Conformation-Independent Comparison of Protein Structures

Robert Adam Nicholls

PhD Thesis

University of York

Chemistry

September 2011

Abstract

The comparative analysis of protein structures is often performed in order to identify and explore similarities/dissimilarities present between target structures. Whilst many tools are available for structural comparison, the development of new tools providing different information is desirable.

The work presented here concerns the development of *ProSMART* (Procrustes Structural Matching Alignment and Restraints Tool), a tool to aid the comparative analysis of protein structures. Primarily, the software is used for conformation-independent pairwise structural alignment, allowing identification of local similarities, and quantification of dissimilarities. Functionality also allows the identification and superposition of rigid substructures, providing output that allows visualisation of local dissimilarities by means of residue-based scoring. The *ProSMART* pairwise alignment method optimises the net agreement of local structures along the chain, using structural fragments. In order to maintain conformation-independence, the structure-based residue alignment does not enforce global rigidity of chains, nor domains. This feature makes the tool suited to the analysis of domain movement and other conformational changes, as well as for the identification of structural units conserved between seemingly different structures.

Given an alignment, *ProSMART* can be used to generate external restraints on the distances between relatively close atoms, for use in the crystallographic refinement of macromolecules. Using one or more similar structures, the software generates restraints that are intended to help the target protein adopt a conformation that is more reasonable, whilst allowing global flexibility. Such restraints may stabilise refinement in some cases, especially at low resolution where experimental data alone may not be sufficient.

We also present a method of Procrustes score normalisation, which allows the consideration of the significance of observed fragment scores. It is suggested that the resulting global scores for the overall pairwise agreement of protein structures may provide an interesting new way of viewing protein fold space.

Contents

List of Figures

List of Tables

Acknowledgements

I would firstly like to thank my supervisor, Garib Murshudov, for all of his advice and guidance over the past few years. I am grateful for the opportunity to work on this project, which has allowed me to learn so much in a variety of fields. Further to proposing the project, and suggesting the underlying methods implemented, he has provided fantastic advice at every stage. Importantly, he has provided an environment where I have been allowed to make my own decisions and explore my own ideas, even when they were not good ones, encouraging my development towards being an independent researcher.

I would like to thank the people who have allowed me to visit, and those who have aided the development of *ProSMART*. Specifically, in order of date, Pavol Skubák, for advice with programming and allowing me to visit and present my work at the University of Leiden; Wladek Minor, for allowing me to visit, present my work, and undertake a project learning about and practising macromolecular structure solution by crystallography at the University of Virginia; Keith Wilson and *CCP*4, for allowing me to attend and present at the *CCP*4 developers' meeting; Fei Long, for working on the *PREFMAC* pipeline; Charles Ballard for working towards the integration of *ProSMART* into the *CCP*4 suite; Martyn Winn, for modifying the *CCP*4*i* graphical user interface in order to allow the use of restraints from *ProSMART* with *REFMAC5*; and Stuart McNicholas, for working on integrating *ProSMART* in *CCP*4*mg*.

I would also like to thank all of the many people that have helped me and provided a friendly and enjoyable working environment during the project, most notably Marcus Fischer, for the many interesting and productive discussions around structural comparison that ultimately led to the implemented methods presented here; Javier Garcia-Nafria and Jens Landström for consultation and permission to use their structures to produce figures in this thesis; also Yuan He, Kamran Haider, Sophie McKenna, Roel Bolsius, Andrey Lebedev, and all at York Structural Biology Laboratory; all those I met whilst at the University of Leiden and University of Virgina; and all of the users of *ProSMART* for their support and feedback, notably Peter Horanyi at the Wiener Lab.

I would finally like to say a special thank you to my parents and to Lucy for their encouragement, support and endurance, who have read my work on many occasions and put up with me throughout.

Author's Declaration

I would like to bring attention to the fact that the use of *ProSMART* for generation of external structure restraints has been mentioned in a published work (Murshudov et al., 2011).

Acknowledging the above exception, I declare that this thesis is my own work and that, to the best of my knowledge, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher education, except where appropriate acknowledgment has been made in the text.

"Procrustes owned two beds, one small, one large; he made short victims lie in the large bed, and the tall victims in the short one...

...we humans, facing limits of knowledge, and things we do not observe, the unseen and the unknown, resolve the tension by squeezing life and the world into crisp commoditized ideas, reductive categories, specific vocabularies, and prepackaged narratives, which, on the occasion, has explosive consequences. Further, we seem unaware of this backward fitting...

It is a very recent disease to mistake the unobserved for the nonexistent; but some are plagued with the worse disease of mistaking the unobserved for the unobservable...

Because our minds need to reduce information, we are more likely to try to squeeze a phenomenon into the Procrustean bed of a crisp and known category (amputating the unknown), rather than suspend categorization, and make it tangible. Thanks to our detections of false patterns, along with real ones, what is random will appear less random and more certain – our overactive brains are more likely to impose the wrong, simplistic, narrative than no narrative at all."

– *Nassim Nicholas Taleb*

The Bed of Procrustes: Philosophical and Practical Aphorisms (2010)

Chapter 1

Comparison of Protein Structures

1.1 Introduction

1.1.1 Background

Protein structures fold in such a way that optimises the energetic favourability of chemical interactions, allowing for conformational flexibility due to thermodynamics. The backbone composition of all proteins is deterministic (i.e. existence and connectivity of N, C^{α}, C and O atoms, ignoring particular conformation) apart from in length (number of residues). However, it is the qualitative nature of residues' side chains (as specified by the amino acid sequence) that cause differences in the global conformation of different proteins, given particular environmental conditions. The hydrophobic and polar properties of side chains, and their ability to form bonds (e.g. hydrogen bonds), is determined by their chemical composition and configuration. Positional properties of atoms/residues relative to the rest of the structure, such as solvent exposure/accessibility, are important in determining the most energetically favourable local, and in turn global, conformation. In particular, hydrophobic residues will generally energetically favour internal regions of the structure, so as to reduce solvent exposure. The size and flexibility of side chains determine the potential range of local backbone conformations. For example, glycine (which has no side chain past the C^{α}) allows a wide range of potential backbone conformations, whilst other amino acid types have a tendency to be much more limited; consider the potential range of ϕ, ψ angles, as depicted in Ramachandran plots (Lovell et al., 2003; Ramachandran et al., 1963). Side chains' size and flexibility determine the ability to accommodate other spatially related regions of the protein, affecting the energetic favourability of global conformations. Furthermore, side chain conformation and flexibility influences nearby atomic interactions (e.g. the ability to form hydrogen bonds, as a donor or acceptor, or form disulphide bridges in the case of cystine). For example, hydrogen bonds are important for the formation/stabilisation of secondary structure, such as those within α-helices and those between strands in β-sheets. The particular state of a protein (i.e. folded or denatured), and the degree of stability (flexibility and disorder), depends further on environmental

factors such as the composition of the surrounding solvent, and temperature; solvent conditions determine the hydrophobic effect and protein-solvent interactions, and temperature determines the degree of flexibility (and disorder) given local stability.

1.1.2 Motivation

Structural comparison/analysis takes many forms, including (but not limited to): alignment, superposition, alignment scoring, structure dissimilarity scoring, and classification. At a base level, it is of interest to do structural comparison in order to:

- Identify potential functional relationships, since structure implies function;
- Identify possible evolutionary links;
- Identify particular residues/regions of interest, e.g. those that may be important for local or global fold stability, or specific regions of importance to biological function (e.g. active sites);
- Extract structural information for use in other applications;
- Investigate the nature of protein folding, at some level.

Macromolecules observed to be structurally similar may be homologous, and any differences between them due to evolutionary divergence; information pertaining to similarity may be of importance in the identification of evolutionary links. In contrast, non-homologous yet structurally similar structures may be considered evolutionarily convergent, and thus may provide insight into the energetic favourability of conformations, at a global or local level. Importantly, the identification of structures exhibiting some similarity allows 'links' to be made between folds (e.g. in the context of a hierarchical classification scheme, such links would be between different structural classes), which is most often done discretely, although in reality such relationships would be continuous. Such linked structures might even have some functional similarity in some cases, making structural comparison biologically relevant for investigations regarding protein function.

Indeed, structural classification is an important problem, which typically involves the clustering of all (known) protein chains according to their dissimilarity. Such schemes may be hierarchical or continuous, manually curated or based on an underlying dissimilarity measure. Other approaches classify domains rather than chains – which is an important distinction. Note that there is ambiguity regarding the actual definition of a 'domain', resulting in potential for different methods to identify domains differently. Consequently, the clustering of domains requires domain identification, which, due to the convolution of multiple techniques, leads to even more ambiguity. As already mentioned, one common application of such a classification facility is the easy identification of links between structures. This may involve the identification of structures sharing a common fold, or those related via subtle, less obvious links. In a slightly more abstract context, such schemes aim to provide a description of fold space. However, such descriptions are not unique, and may provide different

information, meaning that the ability to gain new ways of exploring fold space is a highly desirable future prospect.

1.1.3 Preliminary Acknowledgement of the Problem

In amino acid sequence comparison, the alignment of two sequences is attempted in order to identify correspondences between proteins. The alignment is subsequently quantified using some score of similarity between supposed corresponding residues. In particular, sequence identity may be then used to infer information regarding potential evolutionary relationships. When compared with structural comparison, one advantage of sequence comparison is that it does not require knowledge of protein structures (a 3D model), merely knowledge of the sequence. Furthermore, since the problem is 1D, sequence comparison methods tend to be very fast relative to structural methods. However, the amount of information that can be extracted from 3D structure is vast, often justifying the extra incurred computational expense for many applications.

In structural comparison, rather than considering proteins as a mass comprising thousands of atoms, consideration of the amino acid chain is often used as a way of simplifying the problem into a workable form. This natural representation directly follows from sequence comparison, allowing the use of some similar approaches (e.g. dynamic programming), and allowing analogies to be drawn in some cases. For example, Friedberg et al. (2007) construct a 20 letter 'structural sequence', for analogy with the amino acid sequence. Such an approach is reasonable, since both the in-sequence connectivity of the backbone (peptide bonds), and intra-residue atomic bonds, are deterministically invariant in that they remain bonded throughout proteins' natural dynamical cycles (ignoring alterations, e.g. due to biological function, evolutionary mutations, radiation damage, etc.). Importantly, this connectivity is maintained whether denatured or in the native state.

Reducing the problem to the consideration of a 1D chain in 3D space allows us to intuitively think of protein 3D structure in terms of a flexible backbone, rather than working with a potentially more complicated representation, e.g. a dynamic electron density map. Further to the conceptual simplification, important features of this representation are that it provides sequential order and directionality to the formulation of the problem. In contrast with sequence comparison (in which only amino acid type is used), a wealth of utilisable information is available for each residue in the chain. These properties include:

- Amino acid type;
- Atom types;
- Conformation, in the form of atomic coordinates (and often also a measure of atomic uncertainty, available in the form of B-factors);
- Intra-residue atomic connectivity;

- Inter-residue spatial relationships between atoms (which may be used to predict inter-residue atomic connectivity);
- Relative positions and conformations of atoms from other present molecules (e.g. water, ligands).

Of course, the usefulness and limitations of structural comparison are dependent on the quality of compared models. Whilst we assume a reasonable degree of experimental reliability and accuracy, the potential for model errors should not be overlooked. Indeed, some deposited models have been found to be incorrect (Chang, 2007; Bujnicki et al., 2002), and even those that are considered correct cannot be considered perfect, as suggested by the improvements observed from the re-refinement of deposited models (Joosten et al., 2009a,b). Furthermore, available data are not always complete; often atoms/residues are missing, either due to disorder or poor quality data.

Potential artefacts of the experimental method used to determine the structural model should also be considered, since they may affect both the sensible interpretation of results and methodological development. For example, in X-ray crystallography (which comprises the large majority of structures deposited in the PDB; 62,201 out of 71,516 as of 01/03/2011) proteins are subjected to certain conditions so that they form a crystal, in which the protein structures are approximately homogeneous in conformation (between unit cells), allowing some relatively small amount of atomic positional differences between structures in the crystal, which are often represented using B-factors. One factor that should be considered is that the conditions (solvent composition, temperature) that the protein is subjected to may be different to those in its natural biological environment. Also, crystal contacts between proteins that occur during the process of crystallisation, or change over time in the crystal, may affect surface interactions and possibly cause the overall structure to alter its overall shape in order to comply with crystal packing. These effects may cause the systematic bias of global and/or local conformation.

Another issue is that atomic coordinate data achieved from crystallography are static, whilst proteins are actually dynamic in nature. This is reflected by positional uncertainty (parameterised as B-factors) and, in the case of more extreme flexibility, missing atoms (or alternatively atoms may be assigned zero occupancies). However, even B-factors alone, whether anisotropic or not, are not sufficient to completely describe conformational flexibility. Due to chemical restraints, local atomic positions are correlated, meaning that spatially related atoms cannot be treated independently. However, since atomic bonds are not constraints, local structure cannot be considered a rigid body. In contrast, nuclear magnetic resonance (NMR) spectroscopy allows the reporting of ensembles of states, allowing a description of the protein's range of potential conformations, given environmental conditions. Molecular dynamics (MD) simulations report similar data, although these approaches are theoretical, dependent on prior knowledge in the form of parameters, and as such results should be interpreted suitably.

Given the form of available data, along with knowledge of limitations and associated issues re-

garding reliability, it is our task to design new ways of comparing these protein structures, without loss of generality. Due to the practically infinite variety of potential structures, and the heterogeneity of data quality, this is not a trivial task. Ultimately, the aim is to discern information regarding the structural similarity of proteins that may aid analysis either in specific or unknown applications.

1.1.4 Structural Similarity

Different types of structural comparison may be performed, depending on the degree of similarity, and specific objectives. Ideally, when performing structural comparison, careful thought should be given to the consideration of the particular pair (or set) of structures being compared. However, sensible automation and minimisation of required manual intervention may be preferable, where appropriate. Different approaches and considerations should be adopted depending on the intrinsic level of observed homology by considering both sequence and structural relatedness.

Identical Structures

Even proteins/chains that are identical (or near-identical) in sequence exhibit structural dissimilarities. Observed differences may be due to either systematic effects of environmental conditions, or local/global uncertainties/bias in atomic positions. Identical structures can exhibit different conformations as a result of biologically relevant processes such as binding (and also mutations, for near-identical structures). For example, a structure might exhibit different global conformations when in apo or bound state, or when bound to different ligands, perhaps in the form of a hinging movement. Also, the dynamic nature and conformational flexibility of the structure will cause uncertainty in atomic positions, and in more extreme cases, disorder. This is an important point – available data is in the form of static coordinates, yet in reality protein structures are dynamic. Furthermore, such differences may be a consequence of experimental procedure. When comparing identical or near-identical proteins, it is relevant to consider global backbone conformational change, and also use high-resolution structural features (e.g. side chains) to identify subtle local dissimilarities/conservation that may be important to structural stability or biological function.

Similar Structures

At this point, we must raise the question: what is structural similarity? Protein structures may exhibit similarity in a variety of ways, at a variety of different levels. Consequently, when attempting to identify similarity, it is important to clearly acknowledge the desired type of similarity to be identified, and how such similarity should be qualified and/or quantified. In general, no individual technique can identify all types of similarity, and so, ideally, the use of many different methods should be considered if attempting to identify *some* similarity.

For example, at an abstract level, all proteins might be said to be similar, since they are all

constructed from a polypeptide chain. Also, all structures containing α-helices might be said to be similar, since they all contain α-helices. Of course, such definitions of similarity are contextually meaningless. At a more practical level, structures might be said to be similar if they:

- Have the same, or exhibit some degree of similarity in, overall fold topology;
- Exhibit a similar arrangement of SSEs in the core, regardless of overall topology;
- Conserve any rigid substructure, regardless of its position or composition;
- Both contain some particular motif;
- Exhibit some significant local structural conservation, regardless of overall fold.

These are just a few examples; there are many other types of similarity that could be considered, and many ways in which similarity may be defined. In conclusion, we argue that the definition of similarity depends philosophically on individual beliefs, on the particular structures being compared, and on the purpose of a particular analysis.

Dissimilar Structures

Given that structures exhibit different levels of (a given type of) similarity, a natural thing to do is to attempt to classify/categorise them. More generally, we may wish to explore protein conformation space. In order to do so, it is necessary to determine distances between structures in this space. This raises the question as to how structural dissimilarity should be defined, qualified, and quantified, to which there is arguably no answer. Addressing this issue involves scoring the similarity/dissimilarity of structures in order to discern some way of ordering structure-pairs according to their relatedness in a sensible and meaningful way. There is no unique way of scoring dissimilarity, since there are many ways in which structures could be similar. Ideally, any employed (dis)similarity score should be general enough to make sense for seemingly random (unrelated) structures. Also, the comparison of completely unrelated dissimilar structures is useful for determining the potential range of scores (quantitative) or results (qualitative) that might occur by random for a given measure of similarity.

1.2 Structural Alignment

Various software implementations are referred to in this section. However, for conciseness, only software names are provided (for appropriate citations, see Appendix).

1.2.1 Introduction

The alignment of protein chains (or, in some cases, domains) generally involves the explicit identification of a residue-residue correspondence between two (or more) chains, thus identifying a bijective

correspondence between some subsets of the proteins' residues. Early focus was on sequence alignment, whereby the only input data was the amino acid type (although other information was often utilised, such as presumed dissimilarities between amino acid types, and predicted secondary structure type). However, it has been acknowledged that local sequence similarity does not necessarily imply structural similarity (Saqi et al., 1998). Structure-based methods tend to be much slower due to the inherent increase in computational expense, but are considered to produce more useful (or complementary) information, since it is structure that ultimately determines biological function. Furthermore, structural methods utilise more information, as reflected by the available parameters (i.e. three continuous parameters per utilised atom in structural alignment, versus one discrete parameter per residue in sequence alignment). Note that intermediates exist, utilising both sequence and structural information (see §1.2.3).

Due to huge improvements in both speed and accessibility of computational equipment, and the ever-increasing number of structures (thus information) in the PDB available for exploitation, the last couple of decades has seen a focus on structural alignment and comparison (as demonstrated by the wealth of software developed in recent years, summarised in the Appendix). The amount of structural information that may be exploited in individual structures and between structure-pairs is very large (for various purposes; alignment, identification of similarity, or otherwise). Consequently, the fields of structural alignment and comparison are far from exhausted, and are very relevant at present, as indicated by the accelerated number of structural alignment tools that have materialised in recent years. In complement, the ever-increasing amount of available processing power and computing capabilities will likely continue to make structure-based techniques all the more viable and useful in future years.

Types of Similarity

One important way of categorising methods is global/local. Global methods require compared structures to possess a reasonable amount of global structural rigidity in order to be aligned in a sensible way. A globally rigid central region shared between similar structures is often referred to as the 'core'. In contrast, local methods are based on techniques that are more invariant to global conformation, by requiring only that local structure be conserved in order to identify candidate residues for alignment, thus allowing more flexibility.

Some methods use local similarity criteria to obtain an initial alignment (e.g. *SARF*, *Dali*, *CE*, *SHEBA*), yet subsequently optimise and score the final alignment based on global criteria (most commonly RMSD after superposition). This results in final alignments that require global structural rigidity, and so the resultant alignments/scores are not invariant to global conformation. Note that the method adopted by the developed software *ProSMART* is truly local, whereby both alignment and scoring consistently consider only conservation of local structure, at least for the primary functionality of structural alignment.

Some alignment methods enforce the maintenance of sequence order, whilst some do not. This is an important distinction between methods. If we know that a particular method requires the maintenance of sequence order, then we can be certain that any similarities identified are topologically conserved; this constraint provides extra information regarding the nature of the results. Conversely, if a method does not require the resultant alignment to conserve sequence order, then non-topologically conserved structural similarities can be identified. This ability can provide useful information, since some topologically different structures can exhibit some global structural similarity (Holm and Sander, 1993).

Comparison of Methods

Different alignment methods provide different results, are based on different criteria, and aim to identify different sorts of similarity. Barthel et al. (2007) describe the structural comparison problem as a 'chimera', whereby there is not one single best method or solution, stating that their philosophical approach "*...does not call for the abolition of one method in favour of another one but rather for the intelligent integration of every possible protein structural comparison method...*". This way of thinking effectively validates the development of new alignment and comparison methods, which provide new and complementary information to those methods already in existence.

It could be reasoned that methods based on different criteria cannot be directly compared. Methods are often compared with existing classification databases, e.g. *SCOP* (Murzin et al., 1995; Andreeva et al., 2008) and *CATH* (Cuff et al., 2011; Orengo et al., 1997), and reliability/accuracy are determined by agreement with these 'gold standard' databases. In fact, some methods refine parameters in order to optimally agree with such databases. Of course, if the database is based on an existing method (e.g. *CATH* classifies using information from *SSAP*; *FSSP* (Holm et al., 1992) uses *Dali*) then such a comparison would effectively presume the particular method used in creation of the database to be the 'gold standard' alignment/scoring method. One might argue that it is of much greater interest to identify which methods do not provide the same results, thus providing different and complementary information. However, if two methods provide exactly the same results (or have exactly the same objectives) then it may be more reasonable to compare them in some way. Of course, for applications where speed is critical, knowledge of which methods are computationally faster may be useful.

1.2.2 Selection of Structural Features

The general procedure of alignment/comparison begins by selecting a type of feature that can be compared, such as intra- or inter-molecular interatomic distances (Subbiah et al. (1993), *Dali*, *Protein3Dfit*, *STRUCTAL*, *KENOBI*, *PRIDE*, *MALECON*, *MUSTANG*, *MatAlign*, *SSGS*, *Fr-TM-align*), atomic vectors or environments (*SSAP*, *MAMMOTH*, *FAST*, *YAKUSA*), structural fragments (*SARF*, *CE*, *FlexProt*, *FATCAT*, *MultiProt*, *POSA*, *SP*^{3}, *MUSTANG*, *Matchprot*, *Matt*,

RAPIDO, *Fr-TM-align*, *TOPS++FATCAT*), torsion angles (*SARST*, *TALI*, *SABIC*), backbone differential geometry (*CTSS*, *CURVE*, *CAALIGN*, *ComSubstruct*, *STON*), SSEs (*SARF2*, *VAST*, *DEJAVU*, *PrISM*, *KENOBI*, *K2*, *K2SA*, *MASS*, *SSM*, *FASE*, *TOPS+strings*, *TableauSearch*), substructure volumes in the form of tetrahedrons (*TOPOFIT*, *TetraDA*), and spatial contacts (*Vorolign*, *SABERTOOTH*, *ProBiS*). When atomic coordinates are to be used to represent residues, usually either the positions of the four main chain atoms (N, C^{α}, C, O) or more commonly just the C^{α} atoms are used, although C^{β} (*SSAP*, *STRUCTAL*, *Matras*, *Vorolign*) and other side chain (*SPASM*) atoms have also been utilised. When describing SSEs, different implementations have used varying levels of detail; the most common approach uses vectors (position and orientation), whilst others also utilise other information such as SSE length (*DEJAVU*, *TOPS+strings*) or SSE atomic coordinate matrices (*KENOBI*).

The initial raw data used for structural comparison generally comprises atomic coordinates, together with knowledge of their in-sequence connectivity (i.e. the chemical bonds that are maintained whether or not the protein is denatured). Most structural alignment methods utilise only this knowledge, which pertains to the relative location of features. However, some methods utilise chemical information, which cannot be achieved from structure (atomic coordinates). Such methods use a variety of different sorts of features, such as hydrogen bonding, amino acid type, SSE type, functional grouping, solvent exposure, polarity and aromaticity (*SSAPe*, *SHEBA*, *Matras*, *MolCom*, *CTSS*, *SP*^{3}, *Vorolign*, *ProBiS*). There are benefits in the utilisation of atoms' intrinsic properties (such as polarity, hydrophobicity, etc.), since it is the net effect of these properties, along with restraints in the form of chemical bonds and interactions, that result in proteins' observed conformations. However, there is ambiguity in combining structural and non-structural information in a sensible way.

Structural Resolution

The types of features chosen for comparison determines the structural resolution of the comparison method – the consideration of SSEs indicates a medium/low level of structural resolution, whilst the utilisation of main chain atomic coordinates corresponds to a higher level of detail (and side chains provide further detail again). Tentatively, some vague analogies may be formed with multi-resolution descriptors used in other fields (e.g. Fourier descriptors, wavelets, moments, harmonics). There are benefits associated with using varying levels of structural resolution, since doing so allows to obtain a rich breadth of information that may be used to infer the nature of any observed similarities. Different resolutions may be suitable for identifying different levels of similarity. High-resolution structural conservation (e.g. conserved side chain position) would only be expected for very similar structures, and may distinguish between different levels of similarity for very similar structures. In contrast, low-resolution features (e.g. SSEs) would be very insensitive to such subtle dissimilarities due to the inherent smoothing-out of any atomic-level details, and are more suited

to identifying whether similar overall folds are adopted by less-similar structures. Intermediate resolutions, such as the consideration of backbone C^{α} coordinates, lie somewhere in the middle.

Structural Fragments

We define a structural fragment to be a set of (ordered, in-sequence, adjacent) coordinates forming a subset of the coordinates of one protein. For example, a length n-residue fragment might comprise the four main chain atoms corresponding to n adjacent residues (resulting in $4n$ coordinates), or just the C^{α} atoms (resulting in n coordinates). If another fragment of equal length exists in another protein, then the atomic coordinates of the two fragments are directly comparable, and the fragments form an isomorphism between subsets of the two proteins. For example, if a consecutive residue-residue alignment is expressed as a pair of coordinate matrices of equal size, then these two coordinate matrices may be considered to be structural fragments, since they represent a hypothetical one-to-one atomic correspondence. More importantly, the set of all possible length n-residue fragments from any two proteins is a set of directly comparable landmark configuration matrices (for landmarks in shape analysis, see Dryden and Mardia, 1998).

The concept of n-residue fragments, the fact that a unique set of n-residue fragments can be achieved for any protein structure (providing sufficient atoms exist), and the fact that any two n-residue fragments are directly comparable, is fundamental to the developed method implemented in the software tool *ProSMART*. In context, structural fragments are powerful constructs that allow comparisons to be performed at a chosen resolution, or at multiple resolutions. In contrast with most other features, there is potential for structural resolution to be chosen in a relatively smooth fashion, since the fragment length may be selected as desired.

Secondary Structural Elements (SSEs)

Due to the size of the protein data bank, there is a need for comparison tools to be fast, especially for applications involving database scanning. Proteins exhibit very few SSEs relative to the number of residues, of which there are few compared with the number of atoms. In general, lower resolution methods tend to use fewer observations than more detailed methods and consequently tend to be faster (fragment-based methods excepted, which may exhibit opposite phenomena). However, SSEs are constructs that may or may not exist. Therefore, whilst SSE-based approaches are well-suited to the identification of a rigid common core shared between similar structures, they cannot generally be used to detect more distant relationships, nor used to provide an alignment/score for any arbitrary chain-pair. For example, such approaches cannot meaningfully compare an all-α structure with an all-β structure. Furthermore, potential ambiguities in SSE definition/detection may also cause problems. Traditionally, secondary structure class is defined/detected using hydrogen bonding patterns, according to *DSSP* criteria (Kabsch and Sander, 1983). However, structural flexibility and experimental uncertainty, particularly in low-resolution structures, may cause seem-

ingly SSE-like regions to fall outside the limits of detection. Whilst most methods use the *DSSP* definition, this issue has been addressed by considering alternative strategies for defining SSEs. For example, similarity searches might be performed using 'typical' α-helix and β-strand fragments as representatives (*SARF2*).

1.2.3 Structural Alignment Methods

There are many different methods of structural alignment. Whilst structure-based methods are generally much slower than sequence-based methods, some structural methods are much faster than others. For example, those which consider a very low level of structural resolution, most notably those based on SSEs, have the potential to be very much faster than others that consider a higher structural resolution, where data elements might represent residues rather than SSEs.

Whilst most methods consider features of a specific level of structural resolution, such as SSEs or atomic distances, some consider features general enough to allow a choice of resolution, such as structural fragments. Some implementations simultaneously operate at multiple levels of resolution, so that the final result/score is the aggregate of various types of information regarding similarity. These include: *PRIDE*, which considers differences between the distributions of C^{α} atoms separated by a predetermined number of residues; and *MolCom*, which compares the proportion of residues that exhibit certain properties between supposed corresponding cubes in an octree.

Most methods aim to identify a correspondence between a pair of structures, although some methods extend to the alignment of multiple structures (*SSAP*, *SSAPe*, *SAP*, *STAMP*, *MASS*, *POSA*, *MUSTANG*, *Vorolign*, *CAALIGN*, *Matt*). This is often achieved by iteratively combining the results from pairwise comparisons. The primary purpose of some tools (*MALECON*, *MAMMOTH-mult*, *RESOLVER*, *CBA*) is to perform multiple structure alignment, requiring the provision of pairwise alignments performed externally (or internally, using existing methods). In contrast, some methods align all molecules simultaneously (*MultiProt*).

Given a choice of feature(s), the initial step in structural alignment is generally to represent inter-feature similarities/dissimilarities, based on some pre-defined scoring criteria. Such a representation is commonly in the form of a distance matrix, which may comprise distances between positions, but could also describe other distances such as those based on vectors. The aim of the initial alignment stage is either to optimise a path through the matrix, or to identify pairs of elements (low-resolution methods) or regions (high-resolution methods) that might potentially correspond, based on observed similarities.

For optimal path identification using some methods, e.g. dynamic programming, the sequence order must always be preserved. Some methods may choose not to enforce this constraint, allowing the potential for detection of non-topological correspondences/similarities (*SARF*, *SARF2*, *Dali*, *SPASM*, *KENOBI*, *K2*, *K2SA*, *MASS*, *FLASH*, *SSM*, *SCALI*, *TOPOFIT*, *MultiProt*, *GANGSTA*, *FASE*, *LOVOALIGN*, *Matchprot*, *RAPIDO*, *ProBiS*).

Following the identification of an initial alignment, further refinement stages are often used in order to optimise some criteria. Such criteria (often largely dependent on global RMSD) reflect the particular type(s) of similarity to be identified, according to the implementation's purpose, and prior beliefs regarding the definition of similarity.

Superposition

An often-favoured method of superposition is one that finds the exact optimal solution (subject to numerical error) to the co-minimisation of the squared deviations between corresponding co-ordinates (atoms), without the need for iterative optimisation. This method, which is invariant to the original coordinate frame (specifically, to the translation and rotation of coordinates), is often referred to in biology as the Kabsch algorithm (Kabsch, 1978) or the McLachlan algorithm (McLachlan, 1982), and has historically been used in Procrustes analysis (see Gower and Dijksterhuis, 2004). This method of superposition has been used in many alignment implementations, including a dedicated superposition tool (*ProFit*), where the alignment must be specified by the user, and allows multiple-structure superposition by iteratively converging pairwise superpositions.

This method is actually a special case of Procrustes analysis (also referred to as partial Procrustes analysis, although we shall not make this distinction). In general, full Procrustes analysis may be applied to arbitrary-dimensional objects, and allows scale-invariance by normalising with respect to the average distance between the coordinates and the average position. Since chemical restraints dictate the scale of proteins and their interatomic distances, we suppose that allowing scale-invariance would not be sensible in this context.

Some approaches that aim to identify rigid global conservation score the (dis)similarity of structures after superposition using scores other than the RMSD of aligned residues. The superposition achieved from Procrustes analysis minimises the RMSD, and may not give the superposition that optimises other scores. This issue has been addressed (*LOVOALIGN*) by iteratively refining the transformation in order to optimise a similarity score.

There have been other approaches towards superposition, such as using URMSD (see §1.3.2), which gives a translation-independent result (Chew et al., 1999). In another approach, Falicov and Cohen (1996) superpose structures without requiring an explicit residue alignment in the conventional sense (although one is implied), using an iterative approach that attempts to minimise the area between protein backbones. This is achieved by considering the intermolecular triangulation of C^{α} atoms. Areas between such triangles are expressed as elements of a dynamic programming matrix; performing dynamic programming on this matrix results in the optimal triangulation, and thus the area between backbones. A search algorithm iteratively finds the superposition that minimises the area.

