmacokinetics, Pharmacodynamics and Drug Metabolism for Industrial Scientists*. Springer US. doi: 10.1007/b112416.

Laggner, C. (2005) *SMARTS Patterns for Functional Group Classification*. Available at: https://github.com/cdk/cdk/blob/master/descriptor/fingerprint/src/main/resources/org/openscience/cdk/fingerprint/data/SMARTS_InteLigand.txt.

Landrum, G. (2016) *RDKit Documentation*. Available at: <https://www.rdkit.org/docs/GettingStartedInPython.html#fingerprinting-and-molecular-similarity>.

Latendresse, M. *et al.* (2012) ‘Accurate atom-mapping computation for biochemical reactions’, *Journal of Chemical Information and Modeling*, 52(11), pp. 2970–2982. doi: 10.1021/ci3002217.

Latino, D. A. R. S. and Aires-de-Sousa, J. (2006) ‘Genome-scale classification of metabolic reactions: a chemoinformatics approach’, *Angewandte Chemie International Edition*. Wiley Online Library, 45(13), pp. 2066–2069.

Leach, A. R. and Gillet, V. J. (2007) *An Introduction To Chemoinformatics*. Dordrecht: Springer Netherlands. doi: 10.1007/978-1-4020-6291-9.

Lessel, U. *et al.* (2009) ‘Searching Fragment Spaces with Feature Trees’, *Journal of Chemical Information and Modeling*, 49(2), pp. 270–279. doi: 10.1021/ci800272a.

Lewell, X. Q. *et al.* (1998) ‘RECAP - Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry’, *Journal of Chemical Information and Computer Sciences*, 38(3), pp. 511–522. doi: 10.1021/ci970429i.

Li, Y., Zhang, L. and Liu, Z. (2018) ‘Multi-objective de novo drug design with conditional graph generative model’, *Journal of Cheminformatics*, 10(1). doi: 10.1186/s13321-018-0287-6.

Liebeschuetz, J. W. *et al.* (2002) ‘PRO_SELECT: combining structure-based drug design and array-based chemistry for rapid lead discovery. 2. The development of a series of highly potent and selective factor Xa inhibitors’, *Journal of medicinal chemistry*. ACS Publications, 45(6), pp. 1221–1232.

Lim, J. *et al.* (2018) ‘Molecular generative model based on conditional variational autoencoder for de novo molecular design’, *Journal of cheminformatics*. Nature Publishing Group, 10(1), p. 31.

Lin, X. and Chen, X. (2010) ‘Mr. KNN: soft relevance for multi-label classification’, in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, pp. 349–358.

Lipinski, C. A. *et al.* (1997) ‘Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings’, *Advanced Drug Delivery Reviews*. Elsevier, 23(1–3), pp. 3–25. doi: 10.1016/s0169-409x(00)00129-0.

Litsa, E. E. *et al.* (2019) ‘Machine Learning Guided Atom Mapping of Metabolic Reactions’, *Journal of Chemical Information and Modeling*. American Chemical Society, 59(3), pp. 1121–1135. doi: 10.1021/acs.jcim.8b00434.

Liu, B. *et al.* (2017) ‘Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models’, *ACS Central Science*. American Chemical Society, 3(10), pp. 1103–1113. doi: 10.1021/acscentsci.7b00303.

Lloyd, D. G. *et al.* (2004) ‘Scaffold Hopping in De Novo Design. Ligand Generation in the Absence of Receptor Information’, *Journal of Medicinal Chemistry*, 47(3), pp. 493–496. doi: 10.1021/jm034222u.

Lo, Y. C. *et al.* (2018) ‘Machine learning in chemoinformatics and drug discovery’, *Drug Discovery Today*. doi: 10.1016/j.drudis.2018.05.010.

Lowe, D. (2017) ‘Chemical Reactions from US Patents (1976-Sep2016)’. doi: 10.6084/m9.figshare.5104873.v1.

de Luca, A. *et al.* (2012) ‘Mining Chemical Reactions Using Neighborhood Behavior and Condensed Graphs of Reactions Approaches’, *Journal of Chemical Information and Modeling*, 52(9), pp. 2325–2338. doi: 10.1021/ci300149n.

Luo, X. and Zincir-Heywood, A. N. (2005) ‘Evaluation of two systems on multi-class multi-label document classification’, in *International Symposium on Methodologies for Intelligent Systems*. Springer, pp. 161–169.

Luo, Z., Wang, R. and Lai, L. (1996) ‘RASSE: A New Method for Structure-Based Drug Design’, *Journal of Chemical Information and Computer Sciences*, 36(6), pp. 1187–1194. doi: 10.1021/ci950277w.

Lynch, M. F. and Willett, P. (1978) ‘The Automatic Detection of Chemical Reaction Sites’, *Journal of Chemical Information and Computer Sciences*, 18(3), pp. 154–159. doi: 10.1021/ci60015a009.

Mackerell, A. D. (2004) ‘Empirical force fields for biological macromolecules: Overview and issues’, *Journal of Computational Chemistry*. John Wiley & Sons, Inc., 25(13), pp. 1584–1604. doi: 10.1002/jcc.20082.

Maziarka, L. *et al.* (2019) ‘Mol-CycleGAN-a generative model for molecular optimization’, *arXiv preprint arXiv:1902.02119*.

McCallum, A. K. (1999) ‘Multi-label text classification with a mixture model trained by EM’, *Proceedings of the AAAI’ 99 Workshop on Text Learning*. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.888> (Accessed: 30 May 2019).