Rather than minimising RMSD using least squares, *THESEUS* (Theobald and Wuttke, 2006) uses maximum likelihood to superpose multiple structures, assuming a prior alignment. Rather

than forcing all residuals to assume equal variance, this method allows estimation of the covariance matrix. This effectively allows the weighting of regions based on their agreement in different structures, allowing the contribution of different residues to vary.

In contrast with other methods that directly utilise information regarding atomic positions, another approach is to superpose electron density maps (Vagin and Isupov, 2001). In this case, atoms are inherently weighted according to their electron composition and their positional uncertainty. When using an experimentally obtained map, structural models do not have to be complete in order for features to contribute to the superposition. However, an electron density map can also be obtained directly from a model. Furthermore, the method may be performed at different resolutions by trimming the resolution limit in Fourier space.

Global RMSD Optimisation

One of the most common objectives in structural alignment/comparison is rigid substructure identification. This has been achieved using various methods, such as iteratively refining an alignment using Monte Carlo simulations on intramolecular interatomic distance matrices (*Dali*). However, rigid substructure identification is perhaps most commonly approached by optimising a superposition, using intermolecular RMSD, allowing identification of a common rigid region. One main purpose of this is to identify a common core, if it exists. If two (similar) structures adopt the same overall fold (sometimes ignoring topology), then it is possible that a common core exists, and can be identified and superposed.

The purpose of global RMSD optimisation, in this context, is to iteratively refine an initial alignment so that all aligned features (usually C^{α} atomic coordinates) superpose reasonably well, by achieving some sensible tradeoff between a score, e.g. RMSD, and some parameter(s), e.g. alignment length. For example, *SHEBA* iteratively refines the alignment so that it comprises the maximum number of aligned residues subject to all corresponding atoms having a superposed RMSD less than 3.5Å. Such an approach has been widely adopted, having been applied to initial alignments resulting from dynamic programming (Subbiah et al. (1993), *STRUCTAL*, *PrISM*, *SHEBA*, *MatAlign*, *Fr-TM-align*, *SABIC*); fragment-based methods (*SARF*, *CE*, *FATCAT*, *TOPS++FATCAT*); secondary structure-based methods (*SARF2*, *VAST*, *LOCK*, *KENOBI*, *K2*, *K2SA*, *Matras*, *SSM*, *FASE*); sequence-structure alignment (*STAMP*, *SCALI*); interatomic distance sorting (Petitjean, 1998); and geometric hashing (*CAALIGN*).

The approach of *TOPOFIT* is notable, due to being interestingly different to other methods. The volume of substructures is represented using tetrahedrons defined by Delaunay tessellation of C^{α} atoms. Rigid substructures are identified by combining similarly shaped tetrahedrons, and thus also the implied residues at vertices, into larger substructures subject to maintaining RMSD criteria.

Dynamic Programming

One of the well-known approaches is to use a dynamic programming algorithm to identify the unique optimal correspondence between sequences, subject to scoring criteria. Such techniques have long been used in biology for sequence alignment (Sankoff and Kruskal, 1983), yet have also found application in such interestingly diverse fields as speech recognition (Myers and Rabiner, 1981) and the comparison of musical scores (Mongeau and Sankoff, 1990). Specifically, the method is commonly implemented (Wishart et al., 1994) in the form of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), or alternatively the Smith-Waterman variation (Smith and Waterman, 1981). The resulting alignment is not necessarily one-to-one, allowing any element from one sequence to be mapped to either none, one, or multiple elements from the other sequence, in accordance with time-warping (Myers and Rabiner, 1981). Consequently, further manipulation is needed to achieve a partial bijection, if required, so that each residue is either unaligned, or aligned to only one residue in the other chain. The method is a very powerful path-optimisation technique due to always identifying the optimal solution (although note that the graph-theoretical approach adopted for SSE alignment also identifies the optimal solution). However, one limitation is that it can only align sequences in-order, and consequently cannot detect non-topological relationships. The technique can be generalised to higher dimensions, allowing multiple sequence alignment (see Clote and Backofen, 2000), although this inherently results in increased algorithmic complexity.

Dynamic programming has been used in various structural alignment implementations, with and without using a gap penalty. The method has been used to achieve an initial alignment and for alignment refinement (*STAMP*, Subbiah et al. (1993), *STRUCTAL*, *LOCK*, *CE*, *SHEBA*, *Matras*, *MAMMOTH*, *FATCAT*, *CTSS*, *FAST*, *TetraDA*, *SP*^{3}, *MUSTANG*, *MatAlign*, *CURVE*, *SGSS*, *FASE*, *LOVOALIGN*, *Matchprot*, *SABERTOOTH*, *TALI*, *Matt*, *TOPS+strings*, *TOPS++FATCAT*, *FrTM-align*, *TableauSearch*, *SABIC*). For example, *STAMP* uses dynamic programming in order to refine a superposition, whereby elements of the score matrix are based on positional and orientational agreement *after* superposition, thereby allowing refinement by dynamic programming iterations.

An extension to the method, termed 'double dynamic programming', has also been used, whereby the elements (scores) of the dynamic programming matrix are themselves the net scores resulting from dynamic programming on inter-feature vector matrices (*SSAP*, *SSAPe*, *SAP*, *PrISM*). This approach greatly increases computational cost, increasing the dimensionality from 2 to 4, yet utilises much more available information. *Vorolign* also uses double dynamic programming, although matrices comprising only spatially related residues are used, rather than inter-feature vector matrices. In this case, the residue-based similarity scores are a weighted sum of amino acid and SSE type exchange scores. *MatAlign* also takes a nested approach, using dynamic programming to achieve scores for the agreement of rows from interatomic distance matrices. The alignment is then achieved by performing dynamic programming on the resultant matrix of row-row scores.

Structural Fragment Based Methods

The constructs known as structural fragments have been previously used for various purposes. Whilst most applications, particularly those in the context of alignment, have used relatively short fragment lengths, some have used longer lengths. For example, lengths in the range of 27–180 have been used for investigating the relationship between RMSD and alignment length (Maiorov and Crippen, 1994). Whilst most fragment-based methods consider overlapping fragments, some do not, such as *Fr-TM-align*, which aligns non-overlapping fragments to achieve initial alignment seeds.

One early fragment-based alignment method (*SARF*) uses C^{α} backbone fragments (of length 6 or 7) for the detection of rigid substructures. The method identifies spatially related fragment pairs sufficiently conserved between the two compared proteins, and builds up an alignment by iteratively combining fragments in a favourable way, according to criteria based on RMSD and alignment length. Due to the spatial nature of inter-fragment connectivity, maintenance of sequence order is not required, and so this approach allows the detection of non-topological similarities in the core.

Another method (*CE*) begins by identifying all sufficiently similar fragments (of length 8) between two structures. Starting from an initially aligned fragment-pair, potentially corresponding fragments are iteratively added to the alignment and extended using a combinatorial approach comprising various heuristics. Multiple different starting points are considered, since different alignments may be achieved depending on the choice of the initially fixed fragment pair; the best alignment is selected for subsequent refinement.

Rather than the combinatorial approach, *MUSTANG* uses dynamic programming to identify the optimal path. The structures are searched for all potential fragments of maximal length (at least 6) given a sufficiently small RMSD. Scores for the residue-based dynamic programming matrix are based on intra-fragment C^{α} distances, using an elastic score (similarly to that of *Dali*).

FlexProt begins using a similar approach, which searches for all potential fragments of maximal length (at least 12) given a sufficiently small RMSD. However, a graph-based approach is employed to find an optimal alignment, and gaps are identified as hinges. The alignment does not require global rigidity, although the fragments, which may be sizeable, must be reasonably rigid. Consequently, this globally flexible approach might be considered an intermediate between global and local methods. Rather, it requires/identifies piecewise rigidity along the chain. A similar approach is taken by *FATCAT* (and *TOPS++FATCAT*), which, rather than taking a graph-theoretical approach, uses dynamic programming to identify an optimal path of fragments. This approach allows the chain to twist (rotate) unconstrained about a discrete number of residues, if sufficiently beneficial to the overall RMSD score. *RAPIDO* also takes a similar approach, but uses a graph-theoretical approach so that non-topological similarities can be identified.

The flexible approach of *Matt* considers fragments of length 5–9, and aligns them using dynamic programming. The fragment is iteratively built up over three dynamic programming passes. The

fragment-based score is based on RMSD, displacement and relative angles of consecutive fragments. Only those with sufficiently low RMSD, displacement and relative angles may be aligned.

Another approach, *SGSS*, identifies long structurally stable fragments. The alignment and scoring of such fragments is dependent on predicted SSE type, allowing more flexibility for loop regions, and penalising the alignment of residues with different SSE types.

The tool *SP*^{3} considers a database of fragments (from non-redundant high-resolution structures). For each fragment in a target structure, the 25 best-matching fragments in the library are identified (found using a score that includes information about RMSD and solvent exposure). These fragments are used to estimate the distribution of amino acid types, at each residue position. This information is used to create a residue-based score, which is then used to align structures using dynamic programming.

Note that some methods that do not explicitly mention fragments might also be said to adopt a fragment-based approach, due to the use of a 'sliding window' of consecutive residues to compare local regions. However, rather than considering fragments' atomic coordinates, as per our definition of a structural fragment in §1.2.2, and comparing them using the RMSD, some approaches use other ways of describing dissimilarity of local structure. For example, *Dali* considers intra-fragment interatomic distances, and *ComSubstruct* considers the conformation of runs of tetrahedra. Another example is *MAMMOTH*, which, rather than comparing fragments using coordinate RMSD, compares them using URMSD (see below).

Matchprot allows the consideration of spatial neighbourhoods, which have analogies with structural fragments. However, unlike with structural fragments, ambiguity of ordering and correspondences between spatial neighbourhoods causes problems in pairwise comparison. In this case, correspondences are achieved using a graph-theoretical approach.

Backbone Differential Geometry

Curvature and torsion of the backbone is considered in *CTSS*. This is implemented by smoothing the backbone using approximate splines, and calculating the curvature and torsion of the resultant curve at points closest to residues' C^α atoms. This approach allows the transformation-invariant comparison of local structure in a way that considers backbone conformation, without being concerned with details regarding atomic positions.

A similar approach is taken by *CURVE*, which smoothes the backbone. However, this implementation is only concerned with curvature, not torsion. Specifically, the angles between points, separated by an arbitrary number of residues, along the smoothed backbone are considered. Due to smoothing using an arbitrary number of residues, and considering an arbitrary separation of points, this approach may operate at varying levels of structural resolution.

Another approach is that of *ComSubstruct*, which considers backbone curvature by representing the chain as connected tetrahedra.

Some other methods of exploring local dissimilarities also consider backbone curvature, ignoring torsion. This has been done by considering unit-vectors between consecutive C^{α} atoms, and scoring their conservation using the URMSD (see §1.3.2). This approach is independent of translation, but dependent on rotation, of the original coordinate frames. Consequently, whilst not requiring globally rigid conservation as strict as coordinate RMSD-based approaches, the URMSD is not truly invariant to global conformation.

One method (Chew et al., 1999) uses this approach to identify contiguous in-sequence rigid substructures using an exhaustive combinatorial search, shifting the whole constant-length alignment so as to minimise the URMSD. Upon achieving an alignment, rotation matrices between consecutive unit-vectors are compared using the Frobenius norm; those below some threshold are considered to belong to a rigid substructure. Larger substructures (e.g. domains) are then identified by combining substructures with similar translations and rotations.

MAMMOTH combines this idea with the use of structural fragments, by considering the URMSD of length 7 fragments. Alignment is achieved using dynamic programming on a matrix of URMSD-based similarity scores, before filtering so that the global RMSD is within some threshold.

Rather than considering differential geometry at the inter-residue level, *TALI* considers features at a higher-resolution, specifically backbone torsion angles. Distances between pairs of torsion angles are achieved by integrating over the density of the Ramachandran plot in a straight line between the two (ϕ, ψ) observations, providing the residues are of the same type. Dynamic programming is then used to align the structures using these scores.

SABIC also considers high-resolution features, specifically intra-residue bond lengths, bond angles, and torsion angles. Consequently, the constructed score effectively quantifies differences in backbone differential geometry at the residue level. Sufficiently long runs of sufficiently well-scoring residue-pairs are identified, before subsequent rigid refinement. A similar approach is taken by *CAALIGN*, which considers backbone torsion angles, defined by the conformation of four consecutive residues. Angles are discretised, and consecutive angles are combined into words. Identical words are subsequently used as alignment seeds.

Secondary Structure Based Methods

As previously mentioned, SSE-based methods have the potential to be very fast due to the reduced number of features compared with methods operating at a higher level of structural resolution. Many implementations use SSEs to achieve an initial residue alignment, which is then superposed and further refined. Other methods consider the SSE alignment alone, in order to describe similarities in general folding patterns.

Graph-theoretical representations/approaches have often been used in SSE alignment (*SARF2*, *VAST*, *SSM*, *GANGSTA*, *TOPS+strings*). Maximal clique detection, which identifies the unique

maximal common subgraph isomorphism, requires the prior identification of a suitable feature graph for each structure. Specifically, in this context, the technique requires a presumed set of similar pairs of SSEs. Whilst suitable graphs might be constructed based on residue/atomic connectivity, such an approach is generally not practical due to the high order of complexity of the algorithm (generally NP-complete). Consequently, the approach is limited to low structural resolution features (SSEs), with which it has been deemed to be very fast and successful (at least when there are not too many SSEs). Note that there are exceptions, such as *ProBiS*, which consider a residue-based graph. In this case, the complexity of the graph is reduced by considering only surface residues, whereby nodes are characterised by physicochemical properties.

TOPS+strings represents SSE connectivity as a graph, but does not take the common graph-theoretical approach of attempting to identify a common subgraph isomorphism. Rather, this information is represented as a 1D string, capturing information regarding spatial connectivity from the graph representation. These strings are then aligned using dynamic programming.

Other SSE-based implementations include the use of geometric hashing (*LOCK*, *TableauSearch*, *MASS*), SSE match ranking (*FLASH*, *FASE*), single (*LOCK*, *TOPS+strings*) and double (*PrISM*) dynamic programming on the matrix of pairwise SSE scores. Techniques using iterative optimisation in order to find local minima include branch-and-bound approaches (*Matras*), and stochastic Monte Carlo approaches in the form of genetic algorithms (*KENOBI*, *K2*, *GANGSTA*) and simulated annealing (*K2SA*). Branch-and-bound approaches systematically consider evolving an alignment (branch), whilst reducing the combinatorial problem by removing seemingly unfavourable options (bound). Genetic algorithms inherit operations that occur naturally in evolution, e.g. mutation, along with chosen probabilities of such operations occurring between two predefined alignments. Iterative optimisation (evolution) is performed, using a given list of initial random alignments. Simulated annealing algorithms are (only) subtly different to genetic algorithms, since they stochastically optimise a single alignment, rather than co-optimise multiple alignments.

Intermediates Between Structural and Sequence Alignment

Intermediates between sequence and structural alignment exist, such as 'structure-aided sequence alignment', in which structural information is provided in the form of constraints (*PROMALS3D*). Such constraints might include predicted SSEs, or information from external sources, such as homologous structures.

Some methods employ 'sequence-aided structural alignment', which uses an initial sequence alignment to superpose structures, before the subsequent identification and refinement of a structure-based alignment (*STAMP*). Of course, such methods require the proteins to exhibit sufficient sequence homology. Another approach is to use sequence-structure predictions/probabilities, exploiting the fact that sequence motifs may imply conserved structural motifs (*SCALI*).

Other methods perform 'structural sequence alignment', which involves representing the chain

as a sequence/profile that, rather than comprising amino acid types, describes structural features in some way. For example, Bowie et al. (1991) construct '3D profiles' utilising information regarding residue environments, including solvent accessibility and secondary structure type. More recently, Friedberg et al. (2007) adopted a structural fragment-based approach using a 1D structural sequence of 20 letters, for analogy with the alphabet of amino acid types, and concluded the performance to lie somewhere between that of compared true-sequence and true-structural alignment methods, in terms of both speed and 'quality'. Another approach (*SARST*) clusters torsion angles into 20 groups (using the Ramachandran plot), allowing chains to be represented using this torsion angle sequence. Sequence alignment may then be performed, using a transition score matrix.

Other approaches to structural sequence alignment include *YAKUSA*, which represents the backbone as a sequence of angles, based on the geometry of four consecutive C^{α} atoms. Common substructures are then identified by searching for common subsequences in a database of structures. Another implementation, *TetraDA*, constructs a sequence containing information regarding spatial relationships. These relationships are determined by tetrahedralisation, and aligns the sequences using dynamic programming. *ComSubstruct* takes another approach, encoding the conformation of fragments of tetrahedra so as to re-write the chain as a structural sequence.

Motif Identification

Further to pairwise protein alignment, another related application is motif identification. This is similar to substructure identification, but with the aim of detecting the presence of a particular structural motif, rather than just any common substructure. Such motifs may or may not be connected in sequence, and make use of non-structural information. Particular implementations include the ability to search a protein database for the presence of a particular motif (*SPASM*), and also the opposite counterpart, whereby a particular structure is searched for the presence of motifs from a library (*RIGOR*). Motifs are considered to match if their score, e.g. RMSD, is within some threshold.

1.3 Similarity/Dissimilarity Scoring

1.3.1 Introduction

There are various reasons for scoring an alignment, such as to:

- Identify similarity – which might be any type of similarity, e.g. global rigidity, detection of rigid substructures, local backbone conservation, similarities in overall fold topology, non-topological similarities, conservation of specific regions (e.g. active site), motif recognition;
- Quantify dissimilarity – which could be used to gain insight regarding conformation space. The nature of such a space would be very much dependent on the type of (dis)similarity identified and scored;

- Order structure-pairs by similarity – according to a specific type of similarity, with acknowledgement of the limitations. This may allow the identification of the 'most similar' structures, and enable potential for categorisation;
- Group similar structures – such categorisation, where appropriate, may lead to automated methods of classification;
- Describe the significance of observed similarities – which would be an estimate of significance, highly dependent on prior knowledge;
- Optimise an alignment with respect to a score function – which would mean that the alignment depends on the score, as well as the score depending on the alignment.

In general, given an alignment of features (e.g. residues or SSEs), there are two approaches towards scoring: directly scoring the overall alignment, often done for global methods (e.g. global RMSD); or combining feature-based scores in some sensible way (e.g. sum of scores from the optimal dynamic programming path). The actual method of scoring, and the choice of features used, is highly dependent on the types of similarity desired to be identified, and also on more practical considerations such as computational expense.

Methods that use an RMSD versus N_{align} approach, which are perhaps the most common type of method, effectively identify the number of residues that may be sensibly superposed by a rigid global superposition of the chains, given some criteria, and subsequently report how generally rigid the corresponding superposition is. Whilst this widely adopted approach is very useful and produces intuitive and aesthetic results, it must be noted that this is neither the only way of identifying similarities between structures, nor the only way of scoring the dissimilarity of structures.

Note that, whilst perhaps the most obvious method, RMSD-based scoring is not the only way of scoring a globally rigid alignment. For example, Falicov and Cohen (1996) construct a score, the Area-C^{α} distance, which is the minimum area between the backbones of two superposed structures, normalised with respect to average interatomic distance.

Most methods provide measures representing dissimilarity (e.g. RMSD), possibly along with some parameters (e.g. N_{align}), and often combine them into a similarity score. Many methods attempt to assess the statistical significance of an observed score, often using a z-score. Such statistics are useful for scoring/choosing between alignments. For example, whilst many methods that attempt to search for a globally rigid substructure aim to maximise N_{align} subject to the global alignment RMSD remaining below some threshold, others might choose N_{align} so as to optimise the z-score.

When considering the significance of scores, a favoured approach is to look at the distribution of scores that arise from either an all-on-all or a one-on-all comparison of random structures. To reduce some of the bias in the PDB, often only presumed non-homologous (dissimilar) structures are used in construction of the dataset, so that elements might be considered to represent classes. The

set of 'dissimilar' structures used to estimate the distribution of random scores is often specified by a non-redundant dataset, e.g. *PDBselect* (Griep and Hobohm, 2010). Such a dataset is a subset of the PDB, which might be specified using various criteria, e.g. according to a hierarchical structural classification database, or according to sequence clustering.

The remainder of this section will identify some of the scores that have been previously used to quantify similarities/dissimilarities between protein structures. This includes examples of the scoring of individual features, the scoring of global alignments, and the significance of scores. Note that the nomenclature of equations in this section may not be exactly as appears in the referenced sources. Rather, an attempt has been made to enforce some notational commonality throughout, for clarity.

1.3.2 Feature-Based Scoring

RMSD and Related Dissimilarity Measures

Many methods score the similarity of aligned structures by considering the RMSD of corresponding atoms after superposition:

$$d_{\mathrm{RMSD}}(\mathbf{X},\mathbf{Y}) = \sqrt{\frac{1}{n}\mathrm{tr}\left((\mathbf{X}-\mathbf{Y})^{\mathbf{T}}(\mathbf{X}-\mathbf{Y})\right)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{3}(\mathbf{X}_{ij}-\mathbf{Y}_{ij})^2} \tag{1.1}$$

where the $n \times 3$ matrices $\mathbf{X}$ and $\mathbf{Y}$ represent the pre-superposed corresponding atomic coordinates for the two structures, respectively. This dissimilarity measure scores the overall rigidity of the presumed correspondence in a way that is invariant to the structures' original coordinate frames (i.e. invariant to a rigid body transformation). In order to provide context, it is relevant to determine the significance of an observed score, so that it is possible to infer whether or not structures can be considered to be similar.

However, this score depends on intrinsic properties of the coordinate sets being compared. For example, the potential range of RMSD depends heavily on the overall size of the compared structures, as can be observed by re-writing the RMSD in terms of radius of gyration (Maiorov and Crippen, 1994):

$$d_{\mathrm{RMSD}}(\mathbf{X},\mathbf{Y}) = \sqrt{R_{\mathbf{X}}^2 + R_{\mathbf{Y}}^2 - 2v} \tag{1.2}$$

where $R_{\mathbf{X}} = n^{-1}\mathrm{tr}\left(\mathbf{X}^{\mathbf{T}}\mathbf{X}\right)$ and $R_{\mathbf{Y}} = n^{-1}\mathrm{tr}\left(\mathbf{Y}^{\mathbf{T}}\mathbf{Y}\right)$ are the radii of gyration of the coordinate matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively, and v depends only on the correlation matrix of $\mathbf{X}$ and $\mathbf{Y}$. Therefore, given equal correlation, the RMSD is expected to be increased for larger (sub)structure-pairs.

In order to account for the size-dependence of the RMSD, many methods construct a similarity score as a function of RMSD and the number of aligned residues (N_{align}). The purpose of this is to allow a constant (dis)similarity threshold to be applied in all cases, or, rather, so that an observed (dis)similarity score might imply the same level of significance irrespective of backbone length.

Another related score is the URMSD (unit-vector RMSD) (Chew et al., 1999), which, rather than considering differences between corresponding points in Euclidean space, is the RMSD of

corresponding points on the unit sphere. These points are given by: $\hat{\mathbf{X}}_i = \frac{\mathbf{X}_{i+1.} - \mathbf{X}_{i.}}{|\mathbf{X}_{i+1.} - \mathbf{X}_{i.}|}$, which represents the direction of the C^α of residue $i+1$ relative to the C^α of residue i. These points contain information regarding backbone curvature, ignoring torsion, relative to the original coordinate frame. The URMSD is given by:

$$d_{\mathrm{URMSD}}(\mathbf{X}, \mathbf{Y}) = d_{\mathrm{RMSD}}\left(\hat{\mathbf{X}}, \hat{\mathbf{Y}}\right) \tag{1.3}$$

Superposition of the unit-vector coordinate matrices ($\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$), in the ordinary fashion, allows the URMSD to be minimised with respect to rotation of the original coordinate frames.

Unlike the standard RMSD, this score is independent of translation. However, it is dependent on rotation, in contrast with atomic distance matrices. Whilst not being invariant to global conformation, the translation-independence makes it more flexible than standard coordinate RMSD, allowing favourable scores whenever the aligned backbones are in the same orientation in their coordinate frames. Furthermore, unlike the RMSD, this score has an upper bound, meaning that poor-scoring residue-pairs have limited influence on the overall score. However, due to operating at the residue-pair level, this high-resolution score is sensitive to local changes.

Residue-Based Scores

Rather than using a raw distance/dissimilarity measure, e.g. distance between atoms, many authors have chosen to apply a transformation in order to achieve a similarity score. For example, *STRUCTAL* uses a score:

$$s_{ij} = \frac{\alpha}{1 + \left(\frac{d_{ij}}{\beta}\right)^2} \tag{1.4}$$

where d_{ij} is the distance from an atom i in the target molecule to atom j in the other. This score is used to construct a dynamic programming matrix, which is used for iterative alignment and superposition. A similar functional form is used by *LOCK* to score various distances and angles pertaining to the conservation of SSE-pairs, and also by *Fr-TM-align* to score the similarity of structural fragments.

Similarly, Rossmann and Argos (1976) use a score that is dependent on the alignment's superposition. However, this score, which behaves like a probability, takes a different functional form. Specifically, it is concerned with the globally rigid conservation of not only supposed equivalent residues' positions, but also their backbone orientations. The score for residue i in one structure being aligned to residue j in the other, calculated following the alignment and superposition of the structures, may be expressed:

$$s_{ij} \propto e^{\frac{-d_{ij}^2}{\alpha}} e^{\frac{-v_{ij}^2}{\beta}} \tag{1.5}$$

where d_{ij} is the distance between the aligned residues (C^α coordinates), and v_{ij} represents local backbone orientational conservation by considering positions of adjacent residues. Choice of parameters (α, β) allows weighting of the position and orientation components. This score effectively measures how well-conserved aligned residues are in their superposed coordinate frames, and is

used to iteratively re-weight residues for re-superposition. The approach has been also been used and adapted (*STAMP*) to re-score a dynamic programming cost matrix, allowing the alignment to be iteratively refined along with the superposition.

In contrast, the scoring method of *Dali* (also used by *KENOBI*) is independent of the alignment's superposition, being concerned with the conservation of intra-molecular interatomic distances:

$$s_{ij} = \left(\alpha - 2\frac{|d_{ij}^x - d_{ij}^y|}{d_{ij}^x + d_{ij}^y} \right) e^{-\left(\frac{d_{ij}^x + d_{ij}^y}{2\beta}\right)^2} \tag{1.6}$$

where d_{ij}^k is the distance between aligned atoms i and j in molecule k. This scores how well-conserved distances are relative to how far apart they are, including an exponential term to reduce the effect of atom-pairs very far apart in the molecule.

Fragment-Based Scores

Whilst structural fragments are often scored using standard techniques, e.g. coordinate RMSD after superposition (*SARF*, *CE*, *FlexProt*, *FATCAT*, *MUSTANG*), or based on interatomic distances (*Dali*, *CE*, *MUSTANG*), other scores have been developed. For example, *MAMMOTH* uses the similarity score:

$$s_{ij} = \begin{cases} 10\frac{d^* - d_{\mathrm{URMSD}}(\mathbf{X_i}, \mathbf{Y_j})}{d^*} & d^* > d_{\mathrm{URMSD}}(\mathbf{X_i}, \mathbf{Y_j}) \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

where $\mathbf{X_i}$ and $\mathbf{Y_j}$ are the coordinate matrices corresponding to fragments i and j from the two molecules, respectively, and d^* is the expected value of the URMSD of randomly oriented unit vectors (note: not random protein fragments), given the number of coordinates, which can be determined analytically.

SSE-Based Scores

PrISM uses a score that rewards same-type SSE-pairs for conservation of relative position and orientation:

$$s_{ijmn} = \left(\frac{\alpha}{\beta + |d_{ij}^x - d_{mn}^y|} \right) \left(\frac{d_{ij}^x + d_{mn}^y}{\gamma} \right)^{-\delta} f(|\theta_{ij}^x - \theta_{mn}^y|) \tag{1.8}$$

where d_{ij}^k represents positional difference between SSEs as the average of distances between C^α atoms in SSE i and the closest C^α atom in SSE j in molecule k, θ_{ij}^k is the angle between vectors representing SSEs i and j in molecule k, and $f(\theta)$ is a discrete stepwise function that rewards low, and penalises high, angles. This method of scoring inter-molecular pairs of intra-molecular SSE-pairs is used to construct matrices for double dynamic programming.

Other Feature-Based Scores

Some methods include non-structural information in their scores. For example, the score used by *SHEBA* is the sum of weighted scores of certain properties, allowing a gap penalty, which may be

expressed:

$$s_{ij} = \alpha - \beta\delta_{ij} + \sum_z w_z \log\left(\frac{f_{\text{aligned}}(P^x_{zi}, P^y_{zj})}{f_{\text{random}}(P^x_{zi}, P^y_{zj})}\right) \tag{1.9}$$

where $f_{\text{aligned}}(P_1, P_2)$ (or $f_{\text{random}}(P_1, P_2)$) is the normalised frequency with which the states P_1 and P_2 of some property were found paired in a database of aligned (or random) residue-pairs, P^k_{zi} is the state of property z for residue i in molecule k, w_z is the weight given to the component corresponding to property z, and δ_{ij} is a boolean indicator function specifying alignment gaps. Specifically, the properties z are derived from sequence homology, secondary structure type, solvent accessibility and polarity.

A different approach is adopted by *Matras*, which considers the transition probabilities of certain features. Using Markov transition matrix formalism allows feature transitions to be interpreted in an evolutionary context. Specifically, by considering a database of supposed similar non-identical structures (using *SCOP* classes), scores representing the chance of a given feature evolving from one (discrete) state i to another state j may be calculated *a priori*, allowing an assumed alignment to be scored directly from the predetermined set of transition scores. For each feature, these scores take the form:

$$s_{ij} = \log\left(\frac{P(i \to j)}{P(i)}\right) \approx \log\left(\frac{M^N_{ij}}{f_i}\right) \tag{1.10}$$

where $P(i \to j)$ is the probability of the feature evolving from state i to state j, $P(i)$ is the probability of state i occurring, M^N is the feature's N^{th} order Markov transition matrix, and f_i is the frequency of occurrence of state i. Multiplication of the transition matrices (order N) effectively smoothes the probability distributions, and consequently might be interpreted as allowing hypothetical further evolutionary steps to be taken, allowing for similarities more distant than those present in the original database. *Matras* applies this to various types of features: environment (local backbone conformation categories), distances between C^β atoms (different transition matrix is calculated for each residue interval, up to a separation of 20), and SSE-pairs (numbers of residues in each SSE, distances and angles between SSE-pairs). Ultimately, the final score is only dependent on the C^β distance component.

1.3.3 Global Similarity Scores

Similarity Scores

A common alignment approach is to maximise an alignment subject to the condition that all aligned residues have a RMSD less than an arbitrary threshold, after superposition. The actual threshold varies, e.g. Alexandrov et al. (1992) use 3Å, Alexandrov and Fischer (1996) use 3.2Å, Jung and Lee (2000) use 3.5Å, Ortiz et al. (2002) use 4Å. Using this approach, if two structures are globally rigidly similar, then the number of aligned residues should be large. Whilst some implementations merely quote the RMSD and N_{align}, some aim to use a 1D descriptor in the form of a similarity score. Perhaps the simplest length-normalised similarity score is the ratio of RMSD and N_{align}.

This quantity has been used (*CTSS*), although it is undefined for structurally identical molecules. Subbiah et al. (1993) use the inverted ratio as a dissimilarity score. Various other global similarity scores have been proposed. For example, the form:

$$S = \frac{\alpha N_{\text{align}}}{d_{\text{RMSD}} + \beta} \tag{1.11}$$

is adopted by *SARF2* (and also by *MatAlign*), although note that Alexandrov (1996) and Alexandrov and Fischer (1996) use different values of α, β.

Some scores are normalised with respect to chain lengths, e.g. the similarity and dissimilarity scores proposed by Kleywegt and Jones (1994). Such an approach is also taken by *SSM*, which uses a score based on the square of the RMSD:

$$S = \frac{N_{\text{align}}^2}{\left(1 + \left(\frac{d_{\text{RMSD}}}{\alpha}\right)^2\right) N_x N_y} \tag{1.12}$$

where N_k is the number of residues in molecule k. This score is modified by *SABIC* in order to also utilise information regarding the number of alignment gaps (N_{gap}), and weights the contribution of the RMSD-based component by sequence identity (S_{ID}, which is the proportion of aligned residues that are of the same amino acid type):

$$S_{\text{mod}} = \frac{N_{\text{align}} S}{N_{\text{align}} + N_{\text{gap}}} (1 - S_{\text{ID}}) + \frac{N_{\text{align}}^2}{N_x N_y} S_{\text{ID}} \tag{1.13}$$

where S is given by Equation (1.12).

Another approach (*STAMP*) penalises the number of gaps in the alignment, accounting for the lengths of the two structures, using the global score:

$$S = \frac{\sum_i s_i}{N_{\text{align}}} \left(\frac{N_{\text{align}} - G_x}{N_x}\right) \left(\frac{N_{\text{align}} - G_y}{N_y}\right) \tag{1.14}$$

where N_k and G_k are the numbers of residues and alignment gaps in molecule k, respectively, and the s_i are the scores for aligned residue-pairs (a measure of positional and orientational equivalence, after superposition, as given by Equation (1.5)).

Gaps in the alignment are also penalised in the score of *STRUCTAL*, which arises from Equation (1.4). However, this score is not normalised with respect to alignment length, allowing the score to increase for longer alignments:

$$S = \alpha \left(\sum_i \frac{1}{1 + \left(\frac{d_i}{\beta}\right)^2} - \frac{G}{2} \right) \tag{1.15}$$

where the d_i are the inter-molecular distances between aligned residue-pairs, and G is the number of alignment gaps. A related functional form is also used by *Fr-TM-align*, which includes information regarding chain length, but does not use a gap penalty.

Some SSE-based methods take a very different approach, scoring the final alignment using a variety of information regarding atomic positions and SSE agreement (e.g. *GANGSTA*). To give

an example, the PSD (Protein Structural Distance) score used by *PrISM* includes both C^α RMSD and SSE-based components:

$$S = \left(\frac{-\log\left[\left(\frac{N_1^{\mathrm{SSE}}}{\max(N_1^{\mathrm{SSE}}, N_2^{\mathrm{SSE}})}\right)\left(\frac{s(x,y)}{s(x,x)}\right)\right]}{\log(\alpha)} \right)^2 + \left(\frac{d_{\mathrm{RMSD}}}{\beta}\right)^2 \tag{1.16}$$

where N_k^{SSE} is the number of SSEs in molecule k, and $s(x, y)$ is the sum of scores (given by Equation (1.8)) of aligned SSEs from a dynamic programming matrix. Note that this function is asymmetric with respect to x and y.

z-Scores

A widely adopted approach towards determining the significance of an alignment is to transform the corresponding similarity/dissimilarity score into a z-score, which measures how many standard deviations an observed score is from the mean of a distribution (used in many implementations, including: *SARF2*, *Dali*, *CE*, *SHEBA*, *Matras*, *SSM*, *YAKUSA*, *SP*^{3}, *GANGSTA*, *CURVE*, *SABER-TOOTH*). Such a transformation is achieved by estimating the distribution of scores that would arise from the comparison of random dissimilar structures. The resultant z-score is a similarity measure of how much better an observed score is relative to that expected from presumed random structures. Note that, since score distributions are often non-Normal (as found by Alexandrov and Fischer, 1996), direct interpretation and comparison of z-scores is not always sensible without further knowledge, e.g. the distribution of z-scores that might arise by random.

Given a global score $S(x, y)$ between molecules x and y, the z-score may be calculated:

$$z(x, y) = \frac{S(x, y) - \mu_A(x, y)}{\sigma_A(x, y)} \tag{1.17}$$

where A is the adopted set of dissimilar non-redundant structures. Different ways of estimating the mean (μ_A) and standard deviation (σ_A) have been adopted. One approach is to consider the comparison of only molecule x with all structures in A, so that μ and σ depend only on x (that is, $\mu_A(x, y) = \mu_A(x)$ and $\sigma_A(x, y) = \sigma_A(x)$). However, this results in $z(x, y)$ being asymmetric (Alexandrov and Fischer, 1996, address this issue using a symmetric function of $z(x, y)$ and $z(y, x)$). This approach has been extended by using intrinsic properties of molecule x to describe observed trends. For example, *Matras* uses linear regression on N_{align}^2 to estimate $\mu_A(x)$, potentially allowing improved robustness.

Another approach is to perform an all-on-all comparison of structures in A, and then attempt to describe any observed trends using intrinsic properties of molecules x and y. For example, *Dali* (Holm and Sander, 1998) approximates the mean as a cubic polynomial of the geometric average chain length $N = \sqrt{N_x N_y}$ for small chains ($N \leq 400$) and linear for larger chains ($N > 400$), and standard deviation as a function of the mean ($\sigma(N) = \frac{\mu(N)}{2}$).

STRUCTAL adopts a similar but different approach to *Dali*, relating their score (Equation (1.15)) to the logarithm of the number of aligned residues, resulting in μ_A being approximated by

a polynomial and σ_A a linear relation for $N < 120$, and for larger alignments μ_A linearly related to $\log(N_{\text{align}})$ and σ_A constant. Note that this change in trend somewhat agrees with *Dali*. They approximate the resultant z-score distribution using an extreme-value distribution, thus allowing the assignment of probabilities to observed scores. Note that other implementations (*MAMMOTH*, *GANGSTA*, *SABIC*) have also found their similarity scores to follow extreme-value distributions.

Other Significance-Based Scores

The significance of an observed score is often estimated by considering the distribution of scores observed from the comparison of presumably non-redundant structures. This information might be utilised by inferring the probability of observing a realised score, given some parameters. For example, *VAST* allows the probability to depend on the numbers of SSEs in the compared structures; *SSM* also adopts this approach.

Another approach is to normalise a score, e.g. RMSD, with respect to the distribution of dissimilar structures in a way that is different to the construction of a z-score. For example, some threshold (d_{cutoff}) might be determined, allowing a similarity score to be realised in the form:

$$S = d_{\text{cutoff}} - d_{\text{RMSD}} \tag{1.18}$$

A relation is suggested by Alexandrov and Gō (1994), who consider how the first percentile of the distribution arising from presumed dissimilar structures depends on alignment length (forcing the alignment to be continuous in sequence):

$$d_{\text{cutoff}} = \alpha + (\beta N_{\text{align}} + \gamma)^{\frac{1}{2}} \tag{1.19}$$

This is used to construct a similarity score (for *SARF*), given by the difference between the dissimilarity threshold and observed RMSD:

$$S = 1.37 + (1.16 N_{\text{align}} + 15.1)^{\frac{1}{2}} - d_{\text{RMSD}} \tag{1.20}$$

Whilst most authors determine significance by considering the distribution of scores observed from the comparison of presumably dissimilar structures, other approaches have been suggested. For example, Maiorov and Crippen (1994) suppose that if, upon the comparison of translation-normalised coordinate sets, a lower RMSD is achieved using a roto-reflection (as indicated by negativity of the determinant of the correlation matrix) than using a simple rotation, then the structures might be considered completely dissimilar. This was not contradicted by a comparison of non-redundant structures. Assuming that the alignment is continuous in sequence, results lead to the proposition of the functional form of the RMSD threshold, which represents the first percentile of the distribution of dissimilarity:

$$d_{\text{cutoff}} = \alpha + \beta N_{\text{align}}^{\frac{1}{3}} \tag{1.21}$$

Note that this function is different (but has similarities) to Equation (1.19), demonstrating ambiguities between different studies.

A completely different approach is adopted by *PRIDE*, which does not attempt to align structures, and is only interested in identifying the significance of structural similarity. This method considers the distribution of distances between C^{α} atoms separated by a different numbers of atoms (separations of 3–30 residues by default). Distributions are compared using contingency tables (and later a Kolmogorov-Smirnov test), resulting in the probability of identity of distributions. Due to considering a wide range of residue separations, the method requires similarity at a wide range of structural resolutions.

1.4 Conclusions and Synopsis

There have been many approaches developed for the alignment and comparison of protein structures, providing different, and often complementary, ways of performing comparative structural analyses. These may be roughly classified as: global methods, which require global spatial rigidity; flexible methods, which require piecewise spatial rigidity; and conformation-independent methods, which require only local structural conservation. Even within these categories, there has been a lack of consensus as to an appropriate measure for describing the distance between protein structures, which is of importance for various applications such as structural classification and investigation of protein fold space.

For example, there are many functional forms that have been used to realise scores based on RMSD and alignment length, as seen in §1.3.3. This indicates a lack of consensus regarding an appropriate distance measure even in cases as simple as using scores based on global RMSD. Indeed, some scores seem to have been achieved rather arbitrarily, being dependent on a large number of parameters that are either chosen or empirically determined[1]. In some cases, global scores may be extremely convoluted, being constructed from different types of information, and comprising many parameters. It is argued that such an approach would restrict the ability to achieve a meaningful and concise interpretation of results, beyond inferring that molecule-pair (a, b) scores better/worse than pair (c, d), thus is considered more similar/dissimilar in some way.

It is important to recognise the conceptual distinction between scoring the (dis)similarity of chain-pairs and alignments. Specifically, any score achieved from an alignment corresponds to substructures, not to the complete structures, of the compared chain-pair, unless the alignment identifies a direct isomorphous correspondence between the two chains (this issue is discussed further in Chapter 4). For example, methods whose scores are based on global RMSD often determine the residue alignment by requiring the RMSD to be below a given threshold. Such backward-fitting results in the global RMSD being largely arbitrary, requiring the number (or proportion) of aligned residues to be taken into account in order to achieve a meaningful score. However, any subsequently achieved score would correspond to a substructure of size pre-determined by the arbitrarily RMSD

[1] note that the use of empirically-derived scores can only possibly be reasonable if the data is appropriate/unbiased, the model describes the data well, and the functional form is suitably parsimonious.

threshold, and not to the overall comparison of the structures as a whole[2].

Flexible alignment methods aim to allow the presence of multiple rigid substructures, which is particularly useful in the presence of domain motion. Even in the absence of multiple distinct domains, spatially correlated conformational flexibility (e.g. due to the effect of crystal packing, or ligand binding) can result in more flexible methods providing different information to global methods. Indeed, conformation-independent methods are even more powerful than flexible methods (in this context), being able to account for complex heterogeneous spatial movements, and detect structural conservation in regions that are locally conserved. For example, such an approach would account for surface loops that may be locally highly conserved despite appearing highly flexible relative to the coordinate frame of more sizeable spatially-related rigid substructures (e.g. domains). Consequently, we hypothesise that an approach that is independent of global conformation and spatial correlations would be most suited to scoring chain-pairs' overall structural conservation/dissimilarity as a whole[3].

We surmise that, when searching for a distance between protein structures, it is important that the score makes sense even for completely dissimilar structures. Indeed, it is important to ask how (dis)similar the two structures are, and not ask what similarities can be identified. In particular, the way in which the score is constructed should be the same for similar structures as for dissimilar structures – differences between similar and dissimilar chain-pairs should be quantitative, not qualitative. Therefore, we propose that methods that identify common regions based on thresholds are not suitable for this purpose, since such discontinuity would be reflected in the score, and the score would not make sense for dissimilar chain-pairs. Furthermore, separation into discrete categories should be avoided in order to make comparison tangible, and to avoid excessive reduction of available structural information. For example, we would not choose SSEs as structural features for this application, since such a representation would not allow the sensible comparison of any arbitrary chain-pair, and also such features may not be sufficiently well defined to be detectable in all cases.

Dissimilarity measures have been developed for the comparison of rigid protein substructures, or so-called domains, and such measures have been successfully used for the classification of protein structures and the description of fold space. However, there is currently no such conformation-independent measure with the properties discussed in this section. This observation justifies the development of the alignment method presented as part of this project. It is anticipated that a dissimilarity measure based on net local backbone conformation could be designed in future as a consequence of the presented approach, with the intention of providing information that is different and complementary to existing resources.

[2]Although this does not mean that such scores are not useful for some purposes.

[3]or at least, some maximal correspondence.

Project Synopsis

The objective of the presented alignment method is to identify a residue correspondence between two protein chains that rewards local structural similarity (or penalises local dissimilarity), is insensitive to global conformational changes, and results in a score (or scores) that may lead towards definition of a new and meaningful distance between protein structures in future. Being primarily interested in the conservation of local backbone structure, we intend to achieve a method which is completely independent of spatial relationships. This contrasts with global methods, which require global spatial rigidity, and flexible methods, which require piecewise spatial rigidity. Such an alignment would be intended to provide a representation that is different and complementary to existing approaches, allowing subsequent analysis to be independent of global conformation (conformation-independent to some degree determined by the structural resolution of compared features).

Further to focussing on structural alignment, other applications arising as a consequence of the approach have been explored. Notably, the developed software includes the ability to generate interatomic distance restraints for use in macromolecular crystallographic refinement; the adopted alignment approach is considered to be appropriate for this application, since the generated restraints operate locally, being independent of global conformational differences between the target and reference structures. Also, a new method of rigid substructure identification and superposition is presented, which exploits spatial relationships between aligned regions, allowing a powerful way of visualising spatial similarities between structures. Furthermore, the provision of various residue-based local dissimilarity scores, and the ability to intuitively view results in colour using molecular graphics software, allows a unique and informative way of performing comparative structural analyses. The software also includes various other functionalities, such as the ability to identify, score, and generate restraints for particular structural motifs.

Solutions to the considered problems of structural alignment, rigid substructure identification, and external restraint generation were achieved, and implemented in the form of the software tool *ProSMART*. Whilst the achieved solutions are deemed suitable for the intended purposes, it should be acknowledged that there is potential for improvement or alternative solutions in some areas. For example, a heuristic approach is currently taken towards optimisation/refinement of the final structural alignment; the rigid substructure identification approach is highly dependent on several parameters controlling sensitivity to rigidity; and there is room for improvement in the estimation of restraint sigmas. Such issues are not investigated as part of this project.

Methods employed as part of the adopted approach are detailed in Chapter 2, with examples of application provided in Chapter 3. Properties of the achieved global score (representing net local backbone structural dissimilarity) are considered, and issues are raised regarding the score's dependence on the density of local conformation space. As a consequence, Procrustes score standardisation is investigated in Chapter 4, leading towards a different and complementary way of scoring local structural and global dissimilarities.

Chapter 2

Methods Employed in the Developed Software

ProSMART is a tool for structural analysis in a way that is reasonably independent of global conformation. There are various features, although alignment and generation of restraints are the two major functionalities. These are related, since the restraint generation implementation utilises the achieved alignment. Most other features arose as a consequence of user demand and feedback. The adopted approach focuses only on local structure, at a chosen level of structural resolution. Superposition and substructure identification features are included, but these occur after an alignment has been achieved, and consequently do not influence the alignment towards global rigidity. This chapter describes the main techniques and methods employed in *ProSMART*. Functionality and usage will be subsequently considered in Chapter 3.

One of the objectives of *ProSMART* is to achieve a scoring method that suitably penalises dissimilarities between a chain-pair, whilst being independent of global conformation. One of the key fundamental aspects of the implemented method is that the alignment length is maximised, regardless of the feature-based scores of aligned fragments/residues. Note that such a condition is required, otherwise the optimal alignment would be achieved when the alignment is of length zero. The standard approach to this problem would be to maximise the alignment length subject to a score threshold. For example, globally rigid methods generally tend to maximise the alignment length subject to some function of RMSD being below an arbitrary threshold. Logically, the conformation-independent analogy would be to maximise the alignment length subject to all aligned residues having local RMSD below an arbitrary threshold. *ProSMART* avoids the use of such an arbitrary threshold by maximising the alignment length. Consequently, the resultant alignment length is always either equal to, or slightly less than, the length of the shorter protein chain. This means that alignment length is not indicative of alignment quality. Rather, the average Procrustes score, or more generally the distribution of Procrustes scores, is indicative of alignment quality.

Most existing structural comparison methods (see Chapter 1) are concerned with comparing

chains rigidly (globally rigid methods), comparing rigid substructures within chains (flexible methods), or comparing flexible but similar substructures (conformation-independent methods). In contrast, *ProSMART* is concerned with the comparison of whole chains, identifying the maximal correspondence of residues, subject to local criteria. This amounts to identifying flexible substructures (as with other conformation-invariant methods), but without requiring them to be 'sufficiently' similar, only that they are the most favourable out of all possible alignments.

By maximising the alignment length, and not rejecting poor-scoring fragment-pairs from the alignment, we hypothesise that it is possible to achieve a meaningful score for any pair of structures regardless of their similarity. For example, an all-alpha structure may be aligned with an all-beta structure; a long alignment will be achieved, and a (poor) score will be realised. It is argued that this method of whole-chain comparison is more meaningful than scoring the similarity/dissimilarity of whole structures based on just small portions of the chains. It is anticipated that such an approach may lead to new insightful descriptions of fold space in the future, and may lead to the development of a new metric for measuring the distance between protein chain structures. The pairwise dissimilarity scoring of protein chains is further considered in Chapter 4.

2.1 Structural Fragments

ProSMART uses structural fragments in order to represent the local structural environments of residues. The concept of structural fragments has been used in structural alignment, and in other applications, as discussed in Chapter 1. Fragments have been defined in various ways. Indeed, the word 'fragment' has other meanings in molecular biology, such as small molecule fragments, as used in fragment-based drug discovery. Even within the field of structural comparison, there is ambiguity regarding the definition of a 'structural fragment'. For example, some methods allow fragments to be sequentially discontinuous, and to be of varying lengths (see §1.2.3). For a discussion of fragments as structural features, in the context used here, see §1.2.2.

Specifically, we consider a fragment to consist of the main chain atoms (i.e. N, C^{α}, C, and O) from n consecutive residues. In implementation, the fragment length (denoted by n) may be chosen by the user. The fact that n may be varied allows the approach to operate at different levels of structural resolution, as desired. Note that the value of n is kept constant for all fragments for any single comparison. Consequently, these constructs comprise $4n$ point landmarks that are directly comparable, irrespective of the particular amino acid sequence of the protein chains. Fragments must be complete, and thus only exist where there are n consecutive residues, with all main chain atoms present.

Fragment indexing may be visualised using a sliding window (n residues in length) along the protein chain, constructed on a per-residue basis (fragments may overlap; see Figure 1). For example, fragment 1 may comprise residues 1 to n; fragment 2 residues 2 to $n+1$, etc. However, the potential heterogeneity of residue numbering in PDB (Berman et al., 2002) files causes complications

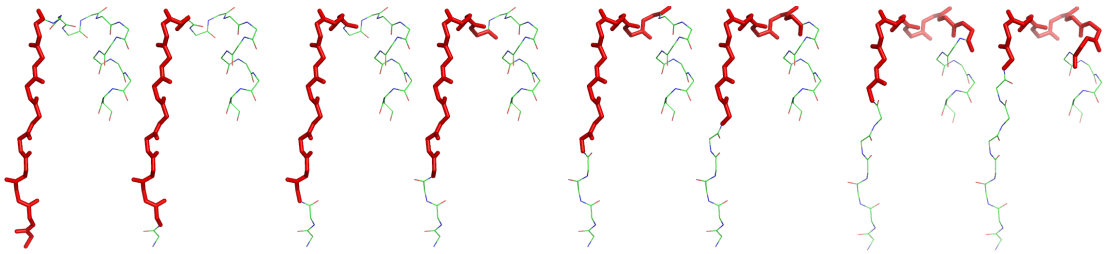


Figure 1: Depiction of the first eight 9-residue fragments (left to right) from 2cex(A), illustrating the sliding window effect of overlapping fragments. Only a small portion of 2cex(A) is shown; residues 2–20, comprising a β-strand, loop and part of an α-helix. Fragments are shown in red.

in the general construction of a set of structural fragments. Such complications include:

- Insertions, in which an insertion code is appended to the residue index;
- Deletions, where residue indexes are missing;
- Missing atoms, e.g. due to regions of high flexibility or disorder;
- Alternative conformations, which induce conceptual one-to-many correspondences.

Insertions, deletions, and missing atoms are dealt with by sensible reindexing of residues, as specified below. In the case of alternative conformations, the first conformation present in the file is selected; any other conformations are ignored. Whilst this may not be the most ideal method of dealing with this situation, any other methods would be more computationally expensive. It is anticipated that the employed techniques should be relatively insensitive to subtle changes in atomic positions, and so it is tentatively assumed that the effect of arbitrarily choosing the first conformation should not overly impact on results. A more suitable approach would be desirable in future.

2.1.1 Residue Reindexing

In order to account for different residue indexing protocols, *ProSMART* reindexes residues into a more unified and workable format as follows:

- All indexes are translated so that the first residue has an index of 1;
- Residue insertion codes are expanded (e.g. $\{1, 1A, 2\} \rightarrow \{1, 2, 3\}$);
- Alternative conformations are removed – only the first present conformation for any particular residue is used;
- If any of the four main chain atoms are missing, then the residue is removed (if only side chain atoms are missing then the residue is still used);
- If a residue is missing, then a separation of n residues is enforced between the surrounding residues (e.g. $\{1, 2, 4, 5\} \rightarrow \{1, 2, 2 + n, 3 + n\}$) so that runs of consecutive residues are well-separated in relation to the chosen fragment length.

As a result, if we denote the set of valid residues present in the structure by $R = \{r_i\}$, then the residue indexes are integral and unique, and residue ordering is maintained:

$$i < j \iff r_i < r_j \qquad i, j \in [1, |R|] \subset \mathbb{N}. \tag{2.1}$$

Furthermore, residue spacing contains deterministic qualitative information pertaining to chain continuity:

$$r_{i+1} = \begin{cases} r_i + 1 & \text{if the corresponding original residues are adjacent,} \\ r_i + n & \text{otherwise.} \end{cases} \tag{2.2}$$

This basic structure is useful for insuring a meaningful final alignment of the two structures, and may be reversed so that results may be provided using the original coding scheme.

2.1.2 Fragment Indexing

Using the new residue indexing scheme, it is now possible to identify the list of structural fragments. In order to ensure that fragments are always complete and comparable, we specify that they may only exist if the n consecutive residues belonging to that fragment exist. Consequently, fragment indexing is based on the presence of runs of consecutively indexed residues. For example, suppose that residue 8 is missing, and that the fragment length is chosen to be $n = 7$ residues, then we might expect that fragment 1 will comprise residues 1 to 7, fragment 2 residues 9 to 15, and fragment 3 residues 10 to 16, etc.

Fragments are identified by specifying the residue-index of the first residue in the fragment. Specifically, the first residue in fragment i is denoted $f_i \in F$, where F is the set of all (valid) fragments in a given protein chain. The indexes f_i may be calculated recursively, according to the following ruleset:

$$f_i = \begin{cases} \min\{r \in R : r, \ldots, r + n - 1 \in R\} & i = 1, \\ \min\{r \in R : r, \ldots, r + n - 1 \in R, r > f_{i-1}\} & i > 1. \end{cases} \tag{2.3}$$

Note that, since the fragments f_i are defined by a single residue index (only the first residue index is required, as the belongingness of the subsequent $n - 1$ residues to the fragment is implicit), the set of fragment identifiers F is a strict subset of the set of residues R. Also, as with residue indexes, strict ordering of fragment indexes is maintained:

$$i < j \iff f_i < f_j \qquad i, j \in [1, |F|]\,. \tag{2.4}$$

Note that the adjacency of residue indexes corresponding to fragment start-positions implies consecutive fragment indexes:

$$f_i + 1 = f_j \quad \Rightarrow \quad j = i + 1 \tag{2.5}$$

although the converse is not true:

$$j = i + 1 \quad \not\Rightarrow \quad f_i + 1 = f_j \tag{2.6}$$

2.1.3 Fragment Scoring

In *ProSMART*, all pairs of fragments between two protein chains are compared by quantifying differences between their main chain atomic coordinates. Specifically, we use a form of Procrustes analysis (Gower and Dijksterhuis, 2004; Gower, 2010) to describe differences between the pairwise distributions of fragment coordinates. Procrustes was originally a code name for the program of Catell and Hurley (1962), named after the mythological Greek villain whose victims were stretched and cut in order to fit the shape of his bed. This analogy is fitting in this context, due to distorting one set of observations relative to the other, by means of translation, rotation and scaling, in order to find the optimal agreement between the two sets.

Here, the method is implemented so as to ensure invariance with respect to translation and rotation of the fragments' original coordinate frames. However, unlike traditional implementations of Procrustes analysis, this implementation is purposely not scale invariant. Scale invariance has proven to be a useful property in other scenarios, such as the shape comparison of natural objects that grow, and images of objects taken from different distances (Dryden and Mardia, 1998). Note that there are other methods for the translation, rotation, and scale invariant comparison of objects, such as the use of Fourier descriptors for the comparison of 2D (Folkers and Samet, 2002) and 3D (Vranić and Saupe, 2001; Li and Hartley, 2006) objects. Fourier descriptors have also been used to allow invariance to an affine transformation (which is also invariant to shearing), useful for the comparison of 2D images of a 3D object taken from different viewpoints (Arbter et al., 1990).

However, scaling is not allowed here because it is desirable to preserve atomic distances. This is due to acknowledging that the distributions of chemical restraints can not be considered to scale under different circumstances. At least, for our purposes, any potential scaling is likely to be dwarfed by the range of fragment conformations. For example, in the extreme case of comparing a helix with a strand, allowing scaling would most likely stretch the helix and shrink the strand (the degree to which would depend on whether scaling was allowed to be anisotropic). This would cause the Procrustes score to be relatively low, which is undesirable since the comparison of a helix and a strand should result in a high dissimilarity score. In general, such scores might imply structures to be more similar than expected, leading to ambiguity in interpretation.

Consequently, the employed method provides a measure of dissimilarity that is invariant to rigid-body transformations. This allows the results to be invariant to the protein structures' particular global conformations, and to their original coordinate frames. In this context, the Procrustes score of a fragment-pair is equivalent to the pairwise RMSD (root mean square deviation) of the corresponding atomic coordinates after superposition (see Figure 2). The score thus represents the rigidity of local structural conservation about a given fragment's central residue. Conceptually, the process of fragment scoring involves translating and rotating the coordinate matrices so that they are optimally superposed, then calculating the average RMSD of corresponding atomic coordinates for use as the fragment dissimilarity measure. However, using the formalism of Procrustes analysis

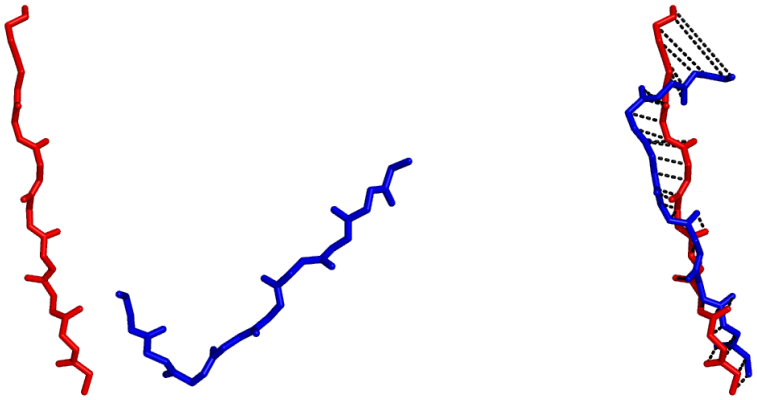


Figure 2: An example illustrating the superposition of fragments (achieved using Procrustes analysis). Left: two 9-residue fragments are shown in different arbitrary coordinate frames; these are the first and third fragments from 2cex(A). Right: the same two fragments, superposed. The longer red fragment remains in the same coordinate frame. The other fragment is translated and rotated so as to minimise the sum of squared distances between corresponding main chain atoms. The RMSD is given by the square root of the average of these squared distances. Distances between corresponding atoms are represented as black dotted lines.

allows the score to be realised without incurring as much computational expense as would arise from physically superposing the atomic coordinates in the traditional fashion, as demonstrated in §3.2.2.

2.1.4 Procrustes Analysis

In preparation for the structural alignment process, we must calculate the fragment dissimilarity matrix $\mathbf{D}$, whose elements are the Procrustes scores of all pairs of fragments between the two protein chains. The procedure for calculating elements of the fragment dissimilarity matrix may be described as follows.

Coordinate matrices corresponding to all fragments in both protein chains are normalised by removing the mean atomic position of the fragment. This means that the coordinates of all fragments are centred at zero, resulting in a set of directly comparable landmark configuration matrices (Dryden and Mardia, 1998), as required. For any fragment $f \in F$, the corresponding normalised coordinate matrix is defined by:

$$\hat{\mathbf{F}} = \mathbf{F} - \vec{\mu}_{\mathbf{F}} \tag{2.7}$$

in which the mean fragment position $\vec{\mu}_{\mathbf{F}}$ is given by:

$$\vec{\mu}_{\mathbf{F}} = \frac{1}{4n} \sum_{k=1}^{4n} \mathbf{F}_{k.} \tag{2.8}$$

where n is the fragment length (in residues), and $\mathbf{F}$ is the $4n \times 3$ coordinate matrix of all main chain atoms belonging to fragment f, in the structure's original coordinate frame.

We compare all fragments f_{1i} (with normalised coordinate matrices $\hat{\mathbf{F}}_{\mathbf{1i}}$) in protein 1 with all fragments f_{2j} (with normalised coordinate matrices $\hat{\mathbf{F}}_{\mathbf{2j}}$) in protein 2, in order to construct the fragment dissimilarity matrix $\mathbf{D}$. Specifically, the elements of $\mathbf{D}$ are given by the Procrustes distances:

$$\mathbf{D}_{ij} = \sqrt{\frac{\mathrm{tr}\left(\hat{\mathbf{F}}_{\mathbf{1i}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{1i}}\right) + \mathrm{tr}\left(\hat{\mathbf{F}}_{\mathbf{2j}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{2j}}\right) - 2\mathrm{tr}\left(\mathbf{S}_{\mathbf{ij}}\right)}{4n}} \tag{2.9}$$

where $\mathbf{S}_{\mathbf{ij}}$ is the diagonal matrix of singular values obtained from the singular value decomposition (see Stewart, 1993) of $\hat{\mathbf{F}}_{\mathbf{2j}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{1i}}$, given by:

$$\hat{\mathbf{F}}_{\mathbf{2j}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{1i}} = \mathbf{U}_{\mathbf{ij}}\mathbf{S}_{\mathbf{ij}}\mathbf{V}_{\mathbf{ij}}^{\mathbf{T}} \tag{2.10}$$

where $\mathbf{U}_{\mathbf{ij}}$ and $\mathbf{V}_{\mathbf{ij}}$ are 3×3 orthogonal matrices. Note that we do not need to physically superpose (rotate) the fragment coordinate matrices in order to achieve the fragment score.

Two example fragment dissimilarity matrices are shown in Figure 3. Rectangles of relatively well-scoring fragment-pairs are observed, due to the presence of repetitive structure. The bright red rectangles evident in the second matrix correspond to pairs of helical fragments, since all such fragment-pairs score relatively well. Only one such region is present in the first matrix, due to the presence of only one helix in each structure (at the C-termini). The less intensely red rectangles in the first matrix correspond to the comparison of β-strand fragments. Less randomly-favourable pairings (generally helix-strand, helix-loop, strand-loop, and loop-loop) correspond to the yellow/white regions. Diagonal red lines are observed between the red rectangles, generally corresponding to randomly well-scoring loop regions.