McGregor, J. J. and Willett, P. (1981) ‘Use of a maximum common subgraph algorithm in the automatic identification of ostensible bond changes occurring in chemical reactions’, *Journal of Chemical Information and Computer Sciences*, 21(3), pp. 137–140. doi: 10.1021/ci00031a005.

McPherson, A. and Gavira, J. A. (2014) ‘Introduction to protein crystallization’, *Acta Crystallographica Section F: Structural Biology Communications*. International Union of Crystallography, 70(1), pp. 2–20.

Méndez-Lucio, O. *et al.* (2018) ‘De novo generation of hit-like molecules from gene expression signatures using artificial intelligence’. ChemRxiv.

Metropolis, N. *et al.* (1953) ‘Equation of State Calculations by Fast Computing Machines’, *The Journal of Chemical Physics*, 21(6), pp. 1087–1092. doi: 10.1063/1.1699114.

Michielan, L., Terfloth, L., *et al.* (2009) ‘Comparison of multilabel and single-label classification applied to the prediction of the isoform specificity of cytochrome p450 substrates’, *Journal of chemical information and modeling*. ACS Publications, 49(11), pp. 2588–2605.

Michielan, L., Stephanie, F., *et al.* (2009) ‘Exploring Potency and Selectivity Receptor

Antagonist Profiles Using a Multilabel Classification Approach: The Human Adenosine Receptors as a Key Study', *Journal of Chemical Information and Modeling*. American Chemical Society, 49(12), pp. 2820–2836. doi: 10.1021/ci900311j.

Miller, T. M. *et al.* (1994) 'Organic reaction database translation from REACCS to ORAC', *Journal of Chemical Information and Computer Sciences*, 34(3), pp. 653–660. doi: 10.1021/ci00019a027.

Mishima, K., Kaneko, H. and Funatsu, K. (2014) 'Development of a new de novo design algorithm for exploring chemical space', *Molecular Informatics*, 33(11–12), pp. 779–789. doi: 10.1002/minf.201400056.

Mitchell, J. B. O. *et al.* (1999) 'BLEEP—potential of mean force describing protein–ligand interactions: I. Generating potential', *Journal of Computational Chemistry*. Wiley Online Library, 20(11), pp. 1165–1176.

Mitchell, T. M. (1980) *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ...

Miyao, T., Arakawa, M. and Funatsu, K. (2010) 'Exhaustive structure generation for inverse-QSPR/QSAR', *Molecular informatics*. Wiley Online Library, 29(1-2), pp. 111–125.

Miyao, T., Kaneko, H. and Funatsu, K. (2016) 'Inverse QSPR/QSAR analysis for chemical structure generation (from y to x)', *Journal of chemical information and modeling*. ACS Publications, 56(2), pp. 286–299.

Montanari, F. *et al.* (2016) 'Selectivity profiling of BCRP versus P-gp inhibition: from automated collection of polypharmacology data to multi-label learning', *Journal of cheminformatics*. Nature Publishing Group, 8(1), p. 7.

Moock, T. E. *et al.* (1988) 'Similarity searching in the organic reaction domain', *Tetrahedron Computer Methodology*. Elsevier, 1(2), pp. 117–128.

Morgan, H. L. (1965) 'The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service.', *Journal of Chemical Documentation*, 5(2), pp. 107–113. doi: 10.1021/c160017a018.

Moriaud, F. *et al.* (2009) 'Computational Fragment-Based Approach at PDB Scale by Protein Local Similarity', *Journal of Chemical Information and Modeling*, 49(2), pp. 280–294. doi: 10.1021/ci8003094.

Mullard, A. (2017) 'The drug-maker's guide to the galaxy', *Nature News*, 549(7673), p. 445.

Murray-Rust, P. and Rzepa, H. S. (1999) 'Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles', *Journal of Chemical Information and Computer Sciences*. American Chemical Society, 39(6), pp. 928–942. doi: 10.1021/ci990052b.

Murray, C. W. *et al.* (1997) 'PRO_SELECT: combining structure-based drug design and combinatorial chemistry for rapid lead discovery. 1. Technology', *Journal of computer-aided*

molecular design. Springer, 11(2), pp. 193–207.

Nam, J. and Kim, J. (2016) ‘Linking the neural machine translation and the prediction of organic chemistry reactions’, *arXiv preprint arXiv:1612.09529*.

NextMove Software (2014) *Unleashing over a million reactions into the wild*. Available at: <https://nextmovesoftware.com/blog/2014/02/27/unleashing-over-a-million-reactions-into-the-wild/> (Accessed: 18 April 2017).

NextMove Software (2017) *NameRxn*. Available at: <https://www.nextmovesoftware.com/namernxn.html>.

Nicolaou, C. A. and Brown, N. (2013) ‘Multi-objective optimization methods in drug design’, *Drug Discovery Today: Technologies*. doi: 10.1016/j.ddtec.2013.02.001.

Nikitin, S. *et al.* (2005) ‘A very large diversity space of synthetically accessible compounds for use with drug design programs’, *Journal of computer-aided molecular design*. Springer, 19(1), pp. 47–63.

Nilakantan, R. *et al.* (1987) ‘Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors’, *Journal of Chemical Information and Modeling*, 27(2), pp. 82–85. doi: 10.1021/ci00054a008.

Nishibata, Y. and Itai, A. (1991) ‘Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation.’, *Tetrahedron*, 47(43), pp. 8985–8990. doi: 10.1016/S0040-4020(01)86503-0.

Nisius, B. and Rester, U. (2009) ‘Fragment Shuffling: An Automated Workflow for Three-Dimensional Fragment-Based Ligand Design’, *Journal of Chemical Information and Modeling*, 49(5), pp. 1211–1222. doi: 10.1021/ci8004572.

Norinder, U. and Boyer, S. (2016) ‘Conformal Prediction Classification of a Large Data Set of Environmental Chemicals from ToxCast and Tox21 Estrogen Receptor Assays’, *Chemical Research in Toxicology*, 29(6), pp. 1003–1010. doi: 10.1021/acs.chemrestox.6b00037.

Norris, J. R. (James R. . (1998) *Markov chains*. Cambridge University Press. Available at: https://books.google.co.uk/books/about/Markov_Chains.html?id=qM65VRmOJZAC (Accessed: 10 May 2017).

Nouretdinov, I. *et al.* (2001) ‘Pattern Recognition and Density Estimation under the General i.i.d. Assumption’, in: Springer, Berlin, Heidelberg, pp. 337–353. doi: 10.1007/3-540-44581-1_22.

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), p. 22. doi: 10.1186/1758-2946-4-22.

Olivecrona, M. *et al.* (2017) ‘Molecular de-novo design through deep reinforcement learning’, *Journal of Cheminformatics*, 9(1). doi: 10.1186/s13321-017-0235-x.

Olson, M. and Wyner, A. J. (2018) ‘Making Sense of Random Forest Probabilities: a Kernel Perspective’. Available at: <http://www-stat.wharton.upenn.edu/~maolson/docs/olson.pdf> (Accessed: 5 June 2018).

- openmolecules.org (1999) *WebReactions*. Available at: <http://webreactions.net/>.
- Organic Syntheses Inc. (1921) *Organic Syntheses*. Available at: <http://www.orgsyn.org/>.
- Otero, F. E. B., Freitas, A. A. and Johnson, C. G. (2010) 'A hierarchical multi-label classification ant colony algorithm for protein function prediction', *Memetic Computing*. Springer, 2(3), pp. 165–181.
- Owen, J. R. *et al.* (2011) 'Visualization of molecular fingerprints', *Journal of chemical information and modeling*. ACS Publications, 51(7), pp. 1552–1563.
- Pajouhesh, H. and Lenz, G. R. (2005) 'Medicinal chemical properties of successful central nervous system drugs.', *NeuroRx: the journal of the American Society for Experimental NeuroTherapeutics*. Am. Soc. for Experimental NeuroTherapeutics, 2(4), pp. 541–53. doi: 10.1602/neurorx.2.4.541.
- Papadopoulos, H. and Haralambous, H. (2010) 'Neural Networks Regression Inductive Conformal Predictor and Its Application to Total Electron Content Prediction', in: Springer, Berlin, Heidelberg, pp. 32–41. doi: 10.1007/978-3-642-15819-3_4.
- Pardridge, W. M. (2002) 'CNS Drug Design Based on Principles of Blood-Brain Barrier Transport', *Journal of Neurochemistry*, 70(5), pp. 1781–1792. doi: 10.1046/j.1471-4159.1998.70051781.x.
- Patel, H. *et al.* (2009) 'Knowledge-based approach to de Novo design using reaction vectors', *Journal of Chemical Information and Modeling*, 49(5), pp. 1163–1184. doi: 10.1021/ci800413m.
- Patel, H. (2009) *Knowledge-Based De Novo Design using Reaction Vectors*. University of Sheffield.
- Pearlman, D. A. and Murcko, M. A. (1993) 'CONCEPTS: New dynamic algorithm for de novo drug suggestion', *Journal of Computational Chemistry*. John Wiley & Sons, Inc., 14(10), pp. 1184–1193. doi: 10.1002/jcc.540141008.
- Pearlman, D. A. and Murcko, M. A. (1996) 'CONCERTS: Dynamic Connection of Fragments as an Approach to de Novo Ligand Design', *Journal of Medicinal Chemistry*, 39(8), pp. 1651–1663. doi: 10.1021/jm950792l.
- Pedregosa, F. *et al.* (2011) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, 12(Oct), pp. 2825–2830. Available at: <http://www.jmlr.org/papers/v12/pedregosa11a.html> (Accessed: 13 November 2017).
- Peto, C. J., Jablons, D. and Lemjabbar-Alaoui, H. (2016) 'Anti-cancer Compounds'. Worldwide.
- Pierce, A. C., Rao, G. and 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), pp. 2768–2775. doi: 10.1021/jm030543u.
- Polishchuk, P. G., Madzhidov, T. I. and Varnek, A. (2013) 'Estimation of the size of drug-

like chemical space based on GDB-17 data', *Journal of Computer-Aided Molecular Design*, 27(8), pp. 675–679. doi: 10.1007/s10822-013-9672-4.

Polykovskiy, D. *et al.* (2018) 'Entangled conditional adversarial autoencoder for de novo drug discovery', *Molecular pharmaceuticals*. ACS Publications, 15(10), pp. 4398–4405.

Poongavanam, V., Haider, N. and Ecker, G. F. (2012) 'Fingerprint-based in silico models for the prediction of P-glycoprotein substrates and inhibitors', *Bioorganic & Medicinal Chemistry*, 20(18), pp. 5388–5395. doi: 10.1016/j.bmc.2012.03.045.