Whilst not required for the calculation of the fragment score, it is important to acknowledge that the rotation matrix $\mathbf{R}_{\mathbf{ij}} \in SO(3)$ required to optimally superpose the two fragments may be calculated as (Challis, 1995):

$$\mathbf{R}_{\mathbf{ij}} = \mathbf{U}_{\mathbf{ij}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & |\mathbf{U}_{\mathbf{ij}}||\mathbf{V}_{\mathbf{ij}}| \end{bmatrix} \mathbf{V}_{\mathbf{ij}}^{\mathbf{T}} \tag{2.11}$$

which is an orthogonal matrix with determinant unity ($\mathbf{R}_{\mathbf{ij}}\mathbf{R}_{\mathbf{ij}}^{\mathbf{T}} = \mathbf{I}$, $|\mathbf{R}_{\mathbf{ij}}| = 1$), as required. The diagonal matrix (the central term) is required in order to ensure that the resulting matrix $\mathbf{R}_{\mathbf{ij}}$ is indeed a rotation, and not a rotoreflection. Note that if a rotoreflection is more favourable than a rotation (i.e. $|\mathbf{U}_{\mathbf{ij}}||\mathbf{V}_{\mathbf{ij}}| = -1$) then the fragments are most likely dissimilar. This observation has been previously acknowledged and exploited for very long fragments (Maiorov and Crippen, 1994). The matrix $\mathbf{R}_{\mathbf{ij}}$ rotates $\hat{\mathbf{F}}_{\mathbf{2j}}$ onto $\hat{\mathbf{F}}_{\mathbf{1i}}$ so that the superposed coordinate matrix for fragment f_{2j} in the coordinate frame of fragment f_{1i} is given by: $\hat{\mathbf{F}}_{\mathbf{2j}}\mathbf{R}_{\mathbf{ij}} + \vec{\mu}_{\mathbf{F_{1i}}}$.

Note that this method of superposition is valid for any pair of comparable coordinate matrices (not just for fragments), and thus is used by *ProSMART* for superposing the protein structures according to their final residue alignment. Given a rotation $\mathbf{R}$ and translation $\vec{t}$, it is possible to

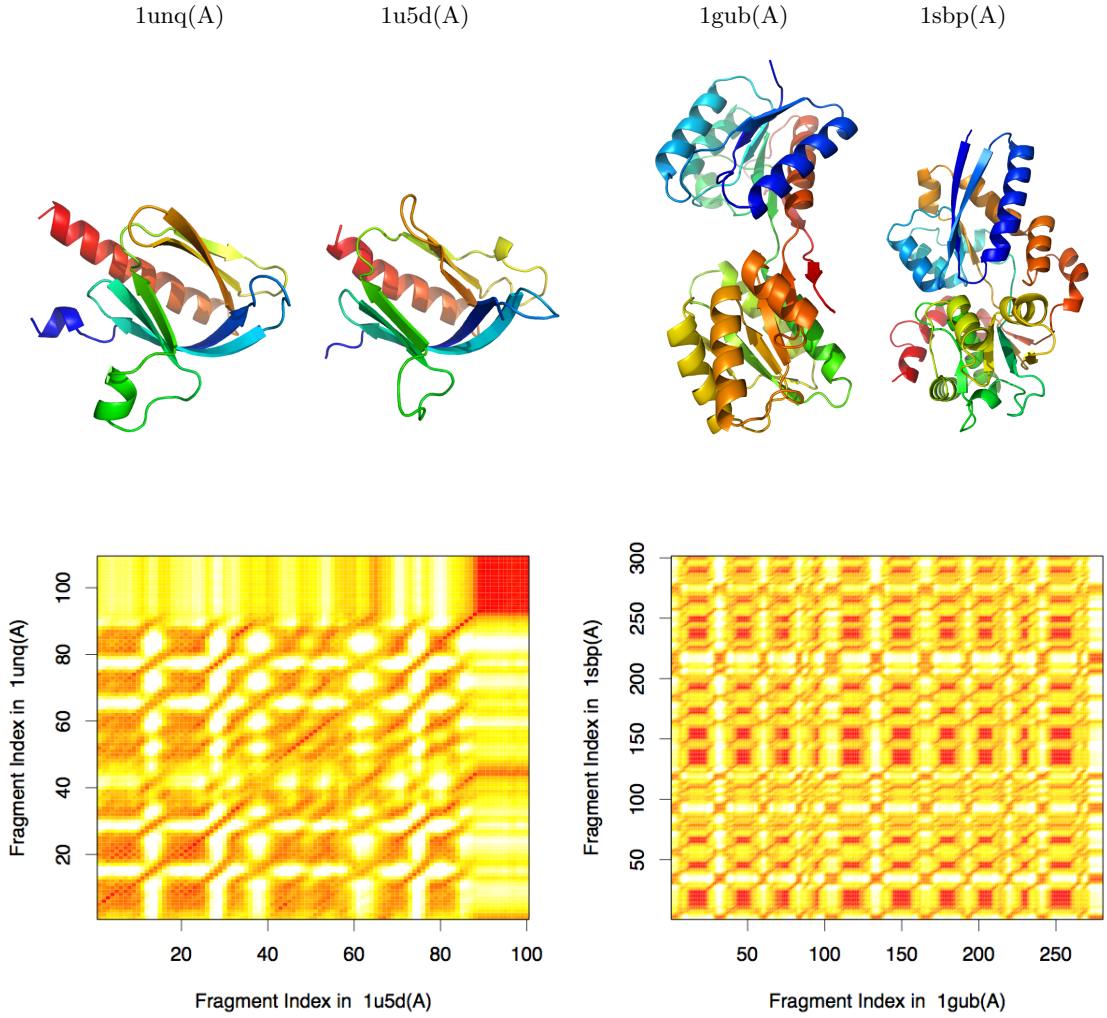


Figure 3: Depictions of two example fragment dissimilarity matrices, for two structurally similar chains 1unq(A) and 1u5d(A) (left), and two dissimilar unrelated chains 1gub(A) and 1sbp(A) (right), for $n = 9$. Illustrations of the compared structures are shown above, rainbow coloured along the chain from blue (N-termini) to red (C-termini). The dissimilarity matrices are coloured according to Procrustes distance; red indicates low distance $\mathbf{D}_{ij} \approx 0$, yellow intermediate, and white high distance $\mathbf{D}_{ij} \geq 4$.

express the superposed coordinates $\mathbf{F}'$ in terms of the original coordinate matrix $\mathbf{F}$:

$$\mathbf{F}' = \mathbf{F}\mathbf{R} + \vec{t} - \vec{\mu}_{\mathbf{F}}\mathbf{R} \tag{2.12}$$

where $\vec{\mu}_{\mathbf{F}}$ is the mean of $\mathbf{F}$. This is more computationally efficient, due to not having to translate $\mathbf{F}$ prior to rotation.

Modifications Allowing Alternative Scoring Methods

ProSMART provides two extra alternative scoring methods, for flexibility. The differences between these options amount to the choice of coordinate matrix representing a fragment for superposition and scoring. The alternative methods are implemented due to the observation that the four main chain atoms each have different properties, in terms of both conformational degrees of freedom and

reliability of position.

The first alternative method uses only C^α atoms for fragment superposition and scoring. Specifically, the $4n \times 3$ coordinate matrix $\mathbf{F}$ that represents a particular fragment $f \in F$ is replaced by the $n \times 3$ coordinate matrix of only C^α atomic positions. These features operate at a lower level of structural resolution, utilising less detail than the default method. Note that the size of the fragment matrix is reduced due to the utilisation of fewer atomic positions, resulting in reduced computation time (see §3.2.2).

The second alternative method uses all main chain atoms for superposition, but only C^α atoms for scoring. All main chain atoms are used for superposition in an attempt to increase robustness. However, in the absence of a sensible method of weighting different atom types, only C^α atoms are used for scoring, as it was speculated that they may be the most sensitive to conformational changes. This is achieved by using a generalised version of the Procrustes distance that allows coordinate weighting:

$$\mathbf{D}_{ij} = \sqrt{\frac{\mathrm{tr}\left(\hat{\mathbf{F}}_{\mathbf{1i}}^{\mathbf{T}}\mathbf{W}\hat{\mathbf{F}}_{\mathbf{1i}}\right) + \mathrm{tr}\left(\hat{\mathbf{F}}_{\mathbf{2j}}^{\mathbf{T}}\mathbf{W}\hat{\mathbf{F}}_{\mathbf{2j}}\right) - 2\mathrm{tr}\left(\hat{\mathbf{F}}_{\mathbf{1i}}^{\mathbf{T}}\mathbf{W}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{2j}}\mathbf{R}_{\mathbf{ij}}\right)}{\mathrm{tr}\left(\mathbf{W}\right)}} \tag{2.13}$$

where $\mathbf{W}$ is a diagonal matrix of weights. Note that if all atoms are assigned equal weights ($\mathbf{W} \propto \mathbf{I}$) then this equation is equivalent to Equation (2.9). Coordinates are weighted by multiplying the distance between the coordinate and the origin by a scale factor, causing particular atoms to have relatively more (or less) influence. A weight of zero causes both corresponding coordinates to lie at the origin, resulting in zero contribution towards the RMSD.

Note that $\mathbf{W}$ could in principle have non-zero non-diagonal elements. In this case, the denominator of Equation (2.13) would be the sum of matrix elements rather than the trace. For example, setting all elements of $\mathbf{W}$ to unity would result in minimising the RMSD of all atoms in fragment 1 against all atoms in fragment 2, rather than assuming a one-to-one atomic correspondence. In contrast, having some negative non-diagonal elements would result in weighted repulsion of specific atom-pairs. Whilst such an approach may have some applications, it is not applicable in our situation.

Specifically, in this case, $\mathbf{W}$ is a $4n \times 4n$ binary diagonal matrix comprising the value 1 every fourth diagonal element (corresponding to a C^α atom) and 0 elsewhere, so that all main chain atoms are used for calculation of the superposition transformation, but only C^α atoms are used for scoring (since all non-C^α atoms are assigned a weight of zero). The rotation matrix $\mathbf{R_{ij}}$ is the rotation matrix that rotates $\hat{\mathbf{F}}_{\mathbf{2j}}$ onto $\hat{\mathbf{F}}_{\mathbf{1i}}$, calculated using Equation (2.11) (note that the weights $\mathbf{W}$ are not used for calculation of the rotation matrix).

It should be noted that, unlike with the other two scoring methods, here it is necessary to explicitly calculate and utilise the rotation matrix $\mathbf{R_{ij}}$ in order to achieve the score. Consequently, this method is more computationally expensive than the other fragment scoring methods discussed above (see §3.2.2).

2.2 Fragment Alignment

The employed alignment algorithm was designed with the intention of satisfying the objectives outlined in §1.4. Our solution is to find an optimal fragment correspondence between the two compared chains, according to minimisation of the net Procrustes score of aligned fragment-pairs. Our approach imposes the constraint that the alignment must maintain sequence ordering (due to our decision to use dynamic programming for alignment), which is deemed a satisfactory condition/limitation. Subject to this constraint, we aim to identify an alignment with the following desirable properties:

1. Fragment alignment must be one-to-one, so that any one fragment cannot be aligned to multiple fragments from the other protein;
2. Alignment implies a one-to-one residue correspondence, allowing unaligned residues;
3. Alignment length must be maximised, subject to maintaining criteria (1) and (2);
4. Sum of aligned fragment dissimilarity scores is minimised, subject to maintaining criteria (1), (2) and (3).

Importantly, note that we maximise the alignment length in order to achieve a global score that suitably penalises any Procrustes dissimilarity that might be identified between a chain-pair. Such a condition is required, otherwise the minimum of the sum of aligned fragment dissimilarity scores would be achieved when the alignment is of length zero. This alignment length maximisation is fundamental to the approach, as discussed above.

Since a score is realised for each aligned residue-pair, there is no reason why a score threshold could not be subsequently applied in order to filter the final alignment. This would only be appropriate when desirable to only identify sufficiently structurally similar regions. However, by default there is no score threshold, allowing all aligned residues to be returned regardless of the score – this is an important feature of *ProSMART*.

In searching for the optimal alignment, it is impractical to consider all possible alignments, due to combinatorial explosion. Rather, techniques are employed in order to find a reasonable approximate solution. Ideally, this should be equal to the optimal solution (which has minimum sum of Procrustes scores, subject to the above criteria). In practice, we aim to achieve a solution such that the achieved score is close to that of the optimal solution. Such a solution is considered sufficient, especially in scenarios where global statistics are of primary interest, e.g. database scanning, pairwise chain dissimilarity scoring, and structural classification.

In *ProSMART*, fragment alignment is achieved using multiple stages (see Figure 4):

- Gap penalty assignment – to specify any bias to be included in the dynamic programming stage;
- Dynamic programming – to achieve the optimal path subject to maintenance of sequence order;
- Path filtering – to identify the initial one-to-one alignment between fragments;
- Refinement – using fragment 'segments', in order to minimise the sum of Procrustes scores whilst maximising the alignment;
- Clash removal – to ensure an implicit one-to-one residue correspondence;
- Optimisation – in order to minimise the sum of Procrustes scores whilst maximising the alignment, ensuring the continued maintenance of a one-to-one residue correspondence.

The residue alignment may then be directly inferred from the final fragment alignment. In the remainder of this section, we shall consider the methods employed to identify the initial fragment alignment; the refinement and optimisation stages will be treated in subsequent sections.

Note that the four stages following dynamic programming are required, since the fragment alignment resulting from dynamic programming may not be the optimal solution when allowing breaks in the path. Furthermore, a unique fragment alignment may not imply a unique residue alignment. These issues will be discussed further below. The alignment stage may optionally be bypassed in the case of sequence-identical chains; this is beneficial as it avoids any undesirable artefacts that may arise (e.g. due to randomness, errors in atomic positions, effects of conformational change), and is faster.

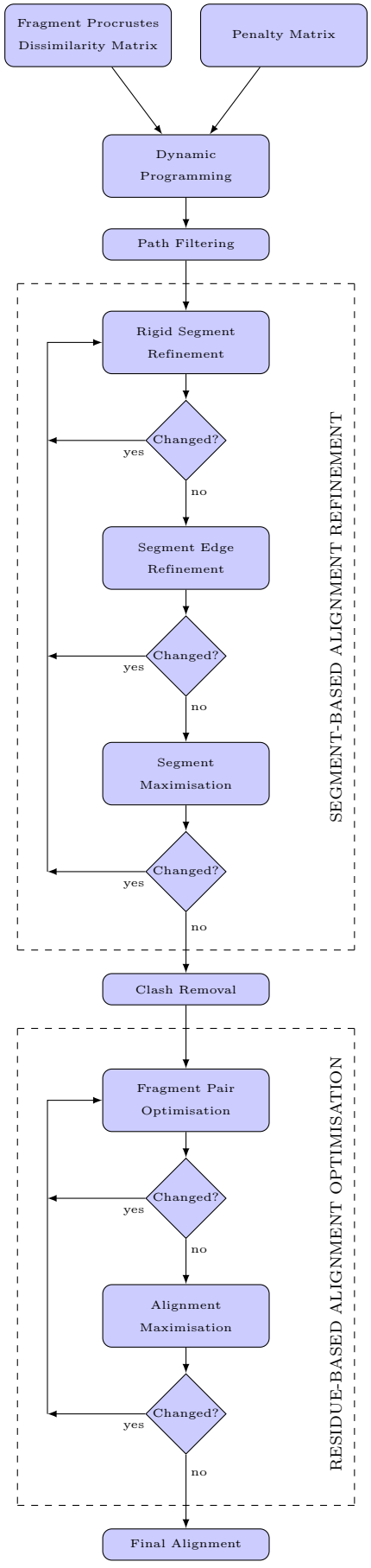


Figure 4: Alignment flow chart.

2.2.1 Gap Penalties

The employed approach allows a penalty to be assigned to non-consecutive alignments (i.e. penalise gaps in the alignment). *ProSMART* does not use a general gap penalty in the conventional sense. This helps to ensure that insertions and deletions are dealt with sensibly, and that the achieved alignment is indeed the best-scoring alignment. However, repetitive structure causes problems with alignment. Specifically, any helical fragment will be well-aligned to all fragments in an α-helix. Since the alignment scores will be very similar throughout the helix, variations in atomic positions and small conformational changes can cause an 'incorrect' alignment to score better, particularly if the aligned helices are of different lengths. Consequently, it is easy to align α-helices in an intuitively incorrect way (despite actually being the most favourable conformation, in terms of local structural similarity). *ProSMART* deals with this situation by allowing the assignment of a gap penalty in helical regions. *ProSMART* tolerates this gap penalty, since the effect on the overall alignment is negligible. This ensures that helical fragments will always be aligned consecutively unless there is strong evidence suggesting the contrary (e.g. a point residue insertion, or a kink in the helix). Despite also being repetitive structural units, this issue is less prevalent with β-strands due to their higher degree of conformational flexibility; observed structural similarity of β-strand fragments implies a higher degree of non-randomness. Note that the helix gap penalty does not contribute to any scores outside of the dynamic programming algorithm.

Specifically, we apply a gap penalty if the current and previous fragments are considered to be helical in both proteins. A fragment i is considered to be helical if the Procrustes score D_i^{helix} between the fragment and an ideal helix is sufficiently small, i.e. $D_i^{\text{helix}} \leq k^{\text{helix}}$, where D_i^{helix} is given by:

$$D_i^{\text{helix}} = \sqrt{\frac{\text{tr}\left(\hat{\mathbf{F}}_\mathbf{i}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{i}\right) + \text{tr}\left(\hat{\mathbf{F}}_\mathbf{helix}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{helix}\right) - 2\text{tr}\left(\mathbf{S}_\mathbf{i}^\mathbf{helix}\right)}{4n}} \tag{2.14}$$

where $\mathbf{S}_\mathbf{i}^\mathbf{helix}$ is the diagonal matrix of singular values obtained from the singular value decomposition of $\hat{\mathbf{F}}_\mathbf{helix}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{i}$, similarly to in Equation (2.10), and $\hat{\mathbf{F}}_\mathbf{helix}$ is the normalised matrix of coordinates of an 'ideal' helix of length n residues. Coordinates of an ideal helix were generated by *COOT* (Emsley et al., 2010). It should be acknowledged that the ideal helix may not be representative of the average helical conformation; it may prove more appropriate to identify helices using an empirical approach rather than one based on ideal conformation. However, the employed method is deemed sufficient for this purpose. Note that rather than using a traditional method of identifying helical fragments based on non-structural information (i.e. hydrogen bonding patterns), they are purposely identified using only structural information in a way that is consistent with the general approach.

For each chain, we calculate the vector $\vec{D}^{\text{helix}}$ of scores that is used to identify whether or not fragments are considered to be helical, termed $\vec{D}_1^{\text{helix}}$ and $\vec{D}_2^{\text{helix}}$, respectively. We may now construct a penalty matrix $\mathbf{P}$, whose elements $\mathbf{P}_{ij}$ correspond to the potential non-consecutive

alignment of fragment i in protein 1 with fragment j in protein 2, defined by:

$$\mathbf{P}_{ij} = \begin{cases} p & D^{\text{helix}}_{1,i}, D^{\text{helix}}_{1,i-1}, D^{\text{helix}}_{2,j}, D^{\text{helix}}_{2,j-1} \leq k^{\text{helix}}, \text{ if } i,j \geq 2 \\ 0 & \text{otherwise,} \end{cases} \tag{2.15}$$

where p is the gap penalty, and k^{helix} is the dissimilarity threshold determining whether or not a fragment is considered helical.

In principle, other gap penalties could be used in order to create a more complex method of penalising gaps in the alignment. However, given present knowledge, the general use of gap penalties conflicts with our fundamental objective of creating a sensible global dissimilarity score between arbitrary structures, and thus shall not be used.

2.2.2 Dynamic Programming Algorithm

Dynamic programming algorithms have been commonly used in biology, for both sequence and structural alignment (see §1.2.3). Differences between implementations often lie in the choice of input matrix that defines the similarity/dissimilarity of features. In sequence alignment (and structural sequence alignment), this feature-based score matrix inherits values from a smaller matrix of predetermined pairwise letter scores, often referred to as substitution or transition matrices. Such matrices commonly used in sequence alignment include *PAM* (Dayhoff and Schwartz, 1978) and *BLOSUM* (Henikoff and Henikoff, 1992), and in structural alignment that of *SARST* (Lo et al., 2007). In contrast, substitution matrices are generally not used for structural alignment, since features are not discretely categorised. Rather, a feature-based scoring scheme (commonly based on interatomic distances) is adopted, so that each feature-pair is assigned a specific score.

Dynamic programming score matrices often comprise similarity scores, potentially allowing both positive and negative values (depending on particular implementation). The score matrix used in *ProSMART* comprises Procrustes dissimilarity scores, and thus elements are bounded below by zero. An upper bound is not explicitly enforced or known, yet exists according to the conformational range of naturally occurring n-residue fragments. Using the raw Procrustes distance as the dissimilarity score means that the score is based on a natural measure of dissimilarity, and not dependent on arbitrary parameters or functional forms. The dynamic programming algorithm implemented in *ProSMART* is a customised version of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970).

The objective of the dynamic programming algorithm is to find the optimal path through the fragment dissimilarity matrix $\mathbf{D}$, beginning at position $(1,1)$ and ending at position (N_1, N_2), where $N_1 = |F_1|$ and $N_2 = |F_2|$ are the numbers of fragments in protein chains 1 and 2, respectively. In our implementation, this is achieved using a dynamic programming (Bertsekas, 2005) algorithm that performs a closed-loop minimisation on the path through the dissimilarity matrix. In other words, we are able to find the unique, optimal, one-to-many fragment correspondence between the two proteins.

However, since this method does not optimise a one-to-one fragment correspondence our fundamental criteria are not satisfied, and the best-scoring fragment alignment is not realised. Despite the fact that the optimal alignment is not necessarily implied by the alignment achieved from dynamic programming, it does give a very good approximation to the solution, and suffices well as an initial alignment. The alignment may then be optimised in subsequent refinement stages.

The aim of the dynamic programming stage is to identify the path $P = \{p_k\}$ of $(i, j) \subset \mathbb{N}^2$ indexes through the dissimilarity matrix $\mathbf{D}$ that minimises the total sum of the scores:

$$D_P = \sum_{k=1}^{|P|} \mathbf{D}_{p_k} \to \min \tag{2.16}$$

subject to the conditions that $p_1 = (1, 1)$, $p_{|P|} = (N_1, N_2)$, and $p_k = (i, j)$ implies $p_{k+1} \in \{(i + 1, j), (i, j + 1), (i + 1, j + 1)\}$ for all $k \in [1, |P| - 1]$.

The employed dynamic programming algorithm may be described as follows. We begin by constructing a $N_1 \times N_2$ cost matrix $\mathbf{C}$ to represent the costs, in terms of the cumulative Procrustes fragment dissimilarity score, of all alignment paths through the dissimilarity matrix $\mathbf{D}$. The elements of the cost matrix may be calculated recursively according to the following rules:

$$\mathbf{C}_{ij} = \begin{cases} \mathbf{D}_{1j} & i = 1 \\ \mathbf{D}_{i1} & j = 1 \\ \mathbf{D}_{ij} + \min\left\{\mathbf{C}_{i,j-1} + \mathbf{P}_{ij}, \mathbf{C}_{i-1,j} + \mathbf{P}_{ij}, \mathbf{C}_{i-1,j-1}\right\} & \text{otherwise.} \end{cases} \tag{2.17}$$

The optimal path P can then be calculated recursively backwards, beginning with the boundary condition $p_{|P|} = (N_1, N_2)$, which corresponds to the last element of the cost matrix, $\mathbf{C}_{N_1 N_2}$. Suppose we know that $p_k = (i, j)$ is an element of the optimal path, then the previous alignment p_{k-1} in the path will either be $(i - 1, j)$, $(i, j - 1)$, or $(i - 1, j - 1)$. The correct alignment is the one that has the minimum associated cost. This decision making process may be expressed:

$$p_k = \arg\min_{(i,j)} \left\{\mathbf{C}_{ij}|_{(i+1,j)=p_{k+1}}, \mathbf{C}_{ij}|_{(i,j+1)=p_{k+1}}, \mathbf{C}_{ij}|_{(i+1,j+1)=p_{k+1}}\right\} \tag{2.18}$$

resulting in the unique optimal path P that satisfies Equation (2.16). Note that the length of path P is bounded: $\max\{N_1, N_2\} \leq |P| \leq N_1 + N_2 - 2$.

Path Filtering to Identify the Initial Alignment

The resultant path P is filtered so that any fragment from one chain is aligned to, at most, one fragment from the other chain. This is achieved by removing all clashing elements of the path keeping only the one that scores most favourably from each clash. The corresponding list of fragment-pair indexes thus represents the initial fragment alignment.

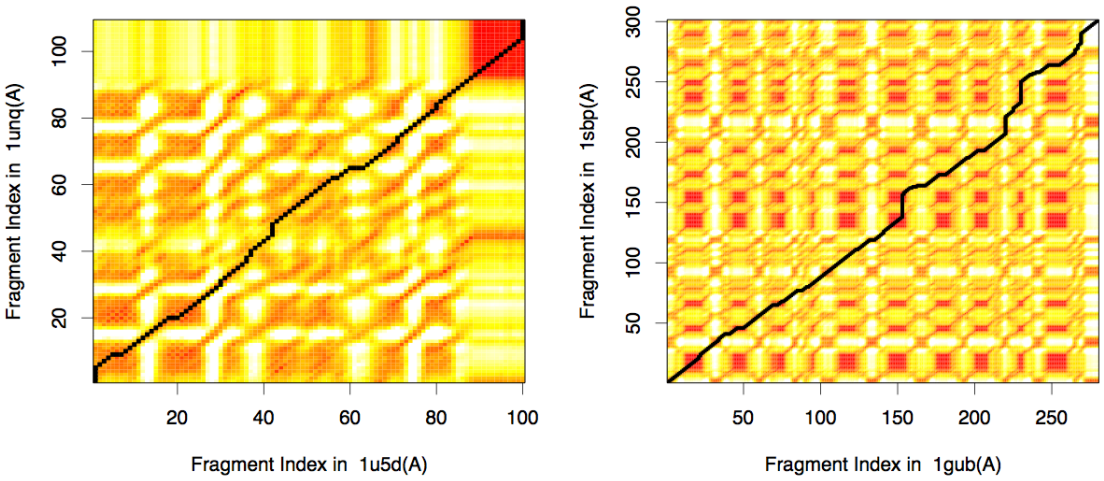
Specifically, we identify the alignment $A = \{a_k\} \subseteq P$ that constitutes a one-to-one fragment correspondence between the two proteins. In order to achieve A, we must consider all elements of P that clash with other elements. Without loss of generality, suppose path P implies that fragment i from protein 1 is aligned to fragments j and $j + 1$ from protein 2, that is, element $p_k = (i, j)$ is

followed by element $p_{k+1} = (i, j+1)$. Then either p_k or p_{k+1} must be removed from the alignment; p_k is removed if $\mathbf{D}_{p_k} > \mathbf{D}_{p_{k+1}}$, otherwise p_{k+1} is removed. By symmetry, the equivalent is true if $p_k = (i, j)$ is followed by element $p_{k+1} = (i+1, j)$. By repeating this elimination process for all elements of P, we achieve the one-to-one set:

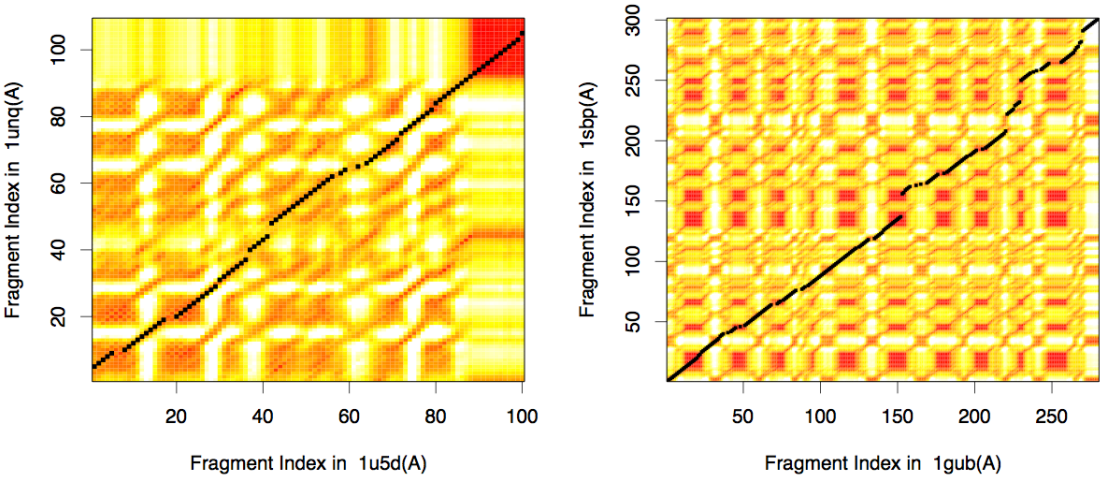
$$A = \{(i,j) \in P : \mathbf{D}_{ij} < \mathbf{D}_{ik}, \mathbf{D}_{ij} < \mathbf{D}_{lj}, \forall k, l : (i,k), (l,j) \in P\} \tag{2.19}$$

that constitutes our initial fragment alignment. Note that $A \equiv P$ if and only if $N_1 = N_2$ and $p_k = (i, i)$ for $i = 1, \ldots, N_1$.

Two example optimal paths are illustrated in Figure 5, corresponding to the examples introduced



(a) Optimal paths from dynamic programming



(b) Initial fragment alignment after path filtering

Figure 5: Depictions of example fragment dissimilarity matrices, for two structurally similar chains 1unq(A) and 1u5d(A) (upper and lower left), and two dissimilar unrelated chains 1gub(A) and 1sbp(A) (upper and lower right), for $n = 9$. The matrices are coloured according to Procrustes distance; red indicates low distance $\mathbf{D}_{ij} \approx 0$, yellow intermediate, and white high distance $\mathbf{D}_{ij} \geq 4$. Black dots correspond to aligned fragment-pairs, specifically the optimal paths P (a), and the filtered initial alignment A (b).

in Figure 3. The left image corresponds the the alignment of similar structures; the right to dissimilar structures.

Path P is the unique solution to Equation (2.16). However, whilst generally giving a good approximation, the alignment A is not necessarily the unique solution to our problem of finding the best-scoring one-to-one fragment correspondence. Consequently, we refine the alignment in order to find a closer solution to the minimisation problem:

$$D_A = \sum_{k=1}^{|A|} \mathbf{D}_{a_k} \rightarrow \min . \tag{2.20}$$

subject to the initial criteria outlined at the beginning of §2.2.

The refinement steps employed by *ProSMART* aim to optimise D_A, i.e. to find the local minimum of the sum of the scores of aligned fragments. It is supposed that this method should be largely successful providing the initial dynamic programming stage is able to find a solution sufficiently close to the global minimum.

2.2.3 Segment-Based Alignment Refinement

Various refinement stages are implemented in order to iteratively approach D_A. The employed refinement algorithms find a unique deterministic solution; the process does not simply halt when the solution is deemed sufficiently well-refined. Whether or not this unique solution corresponds to the global extrema depends on the initial alignment, and the specific heuristics employed. In general, these refinement stages search for better residue correspondences (subject to the same Procrustes score criteria as above), ensuring that the alignment score cannot become worse after an iteration. The one exception to this is the alignment lengthening process (required according to our original criteria), in which the overall score inherently becomes worse, as desired.

Due to the dynamic nature of the alignment A during the refinement process, we temporarily change to a more efficient logical storage method at this point. Specifically, we stop referring to aligned fragment pairs, and instead consider aligned 'segments'. These segments are runs of consecutively aligned fragment pairs, so that we only need to store three pieces of information for each segment (the indexes of the first fragment-pair, and the segment length), rather than two for each aligned fragment pair (the indexes of the two fragments). This results in a much more efficient method, particularly when longer runs of fragments are aligned consecutively. This ensures greatly reduced computational expense during the refinement stages, without compromising quality of results.

During segment-based refinement, we do not force the fragment alignment to maintain a one-to-one residue correspondence. By doing so, computation time is greatly reduced. Furthermore, we allow the space on which refinement occurs to be homogeneous; otherwise, the heterogeneity incurred by alignment discontinuities would hinder potential refinement opportunities. In summary, the process of segment-based refinement is considered a fast and effective way to approach a better

minimum of the alignment score function.

Using our new logical representation for the fragment alignment, the set A of aligned fragments, defined by Equation (2.19), may be equivalently expressed as the set S of aligned segments, defined by:

$$A \equiv S = \Big\{ (i,j,m) \in \mathbb{N}^3 : (i,j) \in A, (i-1,j-1) \notin A, \\ m = \max\{k \in \mathbb{N} : (i+x, j+x) \in A, \forall x \in [1,k]\} \Big\}. \tag{2.21}$$

where i and j are the indexes of the first fragment-pair, and m is the segment length (the number of consecutively aligned fragments).