Popova, M., Isayev, O. and Tropsha, A. (2018) 'Deep reinforcement learning for de novo drug design', *Science Advances*, 4(7). doi: 10.1126/sciadv.aap7885.

Powers, D. M. W. (2011) 'Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation', *Journal of Machine Learning Technologies*, 2(1), pp. 37–63. doi: 10.1.1.214.9232.

Proschak, E. *et al.* (2009) 'From molecular shape to potent bioactive agents II: fragment-based de novo design', *ChemMedChem: Chemistry Enabling Drug Discovery*. Wiley Online Library, 4(1), pp. 45–48.

Putin, E., Asadulaev, A., Vanhaelen, Q., *et al.* (2018) 'Adversarial threshold neural computer for molecular de novo design', *Molecular pharmaceuticals*. ACS Publications, 15(10), pp. 4386–4397.

Putin, E., Asadulaev, A., Ivanenkov, Y., *et al.* (2018) 'Reinforced adversarial neural computer for de novo molecular design', *Journal of chemical information and modeling*. ACS Publications, 58(6), pp. 1194–1204.

Quadrelli, L., Bareggi, V. and Spiga, S. (1978) 'A New Linear Representation of Chemical Structures', *Journal of Chemical Information and Computer Sciences*, 18(1), pp. 37–40. doi: 10.1021/ci60013a009.

Quinlan, J. R. (1983) 'Learning Efficient Classification Procedures and Their Application to Chess End Games', in *Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 463–482. doi: 10.1007/978-3-662-12405-5_15.

Quinlan, J. R. (1993) 'C4.5: Programming for machine learning', *Morgan Kaufmann*, 38, p. 48.

Rahman, S. A. *et al.* (2016) 'Reaction Decoder Tool (RDT): extracting features from chemical reactions.', *Bioinformatics (Oxford, England)*, 32(13), pp. 2065–6. doi: 10.1093/bioinformatics/btw096.

Ram, S. and Pal, S. (2012) 'An Efficient Algorithm for Automating Classification of Chemical Reactions into Classes in Ugi's Reaction Scheme', *International Journal of Chemoinformatics and Chemical Engineering (IJCCE)*. IGI Global, 2(2), pp. 1–14.

Ramström, O. and Lehn, J.-M. (2002) 'Drug discovery by dynamic combinatorial libraries', *Nature Reviews Drug Discovery*. Nature Publishing Group, 1(1), p. 26.

Rankovic, Z. (2017) 'CNS Physicochemical Property Space Shaped by a Diverse Set of Molecules with Experimentally Determined Exposure in the Mouse Brain', *Journal of Medicinal*

Chemistry. American Chemical Society, 60(14), pp. 5943–5954. doi: 10.1021/acs.jmedchem.6b01469.

Rasmussen, C. E. and Williams, C. K. I. (2005) *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ray, L. C. and Kirsch, R. A. (1957) ‘Finding Chemical Records by Digital Computers’, *Science*, 126(3278), pp. 814 LP – 819. Available at: <http://science.sciencemag.org/content/126/3278/814.abstract>.

Read, J. *et al.* (2009) ‘Classifier Chains for Multi-label Classification’, in: Springer, Berlin, Heidelberg, pp. 254–269. doi: 10.1007/978-3-642-04174-7_17.

Read, J. (2010) *Scalable Multi-label Classification*. University of Waikato. Available at: <https://researchcommons.waikato.ac.nz/handle/10289/4645> (Accessed: 28 November 2018).

Read, J., Pfahringer, B. and Holmes, G. (2008) ‘Multi-label Classification Using Ensembles of Pruned Sets’, in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, pp. 995–1000. doi: 10.1109/ICDM.2008.74.

Reutlinger, M. *et al.* (2014) ‘Multi-objective molecular de novo design by adaptive fragment prioritization’, *Angewandte Chemie International Edition*. Wiley Online Library, 53(16), pp. 4244–4248.

Ridder, L. and Wagener, M. (2008) ‘SyGMa: combining expert knowledge and empirical scoring in the prediction of metabolites’, *ChemMedChem: Chemistry Enabling Drug Discovery*. Wiley Online Library, 3(5), pp. 821–832.

Rogers, D. and Hahn, M. (2010) ‘Extended-Connectivity Fingerprints’, *Journal of Chemical Information and Modeling*. American Chemical Society, 50(5), pp. 742–754. doi: 10.1021/ci100050t.

Rokach, L. (2010) ‘Ensemble-based classifiers’, *Artificial Intelligence Review*. Springer Netherlands, 33(1–2), pp. 1–39. doi: 10.1007/s10462-009-9124-7.

Rokach, L., Schclar, A. and Itach, E. (2014) ‘Ensemble methods for multi-label classification’, *Expert Systems with Applications*. Pergamon, 41(16), pp. 7507–7523. doi: 10.1016/J.ESWA.2014.06.015.

Rose, J. R. and Gasteiger, J. (1994) ‘HORACE: An automatic system for the hierarchical classification of chemical reactions’, *Journal of Chemical Information and Computer Sciences*, 34(1), pp. 74–90. doi: 10.1021/ci00017a010.

Rotstein, S. H. and Murcko, M. A. (1993a) ‘GenStar: A method for de novo drug design’, *Journal of Computer-Aided Molecular Design*, 7(1), pp. 23–43. doi: 10.1007/BF00141573.

Rotstein, S. H. and Murcko, M. A. (1993b) ‘GroupBuild: a fragment-based method for de novo drug design’, *Journal of Medicinal Chemistry*, 36(12), pp. 1700–1710. doi: 10.1021/jm00064a003.

Rottenberg, S. *et al.* (2008) ‘High sensitivity of BRCA1-deficient mammary tumors to the

PARP inhibitor AZD2281 alone and in combination with platinum drugs', *Proceedings of the National Academy of Sciences*, 105(44), pp. 17079–17084. doi: 10.1073/pnas.0806092105.

Roughley, S. D. and Jordan, A. M. (2011) 'The Medicinal Chemist's Toolbox: An Analysis of Reactions Used in the Pursuit of Drug Candidates', *Journal of Medicinal Chemistry*, 54(10), pp. 3451–3479. doi: 10.1021/jm200187y.

Roy, K. (2017) 'Advances in QSAR Modeling', in *Applications in Pharmaceutical, Chemical, Food, Agricultural and Environmental Sciences*. Springer, p. 555.

Royal Society of Chemistry (2008) *ChemSpider*. Available at: <http://www.chemspider.com/>.

Royal Society of Chemistry (2017) *RXNO: reaction ontologies*. Available at: <https://github.com/rsc-ontologies/rxno/>.

Russell, S. J. and Norvig, P. (2016) *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Salmina, E., Haider, N. and Tetko, I. (2015) 'Extended Functional Groups (EFG): An Efficient Set for Chemical Characterization and Structure-Activity Relationship Studies of Chemical Compounds', *Molecules*. Multidisciplinary Digital Publishing Institute, 21(1), p. 1. doi: 10.3390/molecules21010001.

Sanchez-Lengeling, B. *et al.* (2017) 'Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC)'. ChemRxiv.

Satoh, H. *et al.* (2000) 'Classification and prediction of reagents' roles by FRAU system with self-organizing neural network model', *Bulletin of the Chemical Society of Japan*. The Chemical Society of Japan, 73(9), pp. 1955–1965.

Satoh, M. S. and Lindahl, T. (1992) 'Role of poly(ADP-ribose) formation in DNA repair', *Nature*. Nature Publishing Group, 356(6367), pp. 356–358. doi: 10.1038/356356a0.

Sattarov, B. *et al.* (2019) 'De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping', *Journal of chemical information and modeling*. ACS Publications, 59(3), pp. 1182–1196.

Schapire, R. E. and Singer, Y. (1999) 'Improved Boosting Algorithms Using Confidence-rated Predictions', *Machine Learning*. Kluwer Academic Publishers, 37(3), pp. 297–336. doi: 10.1023/A:1007614523901.

Schapire, R. E. and Singer, Y. (2000) 'BoosTexter: A Boosting-based System for Text Categorization', *Machine Learning*. Kluwer Academic Publishers, 39(2/3), pp. 135–168. doi: 10.1023/A:1007649029923.

Schneider, G. *et al.* (2000) 'De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks.', *Journal of computer-aided molecular design*, 14(5), pp. 487–94. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10896320> (Accessed: 11 May 2017).

Schneider, G. (2002) 'Trends in Virtual Combinatorial Library Design', *Current Medicinal*

Chemistry, 9(23), pp. 2095–2101. doi: 10.2174/0929867023368755.

Schneider, G. (2014) ‘Future de novo drug design’, *Molecular informatics*. Wiley Online Library, 33(6-7), pp. 397–402.

Schneider, G. and Baringhaus, K.-H. (2007) *Molecular design: concepts and applications for beginners*. Wiley-VCH.

Schneider, G. and Baringhaus, K.-H. (2013) ‘De Novo Design: From Models to Molecules’, in *De novo Molecular Design*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 1–55. doi: 10.1002/9783527677016.ch1.

Schneider, G. and Fechner, U. (2005) ‘Computer-based de novo design of drug-like molecules’, *Nat Rev Drug Discov*, 4(8), pp. 649–663. doi: 10.1038/nrd1799.

Schneider, N. *et al.* (2015) ‘Development of a Novel Fingerprint for Chemical Reactions and Its Application to Large-Scale Reaction Classification and Similarity’, *Journal of Chemical Information and Modeling*, 55(1), pp. 39–53. doi: 10.1021/ci5006614.

Schneider, N. *et al.* (2016) ‘Big Data from Pharmaceutical Patents: A Computational Analysis of Medicinal Chemists Bread and Butter’, *Journal of Medicinal Chemistry*, 59(9), pp. 4385–4402. doi: 10.1021/acs.jmedchem.6b00153.

Schneider, N., Stiefl, N. and Landrum, G. A. (2016) ‘What’s What: The (Nearly) Definitive Guide to Reaction Role Assignment’, *Journal of Chemical Information and Modeling*. American Chemical Society, 56(12), pp. 2336–2346. doi: 10.1021/acs.jcim.6b00564.

Schwaller, P., Gaudin, T., *et al.* (2018) ‘“Found in Translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models’, *Chemical science*. Royal Society of Chemistry, 9(28), pp. 6091–6098.

Schwaller, P., Laino, T., *et al.* (2018) ‘Molecular Transformer-A Model for Uncertainty-Calibrated Chemical Reaction Prediction’, *arXiv preprint arXiv:1811.02633*.

Schwaller, P. *et al.* (2019) ‘Data-Driven Chemical Reaction Classification with Attention-Based Neural Networks’. ChemRxiv.

Scikit-learn (2007) *Classifier comparison*. Available at: http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html.

Scott, L. J. (2017) ‘Niraparib: First Global Approval’, *Drugs*, 77(9), pp. 1029–1034. doi: 10.1007/s40265-017-0752-y.