The total score corresponding to the initial alignment A, as defined in Equation (2.20), may be re-expressed as:

$$D_A \equiv D_S = \sum_{k=1}^{|S|} D_{s_k} \tag{2.22}$$

where D_{s_k} is the sum of scores corresponding to the alignment segment s_k, given by:

$$D_{s_k} = \sum_{x=0}^{m_k - 1} \mathbf{D}_{i_k+x, j_k+x} \tag{2.23}$$

where $s_k = (i_k, j_k, m_k) \in S$. Now, the problem of minimising $D_A \equiv D_S$ may be approximated by the problem of co-minimising the D_{s_k}. The aim of this refinement procedure is to apply permutations and mutations to the segments s_k in order to minimise D_S, subject to criteria, thus achieving a more favourable alignment.

One major challenge is to refine the alignment without incurring too much computational expense. In order to achieve this objective, the employed algorithms should be:

- Simple enough to be relatively quick;
- Sufficiently effective in order to justify the added computational expense;
- Different enough to complement each other, forming an overall effective procedure.

In the current implementation, three refinement stages are employed:

- *Rigid Segment Refinement* – shifts whole fragment segments in search of better scoring alignments in the local vicinity, whilst maintaining the segment length and avoiding clashes with other existing segments;
- *Segment Edge Refinement* – attempts to co-minimise consecutive segment pairs by swapping the first and last alignments at the edges of consecutive segments;
- *Segment Maximisation* – segments are lengthened as appropriate, where possible, in order to satisfy the maximisation criteria.

Details of these three stages are given below.

Note that segments may disappear during the refinement process, i.e. $s_k = (i_k, j_k, 0)$ for some k. This situation is perfectly valid, and usually indicates that an incorrectly aligned segment from the initial alignment has been corrected by the refinement stages. In this case, the segment is removed from the alignment and the remaining segment indexes are adjusted accordingly. Critically, a one-to-one correspondence is maintained between fragments at all times during the refinement process – the three stages are such that they do not allow any one fragment to be aligned to multiple other fragments.

The three refinement stages are called recursively (see Figure 4) until no more changes are made to the segment set S, indicating that the local minimum has been found. The functions representing the three employed refinement stages are described below. It should be noted that careful consideration was given to the order in which these three stages appear in the algorithm in order to maximise the effectiveness of the overall procedure.

The 'rigid segment' and 'segment edge' refinement stages were designed so as to ensure that the alignment may score better, but will certainly not score worse, in subsequent refinement iterations, that is:

$$D_{S_x} \leq D_{S_{x-1}} \tag{2.24}$$

for all x, where S_x is the state of the segment set S after x iterations of refinement.

Rigid Segment Refinement

In rigid segment refinement, we attempt to translate rigid consecutive runs of aligned fragments into more favourable alignments. Specifically, we modify the set of segments S in order to reduce the total dissimilarity score D_S, without changing the number of segments $|S|$ or the segment lengths m_k. To refine an individual segment $s_k = (i_k, j_k, m_k)$, we adjust i_k and j_k so as to minimise the corresponding segment score D_{s_k}. Upper and lower bounds for the i_k and j_k are well-defined, due to the requirement for any segment s_k not to clash with the adjacent segments s_{k-1} and s_{k+1}. This means that s_k is subject to the following constraints during rigid segment refinement:

$$\begin{aligned} i_{k-1} + m_{k-1} \leq i_k \leq i_{k+1} - m_k \\ j_{k-1} + m_{k-1} \leq j_k \leq j_{k+1} - m_k \end{aligned} \qquad 1 < k < |S| \tag{2.25}$$

with boundary conditions:

$$\begin{aligned} i_1, j_1 &\geq 1 \\ i_{|S|} &\leq N_1 - m_{|S|} + 1 \\ j_{|S|} &\leq N_2 - m_{|S|} + 1 \end{aligned} \tag{2.26}$$

The unique solution to the rigid segment refinement problem is the set of segments S^{new}, given set S^{old}, that minimises the total dissimilarity score $D_{S^{new}}$, subject to the following conditions:

1. the number of segments remains unchanged ($|S^{new}| = |S^{old}|$);

2. the individual segment lengths m_k are equal to those in the original set;

3. the above constraints (2.25) and (2.26) are satisfied.

However, the number of combinations to be considered as potential solutions can be large, depending on the particular pair of protein chains to be aligned. In practice, the identification of the unique set of segments S^{new} that is the solution to the above conditions would be extremely computationally expensive, thus jeopardising the original objectives of the software tool.

In order to simplify the procedure computationally, we stipulate for the segments to be refined individually, rather than co-refined. Specifically, the segments are refined in consecutive order of the segment index k. This means that we may replace the constraints (2.25) with a modified version:

$$\begin{aligned} i_{k-1}^{new} + m_{k-1} \leq i_k^{new} \leq i_{k+1}^{old} - m_k \\ j_{k-1}^{new} + m_{k-1} \leq j_k^{new} \leq j_{k+1}^{old} - m_k \end{aligned} \qquad 1 < k < |S^{old}| \tag{2.27}$$

keeping the same boundary conditions (2.26), where the segments $(i_k^{old}, j_k^{old}, m_k) \in S^{old}$ are transformed to the equivalent segments $(i_k^{new}, j_k^{new}, m_k) \in S^{new}$ during the procedure. This simple modification, whilst not necessarily providing the optimal solution to the original problem, does provide a reasonable approximation and greatly reduces the number of segment combinations to be considered.

Sometimes, during rigid segment refinement, consecutive fragment segments become combinable, i.e. segments are translated so as to form a single longer consecutive run of aligned fragments. Specifically, this occurs for segments s_k and s_{k+1} when $i_k + m_k = i_{k+1}$ and $j_k + m_k = j_{k+1}$. In this situation, we splice the segments so as to form one single larger segment. In implementation, s_k is lengthened, s_{k+1} deleted, and segments $s_{k+2}, \ldots, s_{|S|}$ reindexed so as to incorporate this change. The procedure employed for rigid segment refinement may be described by Algorithm 1.

Algorithm 1 Rigid Segment Refinement

$s_1 \leftarrow \arg\min_{(i,j,m_1)} \{D_{(i,j,m_1)} : i \in [1, i_2 - m_1], j \in [1, j_2 - m_1]\}$
for $k = 2 \ldots |S| - 1$ **do**
  $s_k \leftarrow \arg\min_{(i,j,m_k)} \{D_{(i,j,m_k)} : i \in [i_{k-1} + m_k, i_{k+1} - m_k], j \in [j_{k-1} + m_k, j_{k+1} - m_k]\}$
end for
$s_{|S|} \leftarrow \arg\min_{(i,j,m_{|S|})} \left\{D_{(i,j,m_{|S|})} : i \in [i_{|S|-1} + m_{|S|}, N_1 - m_{|S|} + 1], j \in [j_{|S|-1} + m_{|S|}, N_2 - m_{|S|} + 1]\right\}$
for $k = 1 \ldots |S| - 1$ **do**
  if $i_k + n_k = i_{k+1}$ **and** $j_k + n_k = j_{k+1}$ **then**
    $n_k \leftarrow n_k + n_{k+1}$
    for $x = k + 1 \ldots |S| - 1$ **do**
      $s_x \leftarrow s_{x+1}$
    end for
    $s_{|S|} \leftarrow$ NULL
    $k \leftarrow k - 1$
  end if
end for
return S

Segment Edge Refinement

The aim of the second segment refinement stage is to co-refine adjacent segments. This involves modifying the set of segments S in order to reduce the total dissimilarity score D_S, without changing the number of segments $|S|$ or the total number of aligned fragments $|A| \equiv \sum_{k=1}^{|S|} m_k$. Note that, unlike with rigid segment refinement, the segment lengths m_k are allowed to change during refinement.

In segment edge refinement, elements are swapped between adjacent segments s_k and s_{k+1} with the intention of co-minimising their scores, that is:

$$D_{s_k} + D_{s_{k+1}} \to \min \qquad\qquad k = 1 \ldots |S| - 1 \tag{2.28}$$

Aligned segments are sometimes too long in the initial alignment, which causes the edges of segments to be poor-scoring, and hinders attempts at rigid segment refinement. Edge refinement allows such segments to be shortened (if favourable to do so), thus complementing the rigid segment refinement stage, allowing subsequent refinement iterations to be more effective. In concert with this, edge refinement also allows segments that are too short to be lengthened where appropriate, aiding the construction of longer aligned segments. In practice, this procedure allows the evolution and destruction of segments that were initially poorly aligned (whether wholly or in part), thus improving the overall score and enabling subsequent refinement iterations.

In implementation, point mutations (additions and deletions) are allowed to occur at the termini of the segments in order to improve scores, without changing the total number of alignments in the segments ($m_k + m_{k+1} = \text{const}$). Therefore, for any adjacent segment pair s_k and s_{k+1}, there are two potential operations that may occur:

1. Remove a fragment-pair from the end of s_k, and add a fragment-pair to the start of s_{k+1};
2. Remove a fragment-pair from the start of s_{k+1}, and add a fragment-pair to the end of s_k.

Combining the above two operations with the objective of minimising the D_{s_k}, we arrive at the employed segment edge refinement procedure, which is described by Algorithm 2. Unlike in rigid segment refinement, segments will never be able to be combined during edge refinement. However, it is possible for segments to disappear, due to having zero length ($m_k = 0$). In this situation, the segment is removed, and subsequent segments re-indexed.

In implementation, we only allow one operation on each segment-pair (s_k, s_{k+1}) per iteration of edge refinement. Specifically, if a point mutation is applied to (s_k, s_{k+1}), then the possibility of applying further mutations to that segment-pair is not considered during the same refinement iteration; the focus moves instantly to the pair (s_{k+1}, s_{k+2}). By doing this, we enforce a relaxed limitation on the number of mutations that can occur during one refinement iteration, insuring that segments cannot completely metamorphose in a single refinement step. Rather, we allow rigid segment refinement to be executed prior to further iterations of edge refinement, resulting in a more

reliable process overall. This methodology allows the two techniques to better complement each other, allowing them to work in unison to more effectively achieve the objective, hopefully finding a better local minimum of D_S.

Algorithm 2 Segment Edge Refinement

for $k = 1 \ldots |S| - 1$ **do**
  BACKWARD $\leftarrow$ **false**
  FORWARD $\leftarrow$ **false**
  if $\mathbf{D}_{i_{k+1}-1, j_{k+1}-1} < \mathbf{D}_{i_k+m_k-1, j_k+m_k-1}$ **then**
    BACKWARD $\leftarrow$ **true**
  end if
  if $\mathbf{D}_{i_k+m_k, j_k+m_k} < \mathbf{D}_{i_{k+1}, j_{k+1}}$ **then**
    FORWARD $\leftarrow$ **true**
  end if
  if BACKWARD = **true and** FORWARD = **true then**
    if $\mathbf{D}_{i_{k+1}-1, j_{k+1}-1} < \mathbf{D}_{i_k+m_k, j_k+m_k}$ **then**
      FORWARD $\leftarrow$ **false**
    else
      BACKWARD $\leftarrow$ **false**
    end if
  end if
  if BACKWARD = **true then**
    $m_k \leftarrow m_k - 1$
    $s_{k+1} \leftarrow (i_{k+1} - 1, j_{k+1} - 1, m_k + 1)$
    if $m_k = 0$ **then**
      for $x = k \ldots |S| - 1$ **do**
        $s_x \leftarrow s_{x+1}$
      end for
      $s_{|S|} \leftarrow$ **null**
    end if
  else if FORWARD = **true then**
    $m_k \leftarrow m_k + 1$
    $s_{k+1} \leftarrow (i_{k+1} + 1, j_{k+1} + 1, m_k - 1)$
    if $m_{k+1} = 0$ **then**
      for $x = k + 1 \ldots |S| - 1$ **do**
        $s_x \leftarrow s_{x+1}$
      end for
      $s_{|S|} \leftarrow$ **null**
    end if
  end if
end for
return S

Segment Maximisation

Once iterations of rigid segment and edge refinement no longer improve the alignment, the final stage of segment refinement is to maximise the length of the segments. This process ensures that we sufficiently satisfy our objective of making the alignment as long as possible.

In implementation, we detect empty gaps in the alignment, and extend the appropriate segments in a favourable manner. Specifically, upon the identification of a gap between adjacent segments s_k and s_{k+1}, we determine whether it is more favourable to extend segment s_k forward, or to extend segment s_{k+1} backward. The terminal segments are also extended if possible.

This process is very similar to that employed by segment edge refinement, although here we do not reduce the size of the segment not being extended. As with rigid segment refinement, it is possible for segments to become combinable during segment maximisation, in which case segments are spliced as appropriate.

Similarly to edge refinement, we only allow one operation on each segment-pair (s_k, s_{k+1}) per iteration of segment maximisation. Specifically, if an extension is applied to one segment when considering the pair (s_k, s_{k+1}), then the focus moves instantly to the pair (s_{k+1}, s_{k+2}). This allows the previous two refinement stages to be executed prior to further iterations of segment maximisation. Despite the added computational expense, this allows the alignment to achieve a more favourable conformation than if all segments were maximised by one iteration of the segment maximisation procedure, particularly for less similar chain-pairs. This allows the potential identification of better-scoring local minima, resulting in a more effective process overall. The implemented algorithm for segment maximisation is described by Algorithm 3.

It should be noted that adding more fragment-pairs to the alignment will increase the overall score D_S, contrary to our fundamental objective of minimising D_S. Furthermore, at this stage in the process, it is expected for the mean score to increase as more fragment-pairs are added. This is because the fragment-pairs added at this stage would most likely be badly aligned and poor-scoring in comparison with fragment-pairs already included in the alignment. This would suggest that adding more fragment-pairs to the alignment at this stage is counter-productive to our original intentions. However, this is not the case; by maximising the length of the alignment, no matter how poor-scoring the aligned fragment-pairs, we ensure that the final global score may be sensibly compared with the global scores achieved from the alignment of the target structure with other (potentially completely different) proteins/chains. Otherwise, the resulting global alignment dissimilarity score would be meaningless for our purposes; we are not interested in identifying sufficiently similar substructures (at this stage), and rather want to develop a score for the overall agreement of the two chains in their entirety. Ultimately, one future ideal would be to achieve a semimetric (Wilson, 1931) for describing a distance between protein chains in net local conformation space, not to achieve a distance between substructures.

Suppose we implemented the contrary approach, and applied a cutoff on the alignment score. In

Algorithm 3 Segment Maximisation

```
for k = 1 ... |S| − 1 do
  if i_k + m_k < i_{k+1} and j_k + m_k < j_{k+1} then
    if D_{i_k+m_k, j_k+m_k} < D_{i_{k+1}−1, j_{k+1}−1} then
      m_k ← m_k + 1
    else
      s_{k+1} ← (i_{k+1} − 1, j_{k+1} − 1, m_{k+1} + 1)
    end if
  end if
end for
if i_1 > 0 and j_1 > 0 then
  s_1 ← (i_1 − 1, j_1 − 1, m_1 + 1)
end if
if i_{|S|} + m_{|S|} ≤ N_1 and j_{|S|} + m_{|S|} ≤ N_2 then
  m_{|S|} ← m_{|S|} + 1
end if
for k = 1 ... |S| − 1 do
  if i_k + m_k = i_{k+1} and j_k + m_k = j_{k+1} then
    m_k ← m_k + m_{k+1}
    for x = k + 1 ... |S| − 1 do
      s_x ← s_{x+1}
    end for
    s_{|S|} ← NULL
    k ← k − 1
  end if
end for
return S
```

this case, all aligned fragments would score well, regardless of the dissimilarity of the two structures – note that aligned helices always score well, no matter how globally dissimilar structures the chain-pairs. This would mean that the ordering of the global dissimilarity score would not necessarily imply a sensible ordering for the identification of protein dissimilarity (assuming that the global score is the average of the alignment scores). In this case, other factors, such as the proportion of aligned residues, would have to be taken into account in order to reasonably distinguish between similar and dissimilar structures. However, the nature of such an approach would inherently lead to ambiguity in defining a scalar measure for dissimilarity.

In *ProSMART*, we maximise the alignment length in order to achieve a global score that suitably penalises any regions of dissimilarity identified between a chain-pair. A score cutoff may be subsequently applied in order to only identify regions of structure that are considered to be similar. Consequently, by maximising the alignment length at this stage, we are able to provide both functionalities.

2.2.4 Residue-Based Alignment Optimisation

When further iterations of segment refinement fail to find a better-scoring alignment, the refinement process halts, having found the best achieveable fragment alignment. However, although this alignment approximates the best fragment alignment, it may not necessarily imply a valid residue alignment. Specifically, it may not satisfy the required condition of an implicit one-to-one correspondence between residues. Specifically, two cases can occur where a valid one-to-one fragment alignment could imply residue clashes:

1. at the edges of fragment segments, where the edges do not correspond to gaps due to deletions or missing residues. This may occur even for very similar structures, when there are residue insertions;

2. within fragment segments, where there are missing residues or deletions, yet consecutive fragments are aligned nevertheless. This situation may theoretically occur, since $f_i + 1 \neq f_{i+1}$ for some i whenever residues are missing or invalid (see Equation (2.6)).

In order to achieve an alignment of fragments that implies a suitable residue-residue correspondence between the two structures, we now filter the alignment in order to remove any implied residue clashes. The fragment alignment is then further optimised in order to improve the overall alignment, whilst at every step maintaining the required residue correspondence.

It should be acknowledged that it would have been possible to ensure a one-to-one residue correspondence after the dynamic programming stage, upon identification of the initial fragment alignment. The validity of the residue alignment could then have been maintained throughout the subsequent segment refinement stages. However, if this approach was taken, the refinement algorithms would have had to include additional checks, at every step, in order to maintain a valid residue alignment throughout the segment refinement stages. This would have dramatically increased computation time, past what is considered acceptable (as indicated by preliminary experiments). Furthermore, the heterogeneity induced by potential impurities (which are often found in PDB files) would greatly increase the complexity, and thus computational expense, of the procedures. In particular, such impurities (e.g. missing or invalid residues) would hinder the fragment alignment process by creating obstacles that would greatly complicate the score function, significantly impeding the minimisation of D_A. Consequently, in practice, this would result in a worse overall alignment than can be achieved using the implemented method, whereby a residue-residue correspondence is not required until after the segment refinement process.

One may also question why it is necessary for the fragment alignment to directly imply a one-to-one residue correspondence. Indeed, it would be possible to deduce a residue alignment, even if some aligned fragments clashed by indicating a many-to-one (or many-to-many) correspondence between residues. This could be achieved by aligning residues using the best-scoring alignment implied by the fragment alignment, on a per-residue basis. This would bypass the need for alignment filtering

and further refinement. However, the assignment of residue correspondences based on multiple differently-aligned fragment-pairs, we would effectively be willing to accept for a structural region from one protein to be aligned to multiple consecutive regions in another chain. It is argued that if we align a particular residue-pair based on one fragment correspondence, and also accept the alignment of another residue-pair based on a contradictory fragment correspondence, then the alignment methodology would lose physical meaning. Consequently, such an approach would violate the fundamental concept of finding a unique correspondence between the two structures, since the underlying alignment would be based on a logical, abstract, many-to-many residue correspondence. Whilst this may be desirable for some applications, we believe that it does not make sense for the alignment of protein chains. Rather, we prefer to achieve a residue alignment that is based on the optimisation of a fully non-contradictory fragment alignment.

Furthermore, by removing clashing fragments, we expose opportunities for the further optimisation of the alignment. In particular, this allows the lengthening of well-defined fragment segments, thus achieving an overall more sensible and meaningful alignment. This results in having fewer, longer, better-scoring fragment segments, at the potential expense of the (intermediate) minimisation of D_A. Were this stage not performed, we could be left with a greater number of smaller segments. Like with many other concepts in the field of structural biology (e.g. secondary structure, and even the amino acid chain), structural fragments are merely tools with which to achieve an aim. Consequently, when minimising the overall dissimilarity score, it is important not to forget that we intend the overall alignment to be meaningful in a human-readable context.

After the alignment is filtered in order to remove clashing fragments, a one-to-one residue correspondence is maintained throughout all subsequent optimisation stages. Specifically, there are two optimisation stages: pairwise optimisation of aligned fragment-pairs, and alignment maximisation. Each discontinuity or gap in the fragment alignment is only considered once per iteration. This ensures that different results would not be achieved if the sequence ordering was reversed, thus reducing the inherent bias towards the better-refinement of the first half of the chain. These final optimisation stages are iterated (see Figure 4) until no further improvement can be made, indicating that the best achievable minimum has been found.

Clash Removal

Since the adjacency of fragments does not necessarily imply the adjacency of residues (see Equation (2.6)), it is no longer convenient to use segment notation. Rather, we consider each individual aligned fragment-pair $a \in A$ separately. We denote the residue-indexes of the fragments $i \in [1, |F_1|]$ in the first chain by $f_i^1 \in F_1$, and those of the fragments $j \in [1, |F_2|]$ in the second chain by $f_j^2 \in F_2$.

In the detection of fragment clashes, we check that adjacent alignments $a_k = (i, j)$ and $a_{k+1} = (x, y)$ are compatible, in that they do not imply a many-to-one correspondence between residues. Due to the nature of the residue reindexing (Equation (2.2)) and the subsequent definition of

Algorithm 4 Clash Removal

```
for k = 1 ... |A| − 1 do
  (i, j) ← a_k
  (x, y) ← a_{k+1}
  if f^1_x < f^1_i + n or f^2_y < f^2_j + n then
    if f^1_x ≠ f^1_i + 1 or f^2_y ≠ f^2_j + 1 then
      if D_{a_k} ≤ D_{a_{k+1}} then
        for x = k + 1 ... |A| − 1 do
          a_x ← a_{x+1}
        end for
        a_{|A|} ← null
        k ← k + 1
      else
        for x = k ... |A| − 1 do
          a_x ← a_{x+1}
        end for
        a_{|A|} ← null
      end if
    end if
  end if
end for
return A
```

structural fragment indexing (Equation (2.3)), it is not sufficient to confirm that $(x, y) = (i{+}1, j{+}1)$. Rather, we must confirm the stronger condition that their start-residue indexes are adjacent:

$$(f_x^1, f_y^2) = (f_i^1 + 1, f_j^2 + 1). \tag{2.29}$$

An aligned fragment-pair's satisfaction of this condition would result in residues $2, \ldots, n$ from both fragments in alignment a_k being equal to residues $1, \ldots, n-1$ from both fragments in alignment a_{k+1}, confirming compatibility, as required.

If Equation (2.29) is not satisfied, it could also be that the adjacent alignments are sufficiently separated so as to not imply a clash. When comparing similar chain-pairs, this situation often occurs where there is a deletion or missing region common to both structures. Consequently, adjacent alignments are not allowed to overlap:

$$f_x^1 \geq f_i^1 + n \qquad \text{and} \qquad f_y^2 \geq f_j^2 + n. \tag{2.30}$$

The satisfaction of either of conditions (2.29) or (2.30) would indicate a valid compatible alignment-pair. If neither of these conditions are satisfied, then there is a clash, and one of the two fragment-pairs must be removed from the alignment. Specifically, the fragment-pair with the least favourable Procrustes score (D_a) is removed. This is repeated for all adjacent fragment-pairs a_k and a_{k+1} in the alignment, until there are no more clashes. This procedure may be described by Algorithm (4).

Final Optimisation Stages

The removal of less-favourably aligned fragment-pairs during the alignment filtering stage shortens the overall alignment. This results in the alignment being simplified in favour of seemingly well-aligned regions, thus allowing the escape from local minima in the overall score function, enabling the potential for finding a better-scoring local minimum. Consequently, at this stage, we allow the alignment to be improved and extended, in accordance with our alignment criteria. It is intended for this process to result in a more favourable alignment when compared with the alignment prior to filtering. Further to simply enabling a valid (and potentially better-scoring) alignment, this process results in one that is more meaningful in a human-readable context. The optimisation process comprises two parts:

- *Fragment Pair Optimisation* – Searches for more favourable alignments by applying permutations to adjacent aligned fragment-pairs;
- *Alignment Maximisation* – alignment is lengthened where possible, in order to satisfy the maximisation criteria.

It should be noted that, whilst conceptually similar to the segment refinement processes, the algorithms employed here incur comparatively more computational expense per refinement iteration. This is due to working with elements of A, as opposed to elements of S; note that $|A| \gg |S|$ in general, meaning that many more element-pairs must be considered per iteration. Due to the way in which the residues were reindexed, it becomes easy to maintain a one-to-one residue correspondence throughout these fine-tuning stages; this would not be so trivial were we to attempt to refine the segments in S, rather than the fragment-pairs in A, at this stage.

In fragment pair optimisation, adjacently aligned fragment-pairs are co-refined subject to the continued maintenance of alignment validity. Specifically, we keep the total number of aligned fragments $|A|$ constant, and swap elements between adjacent fragment segment edges, as with the segment edge refinement process. If alignments $a_k = (i, j)$ and $a_{k+1} = (x, y)$ imply a discontinuity in the alignment ($x \neq i + 1$ or $y \neq j + 1$), then we adjust the fragment indexes of one of these two alignments in order to minimise their total score:

$$\mathbf{D}_{a_k} + \mathbf{D}_{a_{k+1}} \to \min \tag{2.31}$$

subject to the condition that they must be in agreement with the surrounding alignments. In particular, we must ensure that any heterogeneity in residue indexing does not impede the adjustment of fragment indexes. Specifically, we would have to check that $f_i^1 + 1 = f_{i+1}^1$ and $f_j^2 + 1 = f_{j+1}^2$ in order to extend fragment-pair $a_k = (i, j)$ forward. Again, only one operation is allowed on each discontinuity per optimisation cycle. This procedure may be described by Algorithm (5).

The final optimisation stage is alignment maximisation. As with segment-based refinement, this involves lengthening the alignment where possible. If a discontinuity is detected in the fragment

Algorithm 5 Fragment Pair Optimisation

```
for k = 1 ... |A| − 1 do
  (i, j) ← a_k
  (x, y) ← a_{k+1}
  FORWARD ← false
  BACKWARD ← false
  if x ≠ i + 1 or y ≠ j + 1 then
    if f^1_i + 1 = f^1_{i+1} and f^2_j + 1 = f^2_{j+1} then
      if D_{(i+1,j+1)} < D_{(x,y)} then
        FORWARD ← true
      end if
    end if
    if f^1_x − 1 = f^1_{x−1} and f^2_y − 1 = f^2_{y−1} then
      if D_{(x−1,y−1)} < D_{(i,j)} then
        BACKWARD ← true
      end if
    end if
    if FORWARD = true and BACKWARD = true then
      if D_{(i+1,j+1)} < D_{(x−1,y−1)} then
        BACKWARD ← false
      else
        FORWARD ← false
      end if
    end if
    if FORWARD = true then
      a_{k+1} ← (i + 1, j + 1)
      k ← k + 1
    else if BACKWARD = true then
      a_k ← (x − 1, y − 1)
    end if
  end if
end for
return A
```

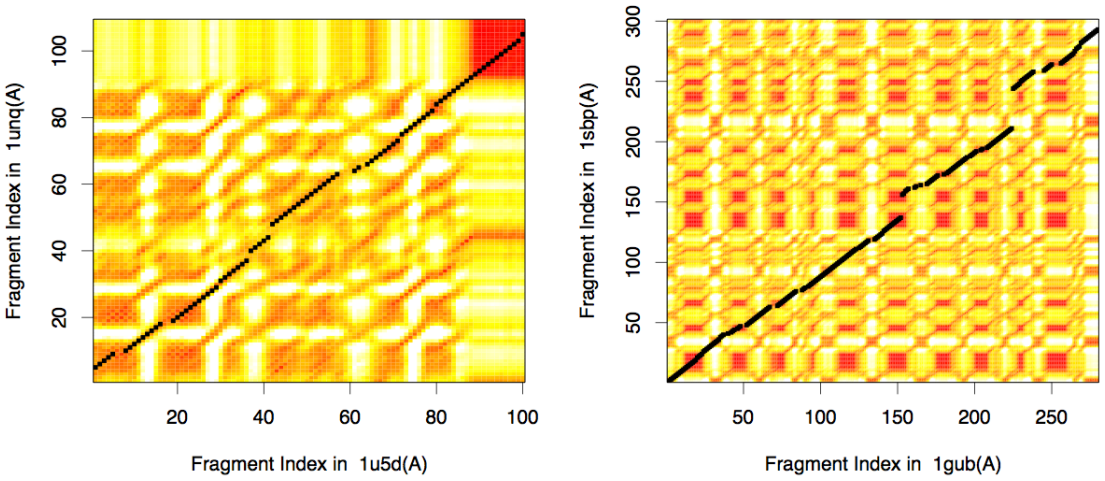
alignment ($x \neq i+1$ or $y \neq j+1$ where $a_k = (i,j)$ and $a_{k+1} = (x,y)$), then a new fragment-pair is added to the alignment, providing there exists a suitable fragment-pair that does not cause a clash. Candidate fragment-pairs are $(i+1, j+1)$ and $(x-1, y-1)$; the most favourable of the two is selected if possible. This procedure is described by Algorithm (6).

The final alignments resulting from two example comparisons are shown in Figure 6 (compare with Figures 3 and 5). The similar structures (left) are sensibly aligned everywhere. As can be seen in Figure 7a, The core is aligned well, and surface insertions/deletions are handled sensibly. Purposefully, the alignment is maximised in these surface regions, even when local structure appears unconserved. The key point is that the conserved regions dominate the alignment. Due to conservation of global conformation and topology, superposition is meaningful in this case. In contrast,

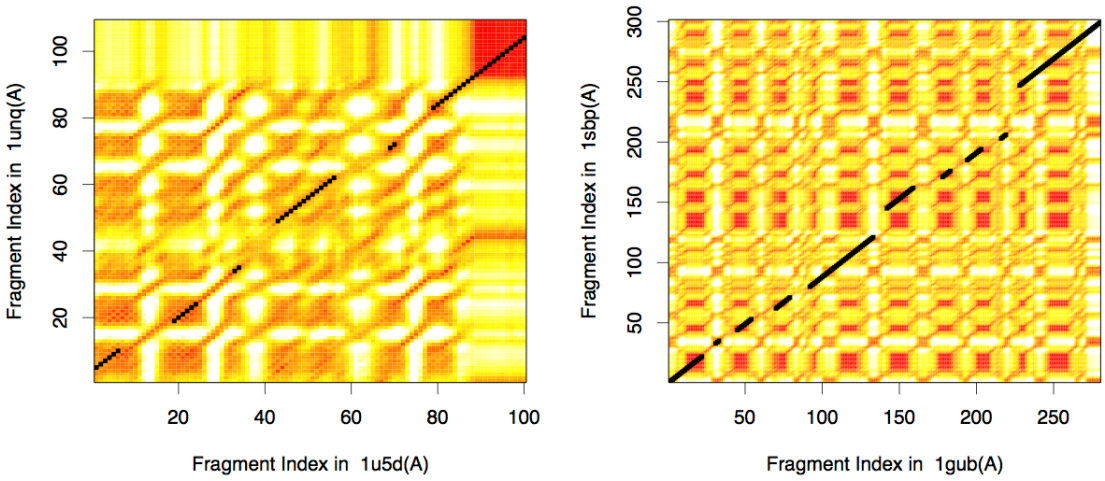
the unrelated structures (right in Figure 6, and also Figures 7b and 7c) appear randomly aligned, as expected. Nevertheless, a maximal alignment is achieved, since the employed methodology does not distinguish between similarity and dissimilarity, a fundamental feature of this approach.