Segler, M. H. S. *et al.* (2017) ‘Generating focused molecule libraries for drug discovery with recurrent neural networks’, *ACS central science*. ACS Publications, 4(1), pp. 120–131.

Sello, G. (1998) ‘Reaction classification by similarity: the influence of steric congestion’, *Tetrahedron*, 54(21), pp. 5731–5744. doi: 10.1016/S0040-4020(98)00261-0.

Sello, G. and Termini, M. (1997) ‘Classification of organic reactions using similarity’, *Tetrahedron*, 53(41), pp. 14085–14106. doi: 10.1016/S0040-4020(97)00911-3.

Shafer, G. and Vovk, V. (2007) 'A Tutorial on Conformal Prediction'. Available at: <http://glennshafer.com> (Accessed: 5 June 2018).

Shannon, C. E. (1948) 'A mathematical theory of communication', *Bell system technical journal*. Wiley Online Library, 27(3), pp. 379–423.

Shen, Y., Aoyagi-Scharber, M. and Wang, B. (2015) 'Trapping Poly(ADP-Ribose) Polymerase', *Journal of Pharmacology and Experimental Therapeutics*, 353(3), pp. 446 LP – 457. doi: 10.1124/jpet.114.222448.

Simonovsky, M. and Komodakis, N. (2018) 'Graphvae: Towards generation of small graphs using variational autoencoders', in *International Conference on Artificial Neural Networks*. Springer, pp. 412–422.

Spyromitros, E., Tsoumakas, G. and Vlahavas, I. (2008) 'An empirical study of lazy multilabel classification algorithms', in *Hellenic conference on artificial intelligence*. Springer, pp. 401–406.

Srinivas Reddy, A., Chen, L. and Zhang, S. (2013) 'Structure-Based De Novo Drug Design', in *De novo Molecular Design*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 97–124. doi: 10.1002/9783527677016.ch4.

Stahl, M. *et al.* (2002) 'A validation study on the practical use of automated de novo design', *Journal of Computer-Aided Molecular Design*. Kluwer Academic Publishers, 16(7), pp. 459–478. doi: 10.1023/A:1021242018286.

Ståhl, N. *et al.* (2019) 'Deep reinforcement learning for multiparameter optimization in de novo drug design', *Journal of Chemical Information and Modeling*. ACS Publications.

Stecher, P. G. (1960) 'The Merck index of chemicals and drugs: an encyclopedia for chemists, pharmacists, physicians, and members of allied professions.', *The Merck index of chemicals and drugs: an encyclopedia for chemists, pharmacists, physicians, and members of allied professions*. Rahway, N. J.: Merck & Co., Inc.

Swain, M. (2017) *Molecule Validation and Standardization*. Available at: <https://pypi.org/project/MolVS/>.

Synarchive S.E.N.C. (2011) *SynArchive*. Available at: <http://www.synarchive.com/>.

Szymański, P. and Kajdanowicz, T. (2017) 'A scikit-based Python environment for performing multi-label classification'. Available at: <http://arxiv.org/abs/1702.01460> (Accessed: 25 November 2018).

Takahashi, Y. (1998) 'Structural similarity analysis based on topological fragment spectra', *Advances in Molecular Similarity*. JAI Press, 2, pp. 93–104.

Takeda, S., Kaneko, H. and Funatsu, K. (2016) 'Chemical-space-based de novo design method to generate drug-like molecules', *Journal of chemical information and modeling*. ACS Publications, 56(10), pp. 1885–1893.

Thabtah, F. A., Cowling, P. and Yonghong Peng (2004) 'MMAC: A New Multi-Class, Multi-Label Associative Classification Approach', in *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, pp. 217–224. doi: 10.1109/ICDM.2004.10117.

Todorov, N. P. and Dean, P. M. (1997) 'Evaluation of a method for controlling molecular scaffold diversity in de novo ligand design', *Journal of Computer-Aided Molecular Design*, 11(2), pp. 175–192. doi: 10.1023/A:1008042711516.

Tratch, S. S. and Zefirov, N. S. (1998) 'A hierarchical classification scheme for chemical reactions', *Journal of chemical information and computer sciences*. ACS Publications, 38(3), pp. 349–366.

Tschinke, V. and Cohen, N. C. (1993) 'The NEWLEAD program: a new method for the design of candidate structures from pharmacophoric hypotheses.', *Journal of medicinal chemistry*, 36(24), pp. 3863–70. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/8254618> (Accessed: 11 May 2017).

Tsoumakas, G. and Katakis, I. (2007) 'Multi-Label Classification', *International Journal of Data Warehousing and Mining*. IGI Global, 3(3), pp. 1–13. doi: 10.4018/jdwm.2007070101.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2011) 'Random k-Labelsets for Multilabel Classification', *IEEE Transactions on Knowledge and Data Engineering*, 23(7), pp. 1079–1089. doi: 10.1109/TKDE.2010.164.

USPTO (1994) *United States Patent and Trademark Office*. Available at: <https://www.uspto.gov/patent>.

Varnek, A. *et al.* (2005) 'Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures', *Journal of computer-aided molecular design*. Springer, 19(9–10), pp. 693–703.

Veber, D. F. *et al.* (2002) 'Molecular properties that influence the oral bioavailability of drug candidates.', *Journal of medicinal chemistry*, 45(12), pp. 2615–23. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12036371> (Accessed: 17 January 2019).