Algorithm 6 Alignment Maximisation

for $k = 1 \ldots |A| - 1$ **do**
  $(i, j) \leftarrow a_k$
  $(x, y) \leftarrow a_{k+1}$
  FORWARD $\leftarrow$ **false**
  BACKWARD $\leftarrow$ **false**
  if $x \neq i + 1$ **and** $y \neq j + 1$ **then**
    if $f_i^1 + 1 = f_{i+1}^1$ **and** $f_j^2 + 1 = f_{j+1}^2$ **then**
      if $f_x^1 - f_i^1 = f_y^2 - f_j^2$ **or** $f_{i+1}^1 + n \leq f_x^1$ **and** $f_{j+1}^2 + n \leq f_y^2$ **then**
        FORWARD $\leftarrow$ **true**
      end if
    end if
    if $f_x^1 - 1 = f_{x-1}^1$ **and** $f_y^2 - 1 = f_{y-1}^2$ **and** $f_i^1 + n \leq f_{x-1}^1$ **and** $f_j^2 + n \leq f_{y-1}^2$ **then**
      BACKWARD $\leftarrow$ **true**
    end if
    if FORWARD = **true and** BACKWARD = **true then**
      if $D_{(i+1,j+1)} < D_{(x-1,y-1)}$ **then**
        BACKWARD $\leftarrow$ **false**
      else
        FORWARD $\leftarrow$ **false**
      end if
    end if
    if FORWARD = **true then**
      $A \leftarrow$Insert$(i + 1, j + 1)$
      $k \leftarrow k + 1$
    else if BACKWARD = **true then**
      $A \leftarrow$Insert$(x - 1, y - 1)$
      $k \leftarrow k + 1$
    end if
  end if
end for
$(i, j) \leftarrow a_1$
if $i > 1$ **and** $j > 1$ **and** $f_i^1 - 1 = f_{i-1}^1$ **and** $f_j^2 - 1 = f_{j-1}^2$ **then**
  $A \leftarrow$Insert$(i - 1, j - 1)$
end if
$(i, j) \leftarrow a_{|A|}$
if $i < N_1$ **and** $j < N_2$ **and** $f_i^1 + 1 = f_{i+1}^1$ **and** $f_j^2 + 1 = f_{j+1}^2$ **then**
  $A \leftarrow$Insert$(i + 1, j + 1)$
end if
return A

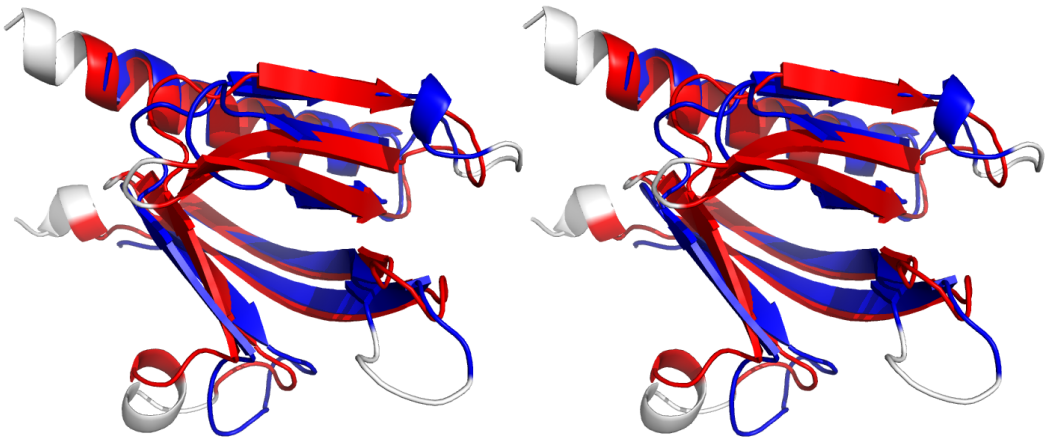


(a) Alignment after segment-based refinement.

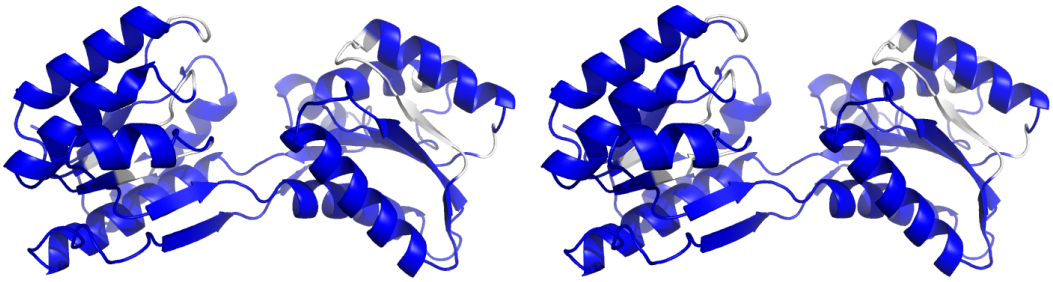


(b) Final alignment after clash removal and final optimisation stages.

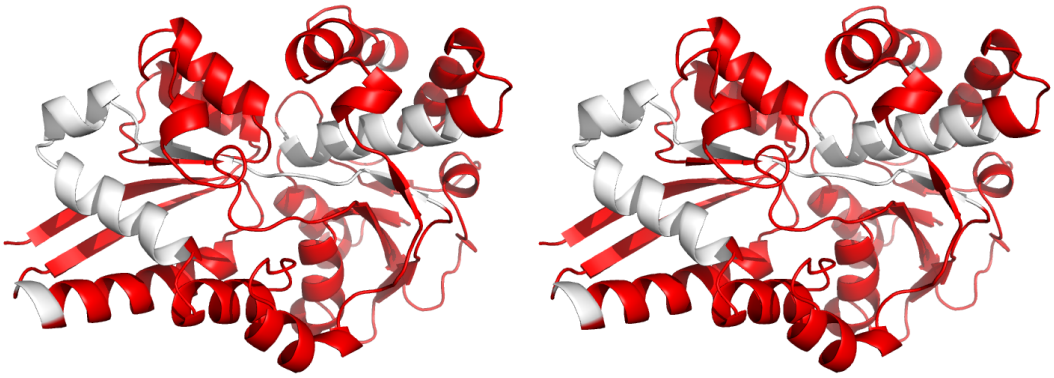
Figure 6: Example fragment dissimilarity matrices, for structurally similar chains 1unq(A) and 1u5d(A) (left images), and dissimilar chains 1gub(A) and 1sbp(A) (right images), for $n = 9$. Matrices are coloured according to Procrustes distance; red indicates low distance $\mathbf{D}_{ij} \approx 0$, yellow intermediate, and white high distance $\mathbf{D}_{ij} \geq 4$. Black dots correspond to aligned fragment-pairs, specifically the alignment after segment-based refinement (a), and the final alignment after clash removal and final optimisation (b).



(a) Stereo superposition of 1unq(A) and 1u5d(A).



(b) Stereo view of 1gub(A).



(c) Stereo view of 1sbp(A).

Figure 7: Depictions of the final alignments, with unaligned residues coloured white. In (a), the structures 1unq(A) and 1u5d(A) are shown superposed in stereo, with aligned residues coloured red for 1unq(A) and blue for 1u5d(A). The structures 1gub(A) and 1sbp(A) are shown in (b) and (c), with aligned residues coloured blue for 1gub(A), and red for 1sbp(A).

2.3 Rigid Substructure Identification

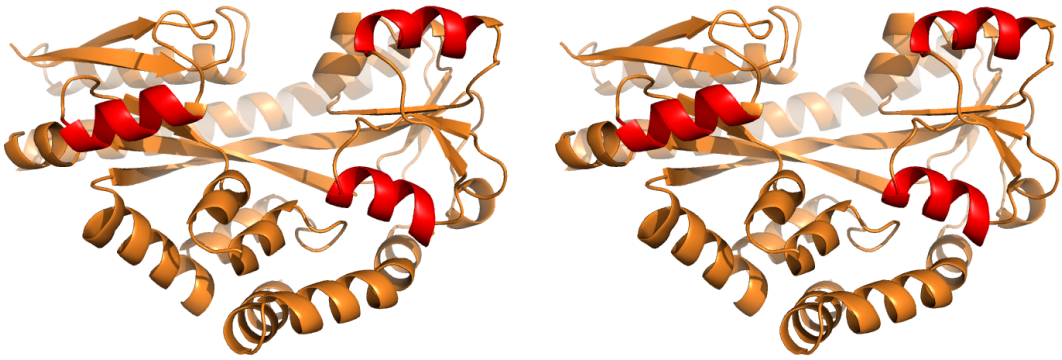
Rigid substructure identification is a problem that has been previously addressed using many approaches. For many other comparison methods, this problem is considered analogous to that of structural alignment. Often, the objective is to identify the maximal list of residue-pairs that, when superposed, result in a measure of dissimilarity (commonly RMSD) below some threshold. In contrast, 'local' methods, such as that employed here, do not optimise global agreement, and thus do not identify rigid substructures. The intermediate 'flexible' approaches search for piecewise rigidity, effectively taking a global approach but allowing the identification, alignment, and superposition of multiple rigid regions rather than producing a single result for the whole chain.

Being invariant to global conformation, the alignment resulting from our method is not concerned with implication of conserved larger rigid substructures. Therefore, there is no reason for the global superposition implied by the resultant alignment to be sensible, let alone intuitively ideal (i.e. seem visually correct). In fact, an intuitively ideal superposition is very rarely achieved based on a global alignment, especially if the alignment is maximised. This is as intended, since superposition was not the primary objective. One possible improvement to a global superposition would be to weight residue-pairs according to their alignment score(s), but that would only address issues regarding badly aligned regions. Further to the influence of dissimilar regions and structural flexibility, one major issue is global conformation. Conformational changes may be dramatic, such as those caused by inter-domain movement (e.g. hinging motions), or more subtle intra-domain changes that occur gradually through space. Consequently, the superposition of protein chains does not always make sense. If there are multiple rigid structural units shared between two similar protein chains, then the optimal superimposition of the two chains will inherently be flawed. We aim to utilise properties of our local approach in order to address these issues.

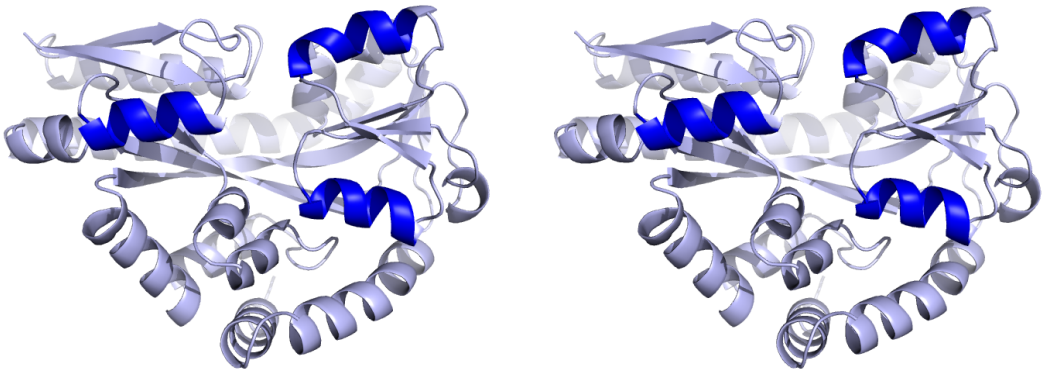
Since the ability to identify and/or superpose substructures well is a desirable and important secondary feature, a method for the identification of conserved rigid substructures, given a prior alignment, is presented. The proposed approach is complementary to our conformation-invariant alignment method, but is also general enough to be applicable to any alignment that can be expressed in terms of n-residue structural fragments. Unlike our alignment approach, this method does not require maintenance of sequence order, and is therefore applicable to a wider class of problems.

2.3.1 Agreement of Aligned Fragment-Pairs

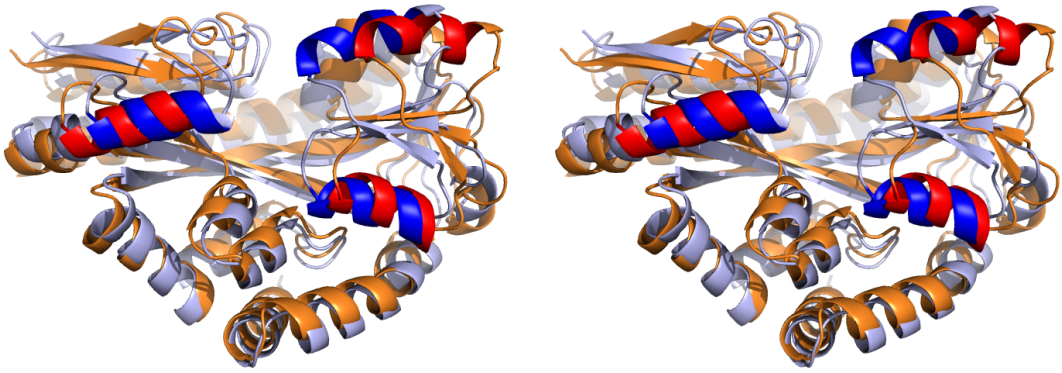
The employed method of rigid substructure identification involves identifying clusters of aligned fragment-pairs that belong to the same approximate coordinate frame. Such clusters may correspond to rigid structural units, in which case identified clusters may be used to superpose each identified common substructure. Even in cases where there is only one cluster identified, this information, which pertains to the exclusion of flexible regions, can be used to achieve a better global



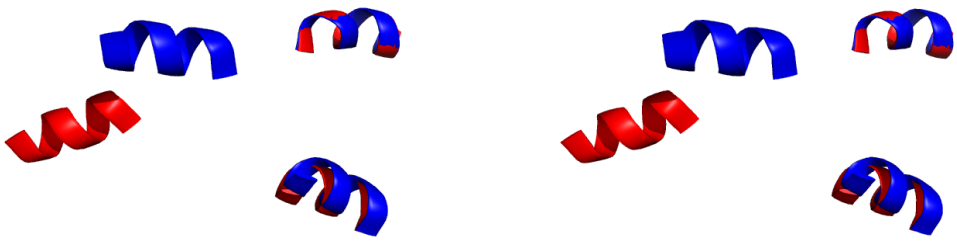
(a) Stereo view of 2cex(A).



(b) Stereo view of 2cex(B).



(c) Stereo global superposition of 2cex(A) and 2cex(B).



(d) Stereo view of fragments from 2cex(A) and 2cex(B).

Figure 8: Example of the superposition of identical structures in different conformations. Chains A (a) and B (b) from PDB file 2cex are shown, along with their global superposition in stereo (c). Three fragment-pairs are identified in bright red/blue. These are placed in a common coordinate frame such that the two fragment-pairs belonging to the same substructure are superposed (d).

superposition than that which can achieved by the default method, whereby all aligned residues are simply superposed.

For illustration, we shall consider the case of domain movement in sequence-identical structures; the method should directly generalise to more challenging scenarios. Figure 8 shows such a chain-pair, where the global superposition is not aesthetic. Three fragment-pairs are identified. Although these do not superpose well under the global superposition, it is evident that two of the fragment-pairs can, whilst the third cannot, be superposed in the same coordinate frame. Consequently, we wish to be able to identify that the two superposable fragment-pairs can, and that the third cannot, be considered to belong to the same coordinate frame, and thus belong to the same substructure.

One simple way of quantifying the agreement between pairs of aligned fragment-pairs would be to consider their RMSD after superposition. However, such an approach would produce results with ambiguous interpretation, since even fragment-pairs in the same coordinate frame would have non-zero score. The magnitude of this 'ideal' score would depend on the particular fragment-pairs' internal disagreement. However, we do not want a method that is sensitive to fragment dissimilarity, since we already have such information from the alignment stage. Rather, we are interested in identifying fragment-pairs in common coordinate frames, regardless of their internal conservation.

Specifically, the approach involves consideration of the differences between the transformations required to superpose the aligned fragment-pairs. Transformations comprise two components: translational and rotational. Note that it is not possible to unambiguously combine these two components into a single measure; a suitable distance metric between elements of SE(3) (rigid-body transformations) has not yet been found (Venkataramanujam and Larochelle, 2010). However, approximate solutions do exist, such as left-invariant metrics, which may or may not be suitable for the present purpose. Such methods include the consideration of higher-dimensional rotations after embedding elements of SE(3) into SO(4) (Larochelle et al., 2007). Whilst such an approach may have useful application elsewhere, we deem the combination of these two components in an arbitrary/ambiguous manner to be unsatisfactory for the present application, without the support of empirical evidence.

Therefore, we consider the translational and rotational components separately. If we considered only the translational part, then any rotation of one coordinate frame would result in there always being a position in space around which it would not be possible to distinguish between substructures (e.g. a hinge). By only considering the rotational part, substructures would only be indistinguishable if they differ only by a translation. Tentatively, it may be reasonable to assume that different substructures in protein chains generally have different orientations. Consequently, this latter compromise is considered more reasonable, and thus only the rotational component is utilised at present.

Scoring Pairs of Aligned Fragment-Pairs

Since the coordinate frames of the two compared chains are in arbitrary orientations, we must ensure invariance with respect to the original coordinate frames. For each aligned fragment-pair, we wish to describe the orientational difference between the local coordinate frames of the two structures (see Figure 9 for an example). For each aligned fragment-pair $a_k \in A$, we calculate the matrix required to rotate the fragment from chain 1 onto the corresponding fragment in chain 2, assuming translational invariance, achieving the rotation $\mathbf{R_k}$, similarly to as defined by Equation (2.11) (see §2.1.4 for details). Note that it is not necessary to physically rotate/superpose the fragments.

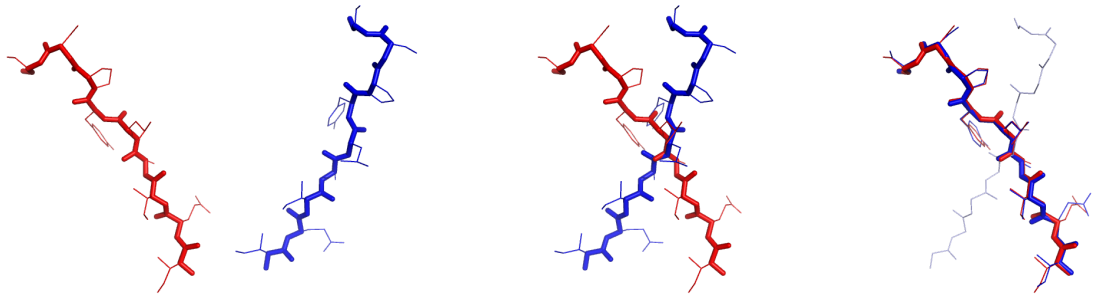


Figure 9: Illustration of the process of calculating inter-chain inter-fragment rotations. The example uses an aligned fragment-pair between chains A (red) and B (blue) of 2cex (residues 36–44). Initially, the two fragments lie in arbitrary coordinate frames (left). Translation-invariance is ensured by translating both fragments to the origin (centre). Finally, the rotation matrix is calculated that rotates the fragment from chain B onto the other fragment (right), resulting in superposed fragments in the same coordinate frame.

Fundamental to the method is the supposition that if any two aligned fragment-pairs i and j are part of a rigid conserved substructure, then they should be superposable, i.e. belong to the same coordinate frame. Consequently, their rotation matrices $\mathbf{R_i}$ and $\mathbf{R_j}$ should be approximately equal. The rotation matrix $\mathbf{R_i^T R_j}$ represents the orientational difference between the two coordinate frames. Note that if $\mathbf{R_i}$ and $\mathbf{R_j}$ are approximately equal then $\mathbf{R_i^T R_j}$ should approximately equal the identity matrix. In order to quantify the magnitude of this difference rotation as a scalar measure, we consider the angle θ_{ij} between the Euler axes of these rotations, in order to achieve the cosine distance (Murray et al., 1994):

$$d_{ij}^{\theta} = 1 - \cos(\theta_{ij}) = \frac{3 - \mathrm{tr}\left(\mathbf{R_i^T R_j}\right)}{2} \tag{2.32}$$

This score represents the dissimilarity of pairs of aligned fragment-pairs. The problem is now to determine which fragment-pairs, if any, belong to a common substructure.

2.3.2 Identification of Initial Substructures

As can be seen in Figure 10, it is possible to visually identify regions placed in different coordinate frames by considering the cosine distance matrix. Red squares are observed on the diagonal, corresponding to rigidly conserved sections of chain. Off-diagonal red rectangles correspond to regions of

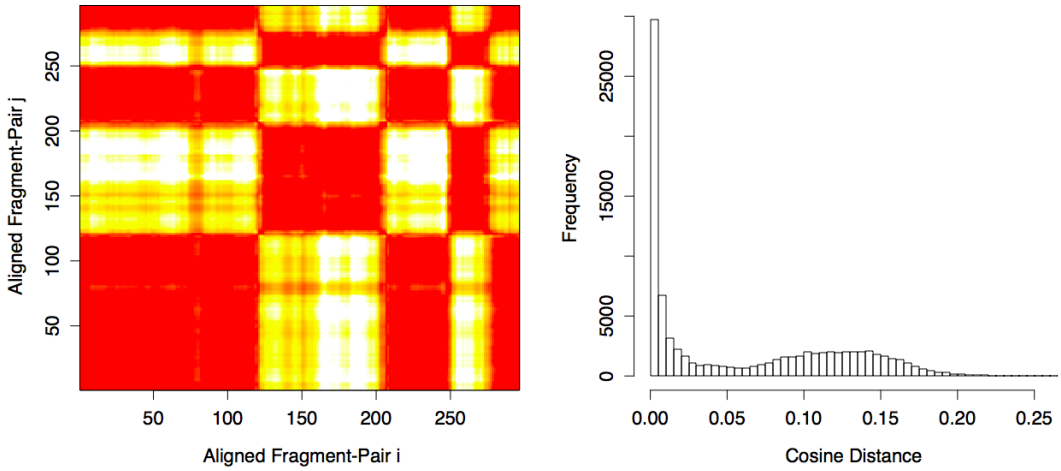


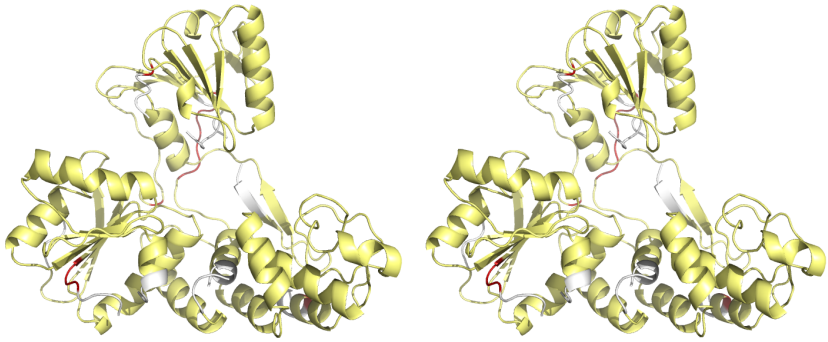
Figure 10: Distribution of cosine distances between pairs of aligned fragment-pairs between 2cex(A) and 2cex(B). Left: Magnitudes of cosine distances are shown in colour; red indicates low distance $d^{\theta} \approx 0$, yellow intermediate, and white high distance $d^{\theta} \geq 0.15$. Right: histogram of cosine distances.

the chain that are disconnected in sequence, but belong to the same coordinate frame. Yellow/white regions correspond to regions that do not belong to the same substructure. Intermediates, such as the region around fragment-pair 80, correspond either to other coordinate frames (e.g. surface loops), or regions that are spatially located between other substructures. There appear to be five distinct regions, corresponding to four coordinate frame transitions, whilst there are only two major coordinate frames. This is indicative of the chain entering and leaving the two substructures multiple times.

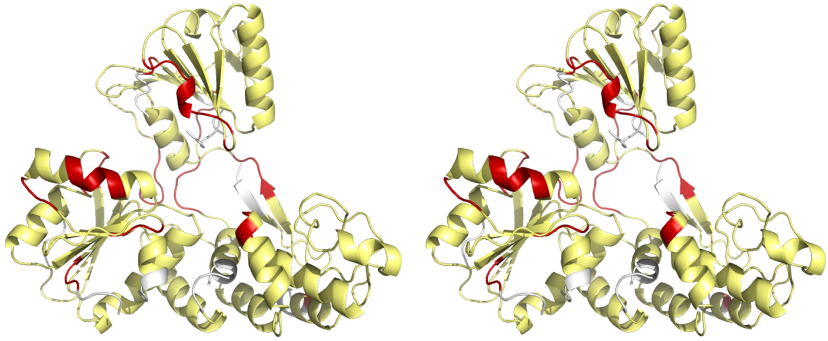
In general, cosine distance matrices may not be quite so interpretable as that in Figure 10. In the case of gradual conformational change, the red (rigid) regions would not be square/rectangular, and substructure borders may be harder to distinguish. Furthermore, transitions between distinct substructures may be less obvious. In order to achieve an automated solution, a form of fragment-pair clustering is used in order to separate orientationally dissimilar regions. Since we are able to extract further useful information from the system, a heuristic approach towards clustering is taken.

Aligned Fragment-Pair Filtering

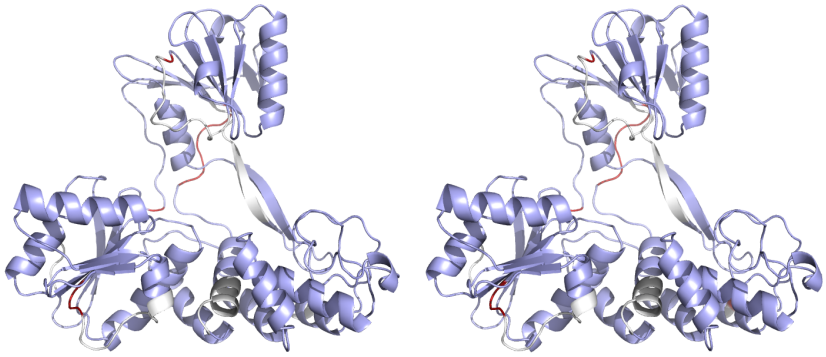
Before attempting to cluster the distances, the list of aligned-fragment pairs is filtered in order to reduce noise, separate clusters and increase reliability of results. Specifically, we remove fragment-pairs with poor Procrustes scores, since it makes sense to only use well-aligned fragments. We also remove those which exhibit a relatively high degree of intra-fragment rotational dissimilarity, since these are likely to be situated between clusters (e.g. hinge fragments) or on the surface, and not superpose well. Removing these aligned fragment-pairs helps to achieve greater cluster separation, thus aiding the initial identification of distinct clusters.



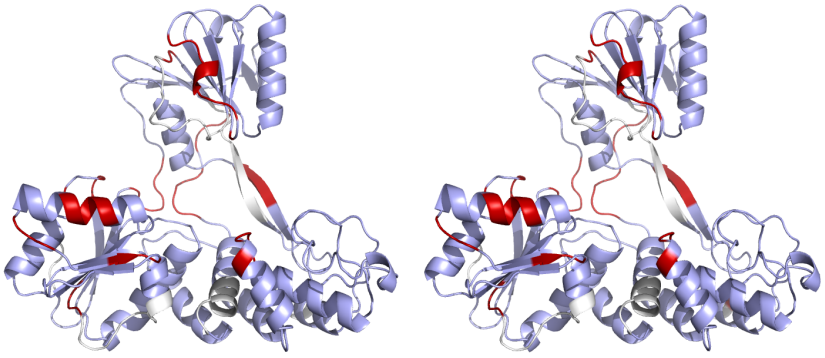
(a) 1a1v(A), identifying residues with poor central score.



(b) 1a1v(A), identifying residues with poor intrafragment rotational dissimilarity score.

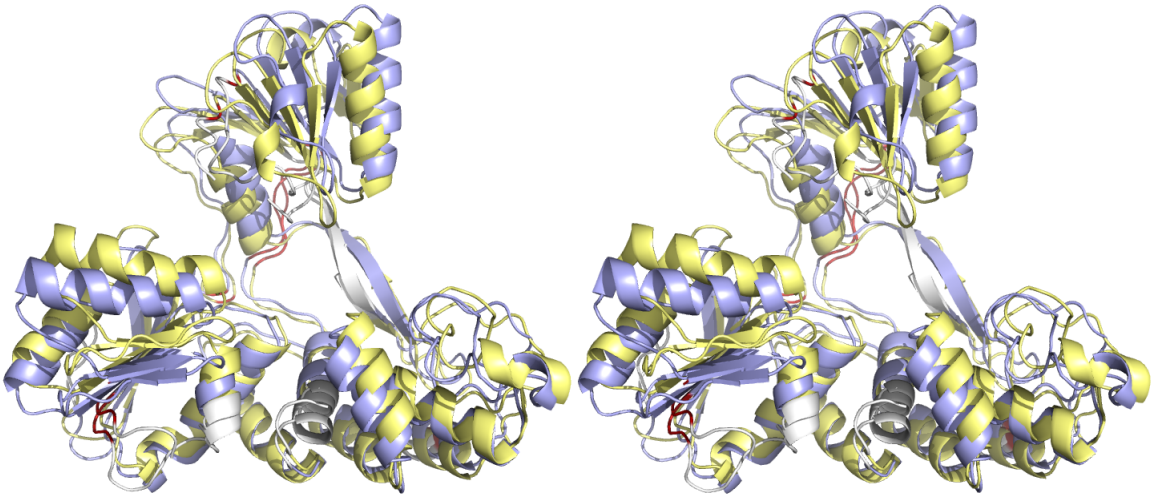


(c) 8ohm(A), identifying residues with poor central score.

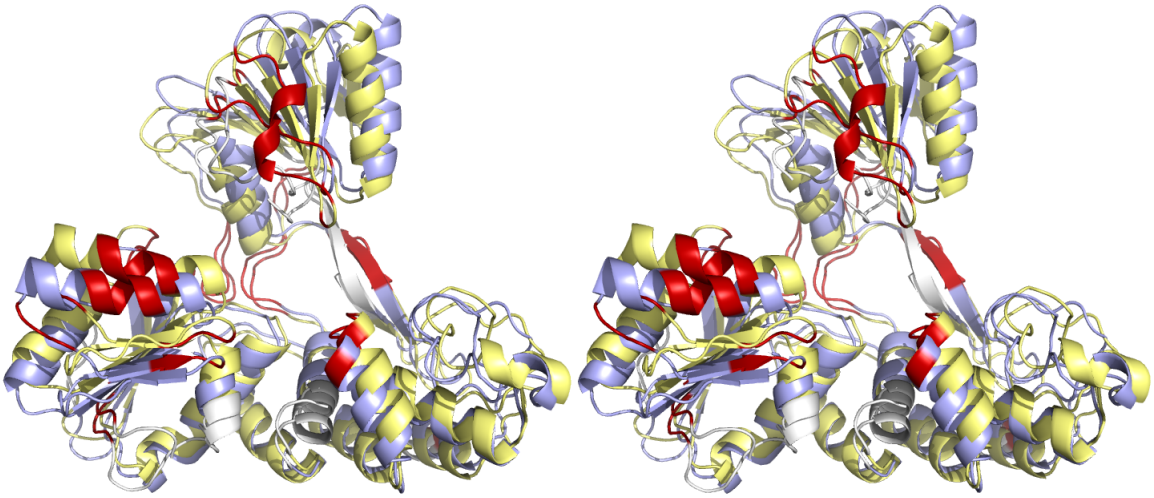


(d) 8ohm(A), identifying residues with poor intrafragment rotational dissimilarity score.

Figure 11: Stereo views of 1a1v(A) (a,b) and 8ohm (c,d), depicting results from their comparison. Aligned fragments are shown in colour, unaligned in white. Only the central residue of a fragment is coloured (e.g. a one-residue alignment gap would cause nine residues to be coloured white, since $n = 9$). In (a) and (c), the (very few) fragments with a relatively poor Procrustes score (> 1Å) have their central residues coloured red. In (b) and (d), fragments with a relatively poor intrafragment rotational dissimilarity score ($> 15°$) have their central residues coloured red.



(a) Stereo superposition of 1a1v(A) and 8ohm(A), identifying residues with a poor central score.



(b) Stereo superposition of 1a1v(A) and 8ohm(A), identifying residues with a poor intrafragment rotational dissimilarity score.

Figure 12: Stereo global superpositions of two sequence-identical chains 1a1v(A) and 8ohm(A). Aligned fragments are shown in colour, unaligned in white. Only the central residue of a fragment is coloured (e.g. a one-residue alignment gap would cause nine residues to be coloured white, since $n = 9$). In (a), the (very few) fragments with a relatively poor Procrustes score (> 1Å) have their central residues coloured red. In (b), fragments with a relatively poor intrafragment rotational dissimilarity score ($> 15°$) have their central residues coloured red.