Velošo, A. *et al.* (2007) 'Multi-label lazy associative classification', in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, pp. 605–612.

Vinkers, H. M. *et al.* (2003) 'SYNOPSIS: SYNthesize and OPTimize System in Silico', *Journal of Medicinal Chemistry*, 46(13), pp. 2765–2773. doi: 10.1021/jm030809x.

Vléduts, G. E. (1963) 'Concerning one system of classification and codification of organic reactions', *Information Storage and Retrieval*, 1(2–3), pp. 117–146. doi: 10.1016/0020-0271(63)90013-5.

Vléduts, G. E. (1977) *Development of a Combined WLN/CTR Multilevel Approach to the Algorithmic Analyses of Chemical Reactions in View of Their Automatic Indexing. Report No. 5399*.

Vovk, V., Gammerman, A. (Alexander) and Shafer, G. (2005) *Algorithmic learning in a random world*. Springer. Available at: <https://dl.acm.org/citation.cfm?id=1062391> (Accessed: 5 June 2018).

Wager, Travis T *et al.* (2010) 'Defining desirable central nervous system drug space

through the alignment of molecular properties, in vitro ADME, and safety attributes.’, *ACS chemical neuroscience*. American Chemical Society, 1(6), pp. 420–34. doi: 10.1021/cn100007x.

Wager, Travis T. *et al.* (2010) ‘Moving beyond rules: The development of a central nervous system multiparameter optimization (CNS MPO) approach to enable alignment of druglike properties’, *ACS Chemical Neuroscience*. American Chemical Society, 1(6), pp. 435–449. doi: 10.1021/cn100008c.

Wager, T. T. *et al.* (2016) ‘Central Nervous System Multiparameter Optimization Desirability: Application in Drug Discovery’, *ACS Chemical Neuroscience*. American Chemical Society, 7(6), pp. 767–775. doi: 10.1021/acscemneuro.6b00029.

Wallace, J. (2016) *Structure generation and de novo design using reaction networks*. University of Sheffield.

Wang, R., Gao, Y. and Lai, L. (2000) ‘LigBuilder: a multi-purpose program for structure-based drug design’, *Molecular modeling annual*. Springer, 6(7–8), pp. 498–516.

Wang, R., Lai, L. and Wang, S. (2002) ‘Further development and validation of empirical scoring functions for structure-based binding affinity prediction’, *Journal of computer-aided molecular design*. Springer, 16(1), pp. 11–26.

Wang, Y. *et al.* (2010) ‘Structure-based design, synthesis and biological evaluation of new N-carboxyphenylpyrrole derivatives as HIV fusion inhibitors targeting gp41’, *Bioorganic & medicinal chemistry letters*. Elsevier, 20(1), pp. 189–192.

Warr, W. A. (2011) ‘Representation of chemical structures’, *WIREs Comput Mol Sci*, 1. doi: 10.1002/wcms.36.

Warr, W. A. (2014) ‘A Short Review of Chemical Reaction Database Systems, Computer-Aided Synthesis Design, Reaction Prediction and Synthetic Feasibility’, *Molecular Informatics*. John Wiley & Sons, Ltd, 33(6–7), pp. 469–476. doi: 10.1002/minf.201400052.

Wassermann, A. M. *et al.* (2015) ‘The opportunities of mining historical and collective data in drug discovery’, *Drug Discovery Today*. Elsevier Current Trends, 20(4), pp. 422–434. doi: 10.1016/J.DRUDIS.2014.11.004.

Waszkowycz, B. *et al.* (1994) ‘PRO_LIGAND: An Approach to de Novo Molecular Design. 2. Design of Novel Molecules from Molecular Field Analysis (MFA) Models and Pharmacophores’, *Journal of Medicinal Chemistry*, 37(23), pp. 3994–4002. doi: 10.1021/jm00049a019.

Weininger, D. (1988) ‘SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules’, *Journal of Chemical Information and Computer Sciences*, 28(1), pp. 31–36. doi: 10.1021/ci00057a005.

Weininger, D., Weininger, A. and Weininger, J. L. (1989) ‘SMILES. 2. Algorithm for Generation of Unique SMILES Notation’, *Journal of Chemical Information and Computer Sciences*, 29(2), pp. 97–101. doi: 10.1021/ci00062a008.

Wieczorkowska, A., Synak, P. and Raś, Z. W. (2006) ‘Multi-label classification of emotions in music’, in *Intelligent Information Processing and Web Mining*. Springer, pp. 307–315.

Wilcox, C. S. and Levinson, R. A. (1986) ‘A Self-Organized Knowledge Base for Recall,

Design, and Discovery in Organic Chemistry', in *Artificial Intelligence Applications in Chemistry*. American Chemical Society (ACS Symposium Series), pp. 18–209. doi: doi:10.1021/bk-1986-0306.ch018.

Willett, P. (1980) 'The evaluation of an automatically indexed, machine-readable chemical reactions file', *Journal of Chemical Information and Computer Sciences*. ACS Publications, 20(2), pp. 93–96.

Willett, P. (1986) *Modern approaches to chemical reaction searching: proceedings of a conference*. Gower Publishing Company.

Willett, P. (2011) 'Chemoinformatics: a history', *Wiley Interdisciplinary Reviews: Computational Molecular Science*. Wiley Online Library, 1(1), pp. 46–56.

Willett, P. (2013) 'Similarity-Based Scaffold Hopping Using 2D Fingerprints', *Scaffold Hopping in Medicinal Chemistry*. Wiley Online Library, pp. 105–118.