Figures 11 and 12 show an example of the identification of fragment-pairs with poor Procrustes and intra-fragment rotational dissimilarity scores, for a pair of sequence-identical chains in different conformations. Removing these poor-scoring fragment-pairs results in a reduction of noise in the cosine dissimilarity matrix, as is visually noticeable in Figure 13.

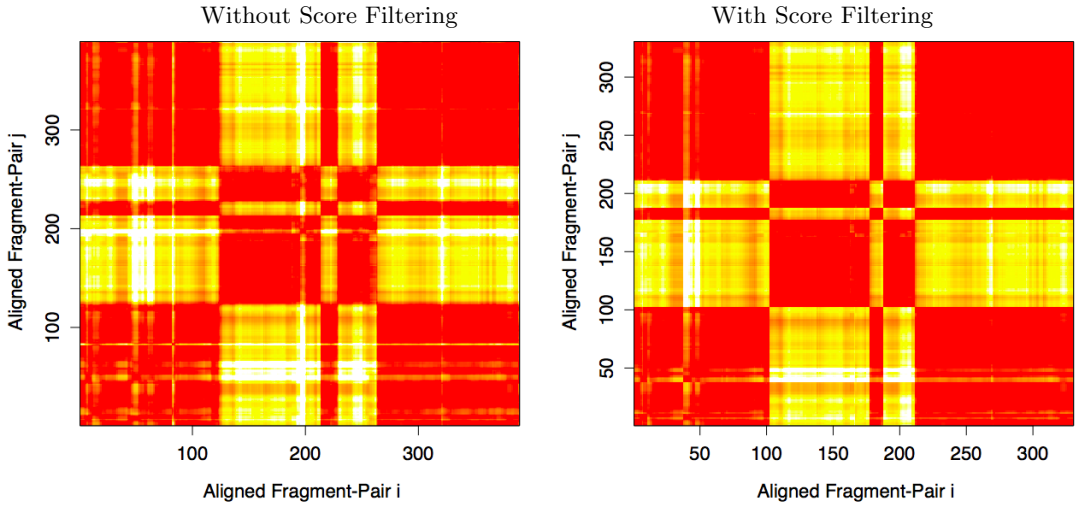


Figure 13: Cosine distance matrices corresponding to the alignment of 1a1v(A) and 8ohm(A) (shown in Figures 11 and 12). The left image depicts the matrix before, and the right image after, filtering with respect to Procrustes score (threshold: 1Å) and intra-fragment rotational dissimilarity score (threshold: 15°). Magnitudes of cosine distances are shown in colour; red indicates low distance $d^{\theta} \approx 0$, yellow intermediate, and white high distance $d^{\theta} \geq 0.15$.

Customised Single Linkage Clustering

After fragment-pair filtering, we wish to identify the distinct clusters that will form the initial definition of any identified substructures. Objectively, we have no preconception regarding the number of clusters, or indeed if there are any clusters, present. The methodology should be applicable to chain-pairs exhibiting:

- No clusters – dissimilar structures, or those without sufficient rigid conservation of chain orientation in aligned regions, given structural resolution (fragment length) and choices of parameters in the clustering algorithm;
- One cluster – e.g. a common core, whereby there is no distinct identifiable global conformational change (given chosen parameter values) excepting noise such as flexible surface loops. Very gradual smooth conformational changes may also be identified as only one cluster;
- Multiple clusters – due to distinct global conformational changes, such as hinging motions.

Given these objectives, we conclude that it would be appropriate for the employed clustering method (Everitt et al., 2011) to be agglomerative, and not partitional. Furthermore, there may be disadvantages to the use of methods such as complete or average linkage clustering, due to the potential for resultant undesirable artefacts; we have no reason to necessarily expect nor enforce clusters to be globular. For purposes of speed, the simple single linkage clustering would be desirable due to not requiring an explicit hierarchy to be identified during the process, nor scores to be iteratively recalculated, as is the case for many other methods. Furthermore, this method would allow

for gradual smooth conformational changes to occur within one cluster. However, single linkage clustering has the disadvantage of the potential for not being able to distinguish between distinct clusters. For example, a smooth conformational transition between two distinct coordinate frames will cause the two clusters to be 'linked', and thus identified as only one cluster.

We use a modified version of single linkage clustering, which utilises knowledge of the specific system. Note that the coordinate frames of consecutive fragment-pairs will likely have very similar orientations, especially if the paired fragments are similar, since the relative conformations of very close fragment-pairs are affected by chemical restraints. Such consecutive fragment-pairs may form undesirable links exacerbating cluster anisotropy, thus inhibiting the identification of distinct clusters. Specifically, it would be reasonable to surmise that adjacent (or overlapping) fragments in regions between two substructures (e.g. hinge regions) would likely be responsible for forging 'links' between the two clusters. In order to reduce this effect, we stipulate that fragments must be sufficiently separated along the chain in order to be allowed to create links. For example, we might specify that fragment-pairs containing more than one shared residue are not allowed to form links; linked fragment-pairs must be separated by at least $(n-2)$ residues. A single linkage clustering score threshold should be chosen appropriately, e.g. $d^{\theta} = 0.002$ ($\approx 3.6°$). The histogram of cosine distances (e.g. as in Figure 10) may be used to aid such decisions. Future work should involve exploration of suitable parameter values.

Note that distant fragments are allowed to form links; fragment-pair proximity is not considered, allowing the detection of rotational conservation for spatially separated fragments. Since there is potential for small clusters to be randomly identified, rather than actually being part of substantial substructures, we allow cluster filtering. Specifically, any clusters comprising fewer than a given number of fragment-pairs (10, by default) are ignored.

It should be acknowledged that this is a very simplistic method, and has the potential to fail to identify separate clusters if there randomly happens to be a smooth string of similarly-oriented fragment-pairs between them. However, due to the initial alignment filtering removing fragment-pairs with high Procrustes and intra-fragment rotational dissimilarity scores, the risk of this occurring is greatly reduced. Importantly, this method is fast (see 3.2.1). There may be limits to the current implementation, due to significant variations in substructures belonging to proteins from different structural classes/families.

2.3.3 Identification of Final Coordinate Frames

Knowledge of the identified clusters is not generally enough to produce a sufficiently good superposition for our purposes. We currently have no control over, or knowledge of, cluster variability, spatial variability, bias, and outliers. Consequently, a superposition arising directly from the implied cluster residues may fail to adequately superpose the cluster core, suffering from a similar effect to the global superposition (although to a much lesser degree). Note that surface fragments

at the edges of a cluster will inherently have a larger influence on the superposition than core fragments at the cluster centre, since magnitude serves as effective weight, contrary to the ideal.

The employed solution involves considering only fragment-pairs near the 'centre' of the cluster, in order to remove those not considered to belong to the orientational core of the substructure. Note that we do not refer to fragment-pairs at the spatial centre of the substructure in Euclidean space, but rather to those at the centre of the cluster, which are considered most consistent with the orientation of the core of the substructure. Consequently, the resultant superposition will not be based on the whole substructure, but rather will be based on some notion of average coordinate frame. This means that, in the event of a substructure that exhibits a gradual conformational change, the extremes of this change will be ignored; the coordinate frame will be chosen to be consistent with the average conformation. This contrasts with traditional approaches, whereby the superposition would be chosen so as to minimise the RMSD between a given set of atoms.

Formulation Using Quaternions

The proposed approach is facilitated by the use of the quaternions (Kuipers, 1999) to represent rotations, rather than representation as an orthogonal matrix. This representation is often preferred for reasons of storage, computational efficiency, and robustness. Note that rotations can also be represented in other forms (e.g. Euler angles, Euler axis/angle notation). A quaternion $\mathbf{q}$ may be written:

$$\mathbf{q} = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} \tag{2.33}$$

To represent a rotation, the coefficients q_i may be achieved directly from a rotation matrix $\mathbf{R} \in SO(3)$ as follows:

$$\begin{aligned} q_1 &= \pm\sqrt{1 + \mathbf{R}_{11} + \mathbf{R}_{22} + \mathbf{R}_{33}} \\ q_2 &= \frac{\mathbf{R}_{32} - \mathbf{R}_{23}}{4q_1} \\ q_3 &= \frac{\mathbf{R}_{13} - \mathbf{R}_{31}}{4q_1} \\ q_4 &= \frac{\mathbf{R}_{21} - \mathbf{R}_{12}}{4q_1} \end{aligned} \tag{2.34}$$

The sign of q_1 may be chosen; the positive will suffice. Note that q_2, q_3, and q_4 have potential to be numerically unstable when the denominator q_1 is close to zero. Due to redundancies in the rotation matrix, there are three other possible formulations of $\mathbf{q}$ using $\mathbf{R}$ (see Diebel, 2006); the one with the greatest denominator is used.

The Euler angle corresponding to a quaternion is dependent only on the scalar part q_1, being given by:

$$\theta = 2\arccos(q_1) \tag{2.35}$$

The quaternion $\mathbf{q}'$ required to transform one unit quaternion onto another is given by:

$$\mathbf{q}' = {\mathbf{q_i}}^{-1}\mathbf{q_j} = {\mathbf{q_i}}^{*}\mathbf{q_j} \tag{2.36}$$

whose scalar part is given by:

$$\mathbf{q'}_1 = \sum_{x=1}^{4} \mathbf{q_i}_x \mathbf{q_j}_x \tag{2.37}$$

Consequently, the cosine distance between two rotations, given by Equation (2.32), may also be expressed in terms of the quaternions corresponding to the two rotations:

$$\begin{aligned} d_{ij}^{\theta} = 1 - \cos(\theta_{ij}) &\equiv 1 - \cos\left(2 \arccos\left(\mathbf{q'}_1\right)\right) \\ &= 2 - 2\left(\sum_{x=1}^{4} \mathbf{q_i}_x \mathbf{q_j}_x\right)^2 \end{aligned} \tag{2.38}$$

using the fact that $\cos(n \arccos(z)) = T_n(z)$, the n^{th} Chebyshev polynomial. This may be more computationally efficient than unnecessary conversion back to rotation matrix notation. However, where required, the rotation matrix corresponding to a quaternion may be calculated:

$$\mathbf{R} = \begin{pmatrix} 1 - 2q_3^2 - 2q_4^2 & 2q_2q_3 - 2q_4q_1 & 2q_2q_4 + 2q_3q_1 \\ 2q_2q_3 + 2q_4q_1 & 1 - 2q_2^2 - 2q_4^2 & 2q_3q_4 - 2q_1q_2 \\ 2q_2q_4 - 2q_3q_1 & 2q_3q_4 + 2q_2q_1 & 1 - 2q_2^2 - 2q_3^2 \end{pmatrix} \tag{2.39}$$

Cluster Tightening

We now wish to further filter the clustered aligned fragment-pairs in order to better represent the 'core' of the cluster, excluding outliers and also the extremes of any gradual conformational change present (as discussed above). Specifically, we identify a rotation representing the centre of a cluster, and remove any fragment-pairs whose cosine distance to the cluster centre is larger than some predefined threshold.

One choice would be to superpose all residues belonging to fragments in the existing cluster, and use the corresponding coordinate frame to calculate the rotation of the cluster centre (this option is also implemented, due to giving different thus complementary results to the method presented here). However, using this method, spatial outliers would have a larger influence on the coordinate frame than those in the substructure core, contrary to our conceptual ideal. Rather, we take an approach that is invariant to the fragments' relative spatial locations.

We assume a fragment alignment A, and a list of clusters $C = \{c\}$, such that $c_i \subseteq A$ for $i = 1 \ldots |C|$, with $c_i \cap c_j = \emptyset$ for $i \neq j$, defined by the result of the single linkage clustering stage described above (only sufficiently large clusters are considered). Let the quaternion corresponding to the aligned fragment-pair $a_k \in c_i$ be denoted by $\mathbf{q_k}$. For each identified cluster, consider the average rotation of aligned fragment-pairs in the cluster. For simplicity, this is approximated/represented using the normalised average quaternion:

$$\hat{\mathbf{q}}_{\mathbf{c_i}} = \frac{\sum_{a_k \in c_i} \mathbf{q_k}}{||\sum_{a_k \in c_i} \mathbf{q_k}||} \tag{2.40}$$

which has elements:

$$\hat{\mathbf{q}}_{\mathbf{c_i}x} = \frac{\sum_{a_k \in c_i} \mathbf{q_k}_x}{\sqrt{\sum_{y=1}^{4} \left(\sum_{a_k \in c_i} \mathbf{q_k}_y\right)^2}} \tag{2.41}$$

for $x = 1 \ldots 4$. This quaternion is used to represent the orientation of the cluster coordinate frame, i.e. the cluster centre. Using Equation (2.38), the cosine distances between each element a_k and the centre of the cluster may be calculated:

$$\hat{d}^{\theta}_{c_i k} = 2 - 2\left(\sum_{x=1}^{4} \hat{\mathbf{q}}_{\mathbf{c_i}x}\mathbf{q}_{\mathbf{k}x}\right)^2 \tag{2.42}$$

The clusters are now further filtered so that the orientational differences between the centre of the current cluster c_i and each of its elements are all below some threshold α. Therefore, the final cluster of aligned fragment-pairs representing the i^{th} cluster is given by:

$$\hat{c}_i = \{a_k \in c_i : \hat{d}^{\theta}_{c_i k} \leq \alpha\} \tag{2.43}$$

Whilst the use of the normalised average quaternion seems adequate for this purpose, in future, should the geometric average of elements of $SO(3)$ be desired, more sophisticated methods are available (e.g. see Moakher, 2003).

Final Rigid Substructure Coordinate Frames

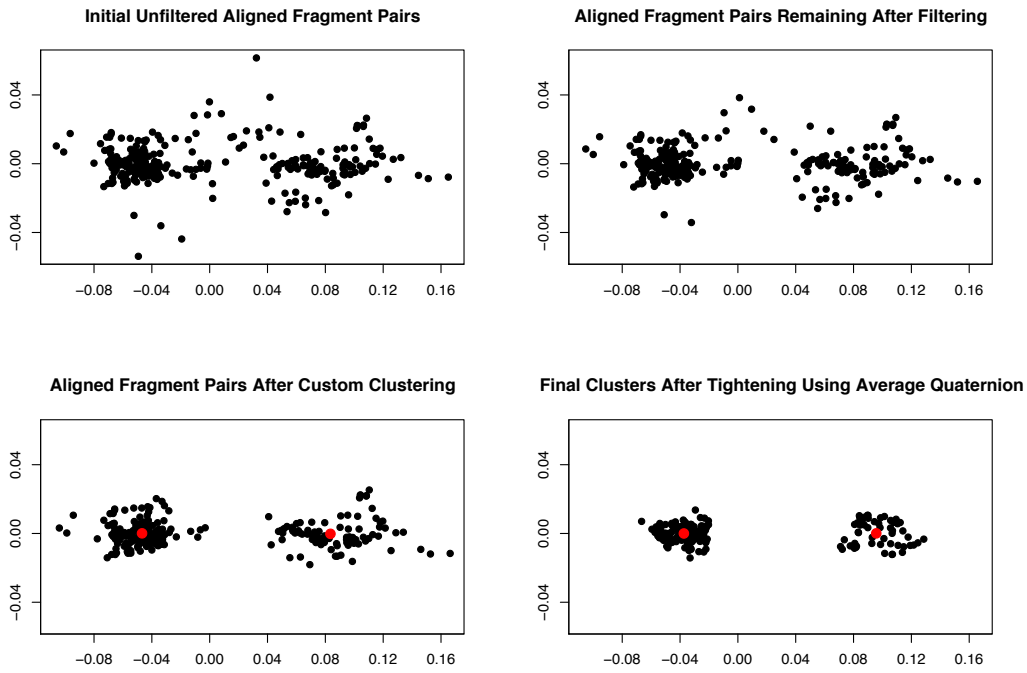
Now that we have identified the final clusters $\hat{c}_i$ to be used in definition of the rigid substructures, it is possible to achieve the final transformations to be used for superposition. For each cluster, the final rotation is given by the normalised average quaternion $\hat{\mathbf{q}}_{\hat{\mathbf{c}}_\mathbf{i}}$, according to Equation (2.40), with corresponding rotation matrix $\mathbf{R}_{\hat{\mathbf{c}}_\mathbf{i}}$ calculated using Equation (2.39). At present, the translation is chosen so as to optimise superposition of fragment-pairs in $\hat{c}_i$ (i.e. so that they have equal means, as with traditional superposition).

Finally, if the mean coordinates of fragments in $\hat{c}_i$ from chains 1 and 2 are given by $\vec{\mu}_{1\hat{c}_i}$ and $\vec{\mu}_{2\hat{c}_i}$, respectively, then the matrix $\mathbf{M_2}$ of atomic coordinates from chain 2 may be transformed so that the coordinate frame corresponding to the identified rigid substructure i in chain 2 is in agreement (according to our method) with that of chain 1:

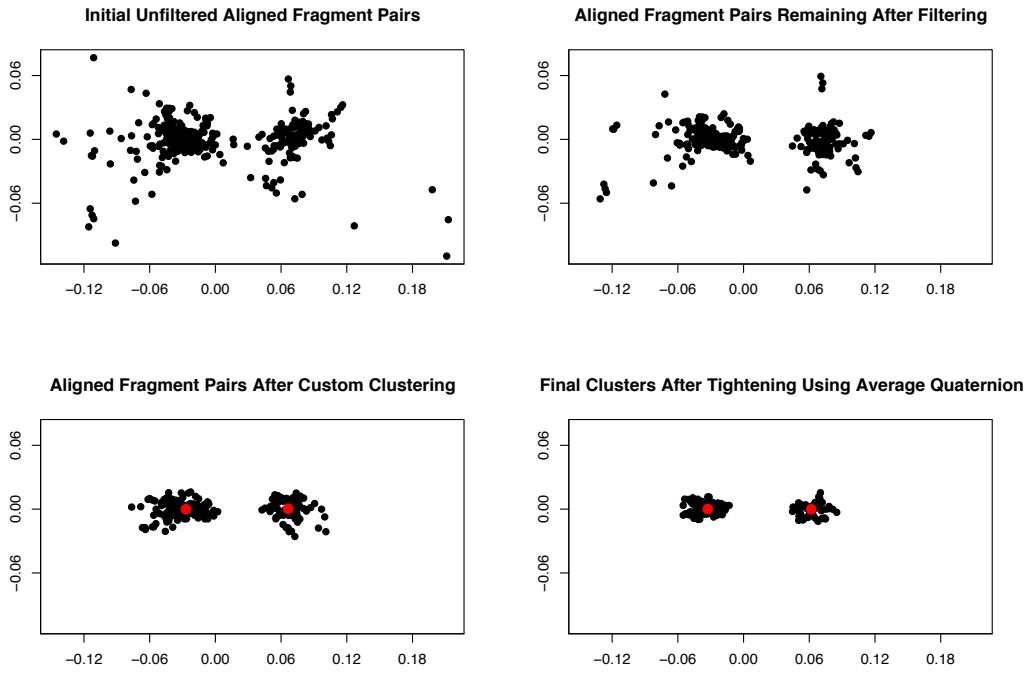
$$\mathbf{M_2'} = \mathbf{M_2}\mathbf{R}_{\hat{\mathbf{c}}_\mathbf{i}} + \vec{\mu}_{1\hat{c}_i} - \vec{\mu}_{2\hat{c}_i}\mathbf{R}_{\hat{\mathbf{c}}_\mathbf{i}} \tag{2.44}$$

according to Equation (2.12).

Although the list of alignment fragment-pairs $\hat{c}_i$ is used for calculation of the coordinate frame definition of substructure i, we do not consider these fragment-pairs to belong exclusively to this substructure. Nor do we consider other fragment-pairs to not belong to this substructure. Rather, it is preferred to take a fuzzy approach, allowing all aligned fragment-pairs to have a degree of belongingness to all identified rigid substructures (our approach is conceptually similar to that of fuzzy logic, but without adopting the formalism; summation to unity is not imposed). In accordance with cluster definition, we choose for the belongingness of aligned fragment-pair a_k to substructure i to be scored by cosine distance $\hat{d}^{\theta}_{\hat{c}_i k}$, describing a fragment-pair's orientational agreement with the substructure. Furthermore, the orientational difference between substructure coordinate frames



(a) Cosine distance clusters for the comparison of 2cex(A) and 2cex(B).



(b) Cosine distance clusters for the comparison of 1a1v(A) and 8ohm(A).

Figure 14: Cosine distances between aligned fragment-pairs, represented using classical multi-dimensional scaling (for illustration purposes only), performed using R (R Development Core Team, 2011), for the comparison of: (a) 2cex(A) and 2cex(B); and (b) 1a1v(A) and 8ohm(A). Each subfigure shows distance between fragment-pairs remaining at various stages: initial unfiltered (upper left); after fragment-pair filtering (upper right); after customised single-linkage clustering (lower left); and final clusters after tightening using the normalised average quaternion (lower right). Red points represent clusters' normalised average quaternions $\hat{\mathbf{q}}_{\mathbf{c_i}}$ (lower left) and $\hat{\mathbf{q}}_{\hat{\mathbf{c}}_\mathbf{i}}$ (lower right).

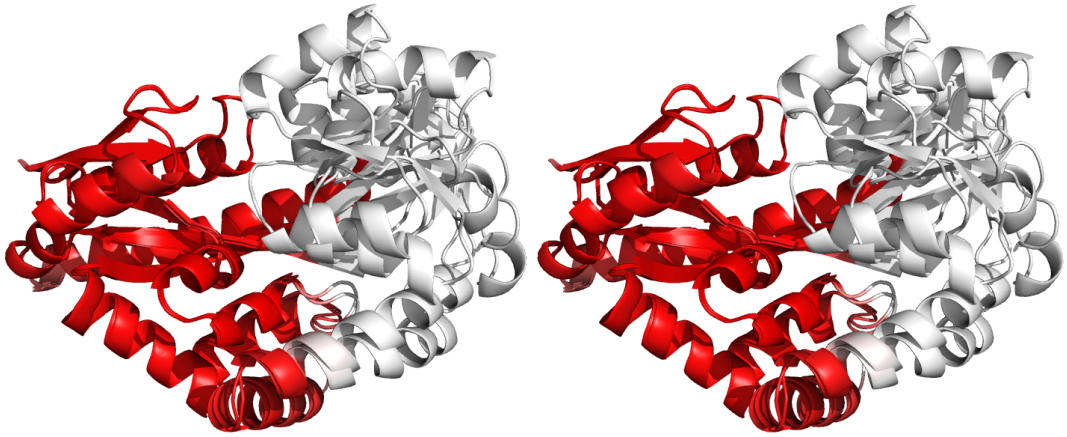
can be described similarly:

$$\hat{d}^{\theta}_{\hat{c}_i\hat{c}_j} = 2 - 2\left(\sum_{x=1}^{4} \hat{\mathbf{q}}_{\hat{\mathbf{c}}_\mathbf{i}x}\hat{\mathbf{q}}_{\hat{\mathbf{c}}_\mathbf{j}x}\right)^2 \tag{2.45}$$

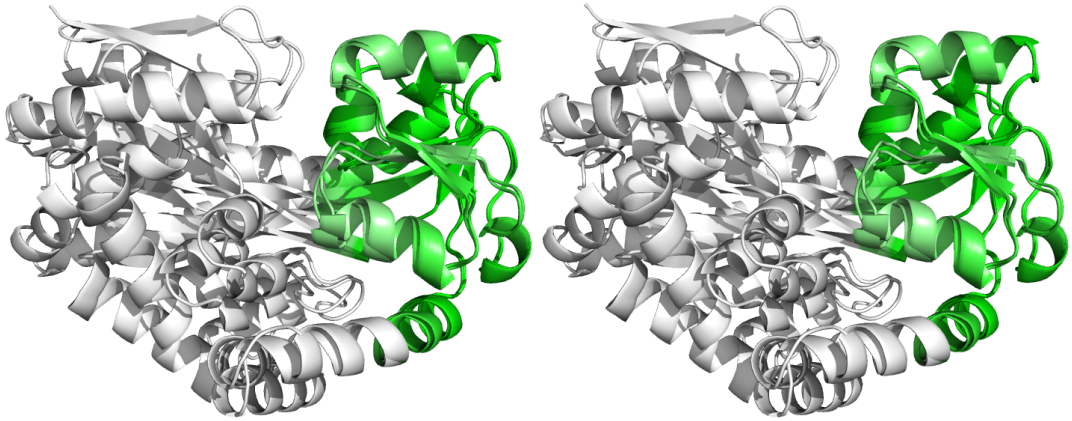
This may be represented as an angle if desired:

$$\theta_{\hat{c}_i\hat{c}_j} = \arccos\left(1 - \hat{d}^{\theta}_{\hat{c}_i\hat{c}_j}\right) \tag{2.46}$$

Figure 14 depicts compactness and separation of clusters at different stages of the rigid substructure identification process, for the two example cases previously shown. In both cases, the fragment-pair filtering stage removes some of the noise and outliers. In the first case (2cex(A) and 2cex(B)), a string of fragment-pairs connects the two clusters. Nevertheless, due to employed

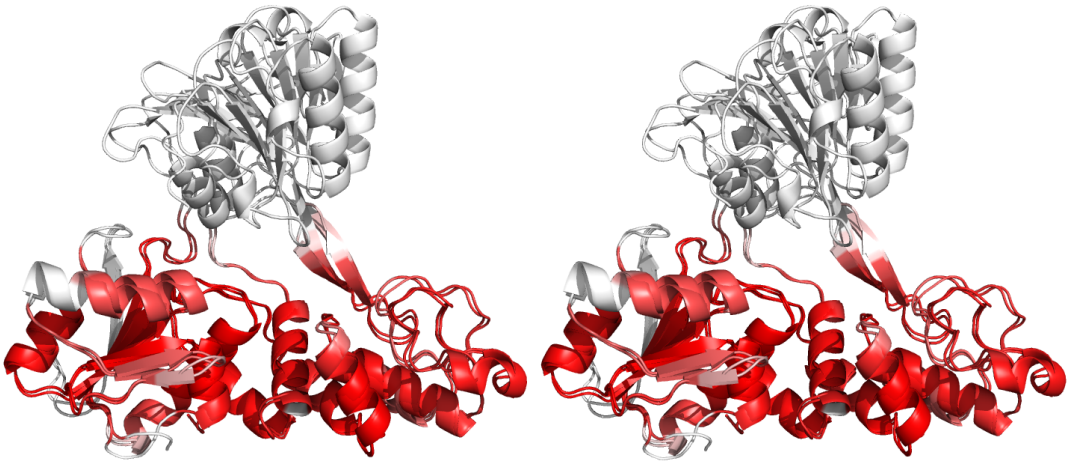


(a) Stereo superposition of the first rigid substructure resulting from the comparison of 2cex(A) and 2cex(B).

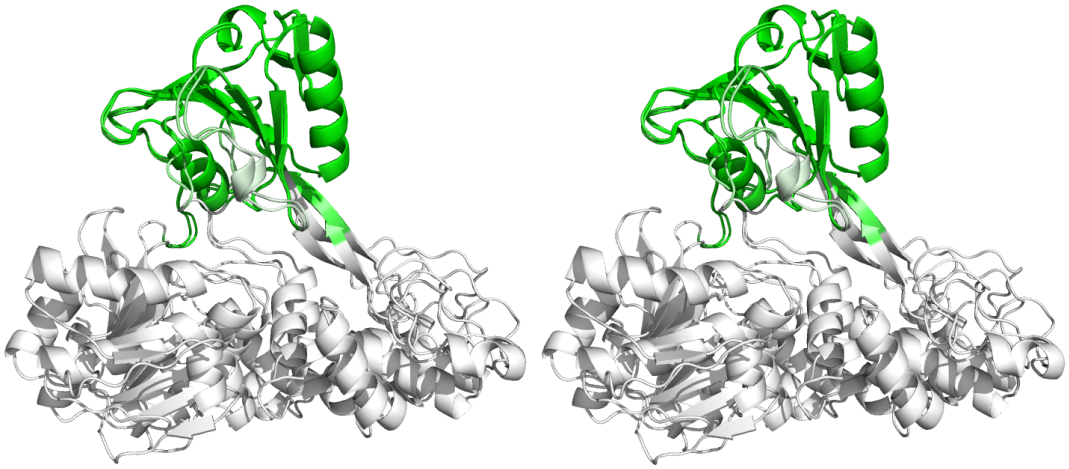


(b) Stereo superposition of the second rigid substructure resulting from the comparison of 2cex(A) and 2cex(B).

Figure 15: Superposition of identified rigid substructures resulting from the comparison of 2cex(A) and 2cex(B), according to the coordinate frames specified by the normalised average quaternions. The first substructure is shown in red (a) and the second in green (b). Residues are coloured according to the best-scoring fragment-pair to which they belong. Colour intensity indicates orientational agreement with the substructure's coordinate frame; all aligned fragment pairs are coloured. Intense colour indicates strong orientational agreement, gradually fading to white at $\hat{d}^{\theta}_{\hat{c}_i k} = 0.005$, corresponding to an angular difference of $\theta_{\hat{c}_i k} \approx 0.1^c$ ($\approx 6°$). Unaligned fragment-pairs are also coloured white.



(a) Stereo superposition of the first rigid substructure resulting from the comparison of 1a1v(A) and 8ohm(A).



(b) Stereo superposition of the second rigid substructure resulting from the comparison of 1a1v(A) and 8ohm(A).

Figure 16: Superposition of identified rigid substructures resulting from the comparison of 1a1v(A) and 8ohm(A), according to the coordinate frames specified by the normalised average quaternions. The first substructure is shown in red (a) and the second in green (b). Residues are coloured according to the best-scoring fragment-pair to which they belong. Colour intensity indicates orientational agreement with the substructure's coordinate frame; all aligned fragment pairs are coloured. Intense colour indicates strong orientational agreement, gradually fading to white at $\hat{d}^{\theta}_{\hat{c}_i k} = 0.005$, corresponding to an angular difference of $\theta_{\hat{c}_i k} \approx 0.1^c$ ($\approx 6^\circ$). Unaligned fragment-pairs are also coloured white.

modification, the single-linkage clustering method is able to distinguish between these clusters. In both cases, the cluster tightening visually increases cluster compactness and separation.

The corresponding identified substructures are superposed in Figures 15 and 16. Being coloured according to the orientational difference between local structure and the particular substructure, it is possible to visualise which residues/fragments have rigidly conserved orientation with the substructure, and which have been forced into a different conformation by the global conformational change (or are more flexible). Using the proposed method, the angular difference between the two identified substructures is estimated as 30° for 2cex(A):2cex(B), and 25° for 1a1v(A):8ohm(A).

2.4 Generation of External Restraints for use in Crystallographic Refinement

2.4.1 Introduction

Heterogeneous organisation of molecules in the crystal lattice can lead to diffraction data being of poor quality. Such heterogeneities may be due to effects such as crystal mosaicity, global chain flexibility, or localised disorder. This results in weak diffraction intensities, causing the data to be collected using a lower resolution threshold. This behaviour is often observed for large complexes. However, structures of individual components of a complex might have been independently solved at a higher resolution. Such information might then be used in order to aid the refinement of the lower-resolution structure.

Information from external sources can be incorporated during refinement using the maximum likelihood framework. Specifically, restraints are generated using external information, with the intention of helping the structure to adopt a conformation that is more consistent with previous observations (prior knowledge). This is conceptually similar to the use of geometric terms in refinement, which aims to help local structure adopt conformations that are chemically reasonable, given presumptions regarding distributions that occur in nature.

External structures are selected on the basis that they are considered to have a higher degree of reliability than the current model, such as solved at a higher resolution, or analytically derived (e.g. secondary structure restraints from an ideal helix).

The use of external restraints may be justified (in some cases) by the resultant increase in reliability of atomic positions. However, it should be acknowledged that such an approach introduces a source of bias; the influence of this bias may result in the model adopting a conformation less consistent with the observed data in the working set. The use of external restraints might make a particular model adopt a conformation very similar to a high-resolution homologue, assuming it is appropriate to do so, and ideally result in improved refinement statistics and geometry. This does not mean that the low-resolution structure can then be considered to have the quality of a high-resolution structure (Wlodawer et al., 2010), due to the empirical uncertainty and inherent bias.