Willett, P., Barnard, J. M. and Downs, G. M. (1998) 'Chemical Similarity Searching', *Journal of Chemical Information and Computer Sciences*, 38(6), pp. 983–996. doi: 10.1021/ci9800211.

Wiswesser, W. J. (1952) 'The Wiswesser Line Formula Notation', *Chemical & Engineering News Archive*, 30(34), pp. 3523–3526. doi: 10.1021/cen-v030n034.p3523.

Witten, I. H. (Ian H. . *et al.* (2016) *Data Mining: Practical Machine Learning Tools and Techniques*, *Data Mining: Practical Machine Learning Tools and Techniques*. doi: 10.1016/c2009-0-19715-5.

Wong, W. W. L. and Burkowski, F. J. (2009) 'A constructive approach for discovering new drug leads: Using a kernel methodology for the inverse-QSAR problem', *Journal of cheminformatics*. BioMed Central, 1(1), p. 4.

Xue, D. *et al.* (2019) 'Advances and challenges in deep generative models for de novo molecule generation', *Wiley Interdisciplinary Reviews: Computational Molecular Science*. Wiley Online Library, 9(3), p. e1395.

Yang, H. *et al.* (2019) 'admetSAR 2.0: web-service for prediction and optimization of chemical ADMET properties', *Bioinformatics*. Edited by J. Wren. Narnia, 35(6), pp. 1067–1069. doi: 10.1093/bioinformatics/bty707.

Yang, X.-S. (2014) *Nature-inspired optimization algorithms*. Elsevier.

Yu, G. *et al.* (2012) 'Transductive multi-label ensemble classification for protein function prediction', in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1077–1085.

Yu, Y. *et al.* (2019) 'Population Pharmacokinetics of Talazoparib in Patients With Advanced Cancer', *The Journal of Clinical Pharmacology*, p. jcph.1520. doi: 10.1002/jcph.1520.

Yuan, W. *et al.* (2017) 'Chemical space mimicry for drug discovery', *Journal of chemical information and modeling*. ACS Publications, 57(4), pp. 875–882.

Yuan, Y., Pei, J. and Lai, L. (2011) ‘LigBuilder 2: a practical de novo drug design approach’, *Journal of chemical information and modeling*. ACS Publications, 51(5), pp. 1083–1091.

Zaliani, A. *et al.* (2009) ‘Second-generation de novo design: a view from a medicinal chemist perspective’, *Journal of Computer-Aided Molecular Design*. Springer Netherlands, 23(8), pp. 593–602. doi: 10.1007/s10822-009-9291-2.

Zefirov, N. S. (1987) ‘An Approach to Systematization and Design of Organic Reactions’, *Accounts of Chemical Research*. American Chemical Society, 20(7), pp. 237–243. doi: 10.1021/ar00139a001.

Zefirov, N. S., Baskin, I. I. and Palyulin, V. A. (1994) ‘SYMBEQ Program and Its Application in Computer-Assisted Reaction Design’, *Journal of Chemical Information and Modeling*. American Chemical Society, 34(4), pp. 994–999. doi: 10.1021/ci00020a038.

Zefirov, N. S. and Tratch, S. S. (1980) ‘Systematization of tautomeric processes and formal-logical approach to the search for new topological and reaction types of tautomerism’, *Chemica Scripta*. CAMBRIDGE UNIV PRESS 40 WEST 20TH STREET, NEW YORK, NY 10011-4211, 15(1), pp. 4–12.

Zhang, M.-L. and Zhou, Z.-H. (2005) ‘A k-nearest neighbor based algorithm for multi-label classification’, in *2005 IEEE International Conference on Granular Computing*. IEEE, pp. 718–721 Vol. 2. doi: 10.1109/GRC.2005.1547385.

Zhang, M.-L. and Zhou, Z.-H. (2007) ‘Multi-label learning by instance differentiation’, in *AAAI*, pp. 669–674.

Zhang, Q.-Y. and Aires-de-Sousa, J. (2005) ‘Structure-based classification of chemical reactions without assignment of reaction centers’, *Journal of chemical information and modeling*. ACS Publications, 45(6), pp. 1775–1783.

Zhang, S., Golbraikh, A. and Tropsha, A. (2006) ‘Development of Quantitative Structure–Binding Affinity Relationship Models Based on Novel Geometrical Chemical Descriptors of the Protein–Ligand Interfaces’, *Journal of Medicinal Chemistry*, 49(9), pp. 2713–2724. doi: 10.1021/jm050260x.

Zhang, T. *et al.* (2012) ‘Classification Models for Predicting Cytochrome P450 Enzyme-Substrate Selectivity’, *Molecular Informatics*. John Wiley & Sons, Ltd, 31(1), pp. 53–62. doi: 10.1002/minf.201100052.

Zhang, W. *et al.* (2015) ‘Predicting drug side effects by multi-label learning and ensemble learning’, *BMC Bioinformatics*. BioMed Central, 16(1), p. 365. doi: 10.1186/s12859-015-0774-y.

Zhu, J. *et al.* (2001) ‘Structure-based ligand design for flexible proteins: application of new F-DycoBlock.’, *Journal of computer-aided molecular design*, 15(11), pp. 979–96. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11989626> (Accessed: 14 May 2017).

Zhu, L., Yang, J. and Shen, H.-B. (2009) ‘Multi label learning for prediction of human protein subcellular localizations’, *The protein journal*. Springer, 28(9–10), p. 384.