We deduce that external restraints should only be used if the benefits of any improvement in reliability are deemed to outweigh the negative effects of introducing such bias. Indeed, this may well be the case for data of poor quality collected at only low-resolution. For example, refinement of the model might cause some regions of very poor electron density to adopt an incorrect non-sensical conformation. Increasing the weight of geometric terms may help the structure to adopt a more chemically-reasonable configuration, but the region may still be incorrectly modelled due to the effect of the misleading density; geometric restraints operate at a very high level of structural resolution. However, external restraints can operate at a much lower level of structural resolution,

as desired. Consequently, the use of external restraints from a homologous structure that is highly locally conserved with the target structure, in the problematic region, may result in the stabilisation of a sensible configuration in this region in the low-resolution model.

The appropriate selection of external weight is not obvious. One potential optimisation criterion might be to minimise (or optimise) the difference between R and R_{free}. However, this may result in both R and R_{free} rising, thus achieving a worse model. Another criterion might be to choose the weight so as to minimise R_{free}. However, this may reduce the subsequent appropriateness of R_{free} as a statistic for use in validation, since the refinement would no longer be independent of the test set.

The idea of using external structural information to aid macromolecular crystallographic refinement is not novel, and has been used in *SHELX* (Sheldrick, 2008) for many years. External information may currently be utilised in various forms, such as secondary structure restraints, homologous reference structures, and homology models, by various modern refinement software packages including *BUSTER-TNT* (Smart et al., 2012; Blanc et al., 2004), *CNS* (Schröder et al., 2010, 2007; Brunger et al., 1998), *phenix.refine* (Adams et al., 2010; Afonine et al., 2005), and *REFMAC5* (Murshudov et al., 2011, 1997) of *CCP*4 (Winn et al., 2011).

Employed Approach

Given that two protein chains can be aligned using the developed software, it is also possible to generate external restraints for use in crystallographic refinement. Since the alignment approach is invariant to global conformation, the alignment is based on the conservation and scoring of local structure, rather than global agreement. The application in interatomic restraint generation naturally follows since interatomic restraints operate locally, and thus inherently also allow global conformational invariance, depending on the chosen level of structural resolution.

We refer to the chain that is to be be refined as the 'target' chain, and to the chain that is to be used to generate the restraints as the 'external reference' chain. The external reference would usually be chosen to be from an existing structure that was refined at a higher resolution. This structure would usually be homologous, often sequence identical, to the target.

In general, we presume that the target and external reference structures are sufficiently similar, although such assessments should be made by the user. We also assume that these chains have been aligned by *ProSMART ALIGN*, and thus there is a known/assumed residue correspondence between them. Given the positions of two atoms in the target chain and thus the distance d between them, it is possible to find the distance between the corresponding atoms in the external reference chain; this distance r is the objective value of the restraint. If the target and external chains share a high degree of similarity, then we might suspect for d to be approximately equal to r, with some degree of error. Consequently, the restraint distances r, with appropriate distributional estimations, can be used as prior information with the intention of improving the corresponding distances d during

crystallographic refinement.

In cases where utilised external structures are identical in sequence to the target structure, as is often the case, the alignment of structures is trivialised due to correspondence of residue numbers. However, *ProSMART* is general enough to allow any structure to be used as an external reference regardless of sequence (or structural) similarity, irrespective of whether the particular choice of external reference is sensible (suitability should be assessed by the user).

Input PDB files are read and interpreted prior to individual executions of *ProSMART ALIGN* and *RESTRAIN*, allowing faster execution due to reducing the number of redundant runtime operations. Consequently, the effective input files provided to *ProSMART RESTRAIN* are pre-processed PDB files with hydrogen atom entries removed, comprising only main chain atoms unless it is specified for side chain atoms to also be considered. In these files, invalid residues are removed, and residues are re-indexed as in *ProSMART ALIGN* (see §2.1.1). Information regarding knowledge of original residue numbering (and insertion codes, etc.) is maintained.

2.4.2 Identification of Close Atoms

The identification of close atoms is performed separately for the target and external reference chains; results are subsequently utilised for generation of the final restraints. Various methods for near-neighbour searching have been developed; here we use a cell technique (Bentley, 1975) previously used in biology (Levinthal, 1966).

Firstly, we identify integer coordinates representing the range of atomic coordinates, so that all considered atoms in the chain are contained within the cuboid with lowest and largest vertices $c_{\min}$ and $c_{\max} \in \mathbb{Z}^3$, respectively, such that all edges of the cuboid are parallel to one of the axes of the structure's coordinate frame.

Using this cuboid as a reference, space is then uniformly partitioned into cubic cells (as with voxelisation). Each cell has dimensions equal to $r_{\max}$, the chosen maximum possible atomic distance restraint. Given the list of coordinates $C \subset \mathbb{R}^3$ corresponding to atoms in the chain, the list of atoms v_{xyz} contained in voxel (x, y, z) is given by:

$$v_{xyz} = \left\{ c \in C : (x, y, z) < \frac{c - c_{\min}}{r_{\max}} \leq (x+1, y+1, z+1) \right\} \tag{2.47}$$

for $(x, y, z) = (0, 0, 0) \ldots (x_{\max}, y_{\max}, z_{\max}) = \max \left\{ c \in \mathbb{Z}^3 : c < \frac{c_{\max} - c_{\min}}{r_{\max}} \right\}$. All atoms thus belong to indexed voxels ($C = \bigcup v_{xyz}$), with no atoms belonging to multiple voxels ($v_{ijk} \cap v_{xyz} = \emptyset$ for $(i, j, k) \neq (x, y, z)$).

Note that restraints just inside the $r_{\max}$ threshold in the external reference structure may have corresponding atomic distances greater than $r_{\max}$ in the target structure. Consequently, using the same threshold for both target and external structures would result in failure to generate all distance restraints between atoms with distances close to $r_{\max}$ in the external structure. This would greatly inhibit future stages of the restraint generation process, particularly estimation of sigmas.

As a result, we allow the threshold in the target structure to be different to that in the external reference:

$$\begin{aligned} r_{\max} &= r_{\max} && \text{for external structure} \\ r_{\max} &= \alpha r_{\max} && \text{for target structure} \end{aligned} \tag{2.48}$$

where $\alpha > 1$ is arbitrarily chosen to be large enough to include all required restraints, whilst being small enough to not incur too much computational expense (by default, $\alpha = 1.5$). The intention is that identified restraints should be constrained by $r_{\max}$ in the external structure, whilst being effectively unconstrained in the target structure.

Now, all atoms that are sufficiently close to atom i to be identified as a potential interatomic distance restraint (i.e. within $r_{\max}$) are contained within either the same or adjacent voxels. Consequently, a list of atoms containing all atoms sufficiently close to atom i, and some that are not, is given by the union of adjacent voxels:

$$\hat{v}_{xyz} = \bigcup_{i=-1\ldots1} \bigcup_{j=-1\ldots1} \bigcup_{k=-1\ldots1} v_{x+i,y+j,z+k} \tag{2.49}$$

with appropriate boundary conditions. The list of atoms containing only the atoms sufficiently close to atom i is then given by:

$$V_i = \{c_j \in \{\hat{v}_{xyz}\} : |c_i - c_j| \leq r_{\max}, i < j\} \tag{2.50}$$

The implied atom-pairs, specifically atoms i and j such that $c_j \in V_i$, constitute the atom-pairs in a given chain that may be used as restraints (for external structures), or may be restrained (for target structures), independent of the other compared chain. Note that the list V_i comprises only atoms that have higher indexes; this ensures that each atom-pair is included only once, so that restraints are not generated twice.

2.4.3 Identification of Atom-Pairs to be Restrained

Alignment files output by *ProSMART ALIGN* are automatically interpreted in order to obtain information regarding assumed residue correspondences, and associated scores. In particular, *ProSMART RESTRAIN* utilises the 'minimum' main chain and 'average' side chain scores, noting that these scores pertain to agreement of the residues (and their structural environments) in an assumed local coordinate frame, according to the n-residue structural fragments centred on the particular residue-pair.

Since *ProSMART ALIGN* maximises the alignment length regardless of structural similarity, the achieved alignment may include residue-pairs that score poorly, indicating the local structural environment to not be well-conserved. Intuitively, it may make sense for external restraints to not be generated in such cases; we primarily intend these restraints to be used to aid the refinement of

conserved portions of structure. We specify that aligned residue-pair i must satisfy two conditions:

$$\begin{aligned} d_i^{\text{main}} &\leq d_{\max}^{\text{main}} \\ d_i^{\text{side}} &\leq d_{\max}^{\text{side}} \end{aligned} \tag{2.51}$$

where $d_{\max}^{\text{main}}$ is the threshold on the 'minimum' main chain score, and $d_{\max}^{\text{side}}$ is the threshold on the 'average' side chain score, in order for the residues to be considered sufficiently similar for restraint generation.

It may also be possible to use external restraints to encourage structural similarity in seemingly unconserved regions by not filtering the assumed alignment in this way (or alternatively setting high dissimilarity thresholds). Whilst this may aid early refinement in some cases, this is not our focus here.

For each remaining aligned residue-pair i, we obtain the lists of atoms, a_{1i} and a_{2i}, belonging to the aligned residues in the two chains, respectively. If the i^{th} aligned residues are not of the same amino acid type, then only the main chain atoms are considered, so that $|a_{1i}| = |a_{2i}| = 4$. Otherwise, side chain atoms are also considered, if desired, resulting in lists of potentially different lengths ($|a_{1i}|, |a_{2i}| \geq 4$).

Now, let the index of the j^{th} atom in the i^{th} aligned residue from chain x be denoted by a_{xij}. Then the atom in the external chain that corresponds to atom a_{1ij} in the target chain is given by a_{2ik}, where k is such that the atoms are of the same type, as specified in the input PDB file (although note that such a k may not exist).

We may now identify the initial list of atom-pairs to be restrained. By Equation (2.50), the list of atomic coordinates close to the atom with index a_{1ij} in the target chain is given by $V_{a_{1ij}}$. Consequently, interatomic distance restraints would be between atomic coordinates c_{1ij} and $c_{1xy} \in V_{a_{1ij}}$. The corresponding coordinates in the external reference structure would be given by c_{2ik} and $c_{2xz} \in V_{a_{2ik}}$, such that the atoms with indexes a_{1ij} and a_{2ik} are of the same type, and the atoms with indexes a_{1xy} and a_{2xz} are of the same type (if such atoms exist). The corresponding interatomic distance restraints are thus identified:

$$\begin{aligned} d_{mn} &= |c_{1ij} - c_{1xy}| & c_{1xy} &\in V_{a_{1ij}} \\ r_{mn} &= |c_{2ik} - c_{2xz}| & c_{2xz} &\in V_{a_{2ik}} \end{aligned} \tag{2.52}$$

subject to the above conditions and existence, where $m = a_{1ij}$ and $n = a_{1xy}$. Here, d_{mn} is the current distance between the positions of atoms m and n in the target structure, and r_{mn} is the objective value of the restraint, which is the distance between the assumed corresponding atoms in the external reference structure.

Restraint Filtering

Restraints between atom-pairs that are already tightly restrained by principle geometric terms are not included in the list of external restraints, by default. In particular, we remove the short restraints

that are separated by only one or two chemical bonds, which are already restrained in *REFMAC* (i.e. bond and angle restraints). These atom-pairs have relatively few effective conformational degrees of freedom, and thus have reasonably tight interatomic distance distributions relative to longer-range restraints, as can be seen in Figure 17.

In general, for both short and longer restraints, different types of atom-pairs will have different interatomic distance distributions, as is evident by the peaks in Figure 17. Future studies may consider the separate estimation of restraint distributions for different types (or classes) of interatomic distances, although such further categorisation is not considered here.

A crude pre-filter may be used to remove short restraints by allowing a threshold on the minimum restraint distance, requiring:

$$r_{mn} \geq r_{\min} \tag{2.53}$$

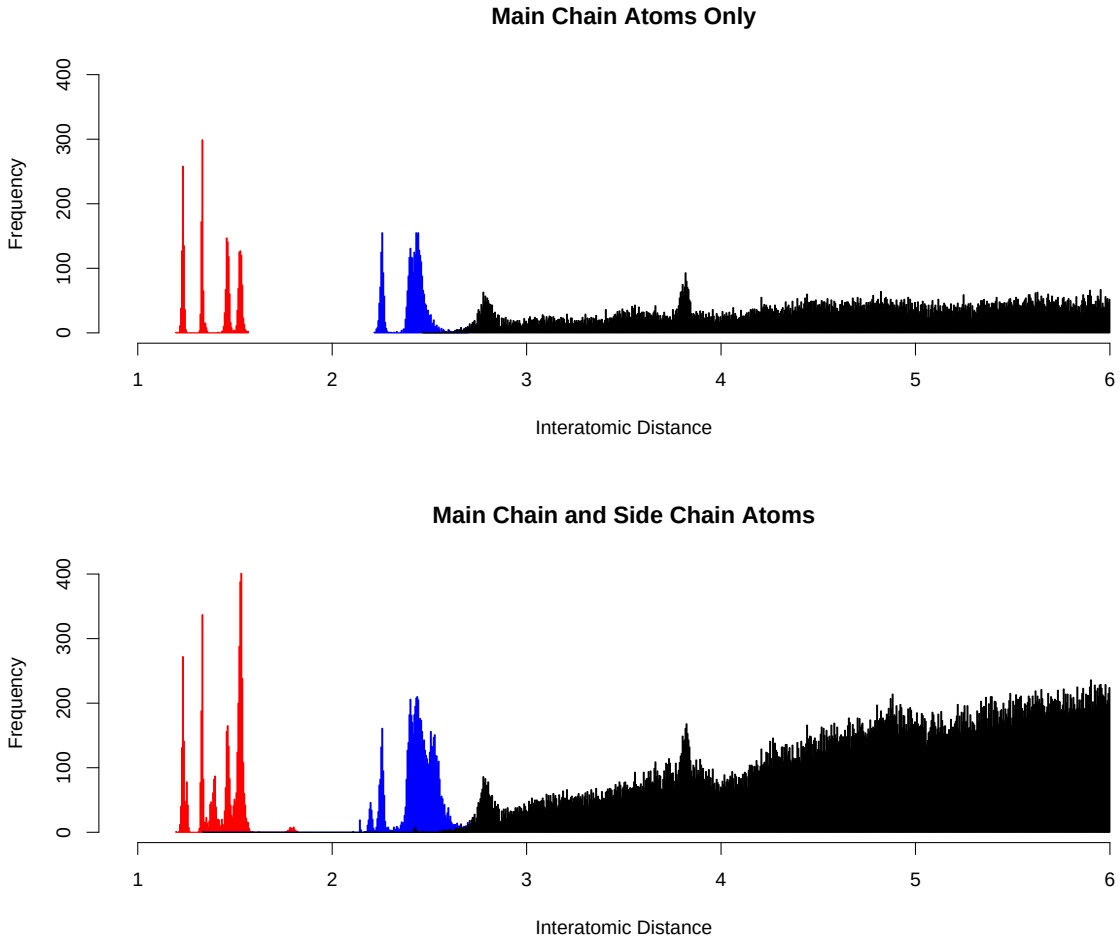


Figure 17: Histograms of the interatomic distances in the structure with PDB ID 2jhp (Sutton et al., 2007). The upper image corresponds to main chain atoms only; the lower image displays interatomic distances for both main and side chain atoms. Distances corresponding to atom-pairs separated by one chemical bond are shown in red, those separated by two bonds are shown in blue, and all other atom-pairs are shown in black.

such that $0 \leq r_{\min} < r_{\max}$. For example, a threshold $r_{\min} = 2.5$Å may be chosen so as to remove many of the already effectively restrained interatomic distances.

The restraints are inspected to identify those separated by one or two chemical bonds; these are removed from the restraints list. This is only performed for restraints between main chain atoms, and is achieved analytically (rather than based on distance criteria).

If restraints are to be generated for side chain atoms, then *REFMAC* is used to generate a list of bonded atom-pairs. Following the conversion of residue indexing to the appropriate format, bonded atom-pairs are removed from the list of restraints. The list of atom-pairs separated by two bonds is then inferred, and these are removed from the restraints list also. The primary reason why *REFMAC* is used in this case is in order to ensure compliance and future compatibility, regardless of side chain atom nomenclature.

If *REFMAC* is to be used for this purpose, it is automatically executed prior to execution of *ProSMART RESTRAIN*, but in parallel with *ProSMART ALIGN*. If the *REFMAC*-generated bonds file for a given input PDB file is already present, then it is not re-generated.

2.4.4 Fragment-Based Restraints

Further to generating restraints using an external reference structure, *ProSMART* is able to generate restraints using individual structural fragments. This functionality may have broad application in the generation of restraints for secondary structural elements. In particular, external restraints may be generated using an n-residue fragment corresponding to an 'ideal' α-helix, which may be useful in situations where it is desirable to improve the geometry of helices. Such restraints might be used when a suitable structurally similar chain is not available for use as an external reference, or when the reference chain is itself not sufficiently well-refined. However, the suitability of other general restraints, such as β-strand restraints, is less obvious due their comparatively high degrees of conformational flexibility. Another potential application would be when it is desired for a particular region to adopt a known conformation; the suitability of such an approach would have to be carefully considered for the particular case.

Since aligned fragments may overlap (e.g. consecutive helical fragments), it is possible for a particular atom-pair to be restrained to various atom-pairs in the reference fragment. For example, in a helical fragment, the distances between main chain atoms in residues i and j may be very similar to those in residues $i + 1$ and $j + 1$. Therefore, restraints for a target atom-pair in a helix might be generated using corresponding atoms from residues i and j, or those from residues $i + 1$ and $j + 1$ (and possibly others) in the reference fragment. In such cases, it is necessary to decide which residues to use for restraint generation.

More generally, any restraint between atoms from residues i and j may result from various fragment alignments. Specifically, the reference fragment, which has residue range $[1, n]$, may be aligned to any of the residue ranges $[j - n + 1, j], \ldots, [i, i + n - 1]$ in the target structure, whilst

still implying correspondences for residues i and j (providing $i > j - n$). Therefore, ignoring heterogeneities and boundary conditions, there may be up to $i - j + n$ potential alignments of residues i and j with some residues in the reference fragment.

The list of potential residue correspondences is reduced by fragment score criteria, since we only want to generate fragment-based restraints for regions of structure sufficiently similar to the reference fragment; only configurations with associated fragment alignment scores below some threshold are included (this selection process uses the fragment type identification functionality of *ProSMART ALIGN*). Of the remaining potential residue-pair alignments, if any, the one with the most favourable associated fragment Procrustes score is selected.

In implementation, identification of appropriate residue-pair correspondences between the target structure and the reference fragment is performed by *ProSMART ALIGN*. This information is interpreted by *ProSMART RESTRAIN*, which then deduces appropriate atom-pair correspondences, and continues to generate restraints in the same fashion as when using an external reference structure.

2.4.5 Maximum Likelihood Estimation of Restraint Distributions

Further to removing close restraints for the purpose of not generating restraints for atom-pairs that are already restrained in *REFMAC*, their removal is vital for purposes of sigma estimation. It is reasonable to surmise that appropriate variability of long-range restraints is qualitatively very different to that of restraints that have few conformational degrees of freedom. Specifically, it might be expected that the distance between atom-pairs separated by one chemical bond might have very low variability (depending on bond type); those separated by two bonds might have higher variability; those separated by three bonds, higher variability again, etc. Consequently, it would make sense for the qualitative nature of the distribution of restraints to be investigated separately for atom-pairs separated by different numbers (and perhaps different types) of atomic bonds.

Here, we assume that the qualitative nature of trends in the variability of restraints is comparable for all atom-pairs separated by at least three atomic bonds. In future, it would be of benefit to further explore the effect of bond separation on the qualitative nature of restraint variability.

Form of the Restraint Distributions

Suppose the distributions of the positions of two atoms in the target structure are $\vec{x}_1 \sim N(\vec{c}_1, \sigma_1^2)$ and $\vec{x}_2 \sim N(\vec{c}_2, \sigma_2^2)$, where $\vec{c}_i$ is the coordinate corresponding to atom i. Since we are generally interested in low-resolution structures, the assumption of spherical Normality seems reasonable; the variance terms are scalar to emphasise this point. Note that B-factors are closely related to the variabilities of these distributions, and these parameters are usually chosen to be isotropic for low-resolution structures.

The distribution of vectors from the first atom to the second is given by:

$$\Delta\vec{x} = \vec{x}_2 - \vec{x}_1 \sim N\left(\vec{c}_2 - \vec{c}_1, \sigma_1^2 + \sigma_2^2 - 2\text{cov}(\vec{x}_1, \vec{x}_2)\right) \tag{2.54}$$

If the atoms are close, then their positions are likely to be correlated, which will reduce the variability of the distance between them. Conversely, if the atoms are far apart, then it is reasonable to surmise that their positions would be more independent, and thus the variability of their interatomic distance would be larger. This is supported by Figure 18, which demonstrates lower variability for atom-pairs closer together.

The distribution of interatomic distances is then given by $D = \sqrt{\sum_{i=1}^{3} \Delta\vec{x}_i^2}$. Since $\sqrt{\sum_{i=1}^{3} \left(\frac{\Delta\vec{x}_i}{\sigma}\right)^2}$ follows a noncentral chi distribution with 3 degrees of freedom and non-centrality parameter $\sqrt{\sum_{i=1}^{3} \left(\frac{E(\Delta\vec{x}_i)}{\sigma}\right)^2}$, we deduce that D is related to the noncentral chi distribution; specifically, $D\sigma^{-1} \sim \chi'_3$ where $\sigma^2 = \text{var}(\Delta\vec{x})$. This relies on the assumption of isotropic variability of atomic positions, resulting in σ being equal in all directions. Consequently, in future, such distributional assumptions might be utilised in estimation of the distribution of restraints, and in concert imple-

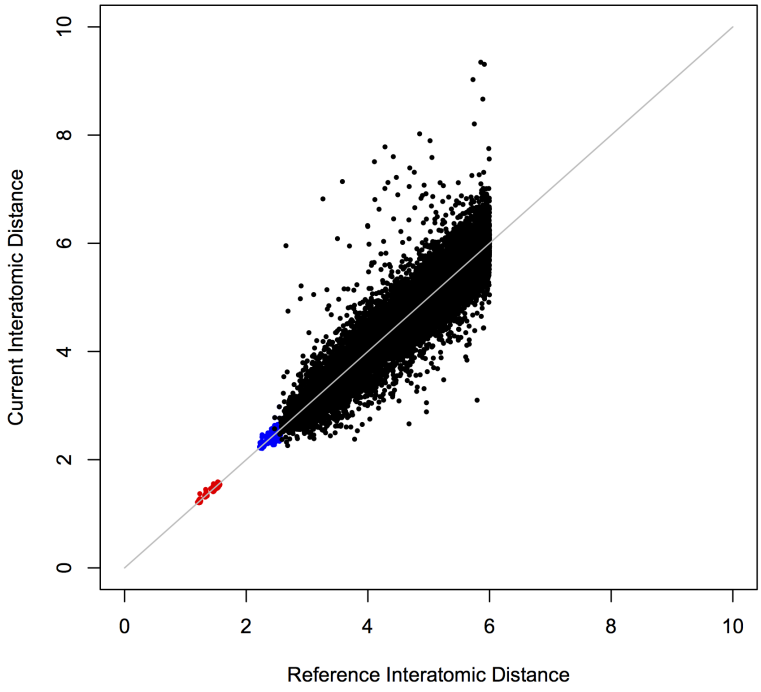


Figure 18: Distance dependence of the distribution of interatomic distances (for main chain atoms only), for the target structure 2jha (Sutton et al., 2007), using sequence-identical 2jhp as the external reference. The graph shows the interatomic distance in 2jha against the corresponding distance in 2jhp, for all atom-pairs whose interatomic distance is less than 6Å in 2jha. Distances corresponding to atom-pairs separated by one chemical bond are shown in red, those separated by two bonds in blue, and all others in black. A diagonal grey line is shown to aid visualisation, representing equivalence of interatomic distances.

mented in the crystallographic refinement of low-resolution structures. The practical applicability of this should be investigated in future.

However, since external interatomic distance restraints are currently implemented assuming Normally distributed residuals, here we also use this approximation. The posterior distribution of an interatomic distance, which had original value d, is given by:

$$D \sim N(\mu, \sigma^2) \tag{2.55}$$

which constitutes the restraint to be used in refinement. Given knowledge of the external reference structure, we are able to estimate the mean as the distance between the corresponding atoms in the external structure, so that $\mu = r$, where d and r are calculated using Equation (2.52). Estimation of the standard deviation σ is less obvious, and various approaches may be taken.

In *ProSMART*, restraints may be generated with one of three types of sigma:

- Constant pre-specified sigmas;
- Constant estimated sigmas;
- Distance-dependent sigmas.

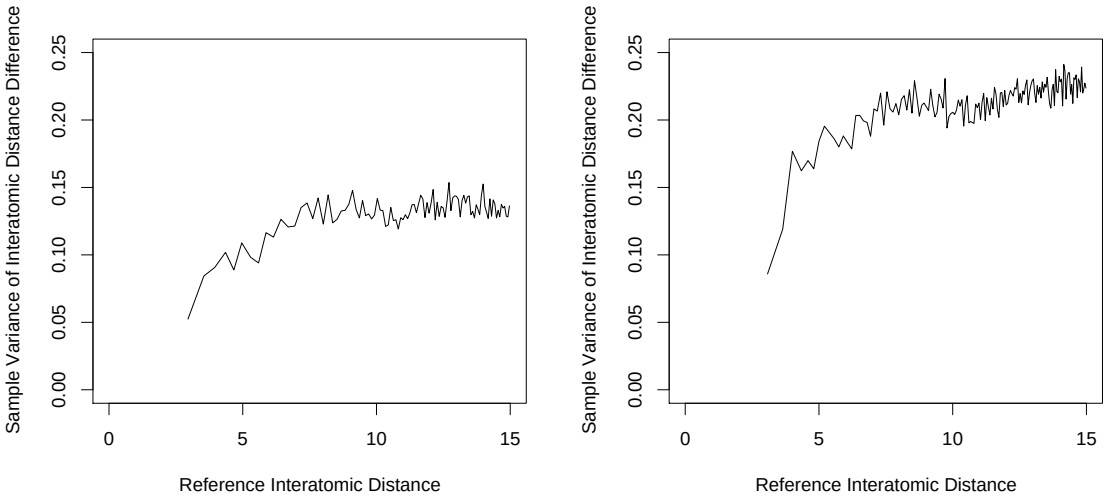
The use of constant pre-specified sigmas gives the user control over the choice of sigmas (by default, $\sigma = 0.1$ is arbitrarily chosen), and allows the same sigmas to be generated for different chain-pairs, if so desired.

Constant estimated sigmas allow restraints to be automatically uniformly weighted according to the overall agreement between corresponding atom-pairs in the target and external structures. This should result in lower weights being assigned when the external structure is less similar to the target, in terms of net local structural agreement. Note that this information is derived directly from the distribution of all restraints' mean values throughout the whole structure, and does not depend on the fragment-based alignment score.

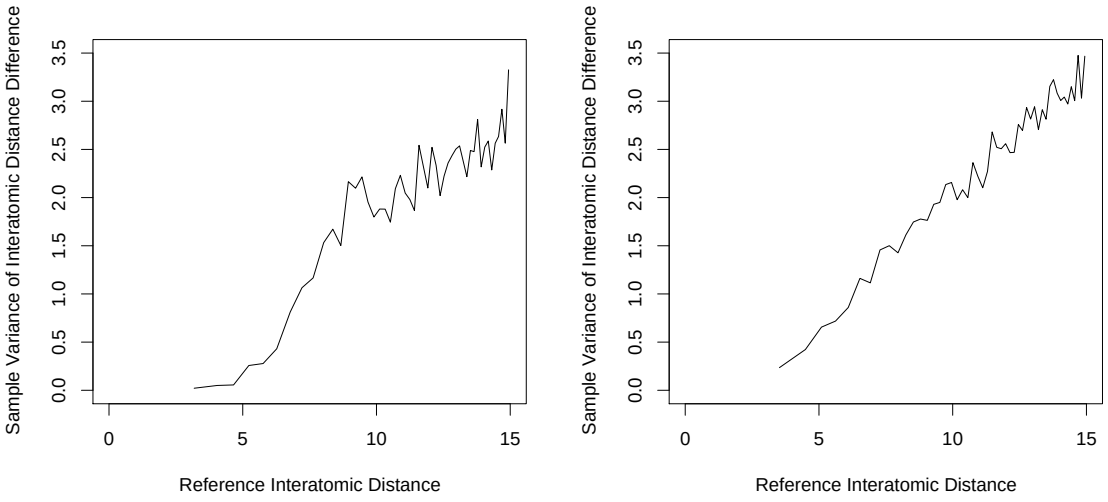
Similarly, distance-dependent sigmas are estimated using the observed distribution of restraints. However, the restraint variance is allowed to increase linearly with restraint distance r. As restraint distance increases, any signalling causing correlation in atomic position will become weaker, in general. This causes restraint variability to increase with the mean $\mu = r$. However, peculiar behaviour may be observed when there are multiple rigid substructures (e.g. domains) present; this would be exacerbated when the maximum restraint distance $r_{\max}$ is large. Specifically, the presence of multiple domains will tend to cause a deterministic increase in observed variability. Examples of the distance-dependent variability of restraint distributions are shown in Figure 19. Increased variability is observed with increased restraint distance, especially in the presence of global conformational change. Even in the absence of conformational change, correlation with distance is observed for lower restraint distances.

In summary, constant estimated sigmas take the form:

$$\sigma^2 = k \tag{2.56}$$



(a) Variability of restraints for 2jha using the external reference 2jhp.



(b) Variability of restraints for 2cex(A) using the external reference 3b50.

Figure 19: Sampled variance of the restraint distributions, for (a) 2jha (Sutton et al., 2007) using the external reference 2jhp; and (b) 2cex(A) (Muller et al., 2006) using the external reference 3b50 (Johnston et al., 2008). The subfigures represent the distribution of restraints, using sequence-identical chains as external reference structures, in the absence (a) and presence (b) of global conformation change. In both subfigures, the left images correspond to restraints between only main chain atoms, and the right images correspond to restraints for both main chain and side chain atoms. Variability is calculated about the assumed mean, which is given by the interatomic distance in the reference structure. For each image, variance is sampled in bins, chosen so that all bins contain an equal number of observations.

and distance-dependent sigmas take the form:

$$\sigma^2 = k_1 + k_2 r \tag{2.57}$$

Other functional forms may also be considered; suitability of the inclusion of other terms, such as those based on B-factors or structural (dis)similarity scores, could be investigated in future. Whilst only these two simple cases are implemented at present, the implemented maximum likelihood approach is sufficiently general to require relatively little modification in order to allow estimation of parameters from other more complicated functional forms.

In cases where the target and external chains are identified as structurally identical (detected using residue-based alignment scores), the method automatically defaults to using constant pre-specified sigmas, avoiding the realisation of arbitrarily small sigmas.

After restraints have been generated, including calculation/estimation of sigmas, *ProSMART* allows the sigmas to be scaled. Specifically, all sigmas may be multiplied by a constant value. Also, sigmas may be scaled by the average of B-factors of the restrained atom-pair. However, such post-modification is not used, by default.

Maximum Likelihood Estimation of Distributional Parameters

In order to achieve estimates of the sigmas, the parameters k (for constant sigmas), or k_1 and k_2 (for distance-dependent sigmas) must be estimated. This is achieved by considering the distributions of interatomic distances d in the target structure and r in the external reference structure. Parameters are optimised using maximum likelihood estimation. This is performed during *ProSMART RESTRAIN* runtime, so that the parameters are specific to the correlations observed between the particular chain pair. In future, upon the identification of sufficient reliable cases, it would be of benefit to investigate whether these parameters are dependent on certain factors, such as whether they can be predicted using structural alignment scores or B-factor distributions.

In general, given Equation (2.55), the probability density function of D is given by:

$$f_D\left(d; \mu = r, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(d-r)^2}{2\sigma^2}} \tag{2.58}$$

Using this method, given N restraints, parameters are optimised when the likelihood $L(\vec{k}) = \prod_{i=1}^{N} f_D\left(d_i; r_i, \sigma_i^2\right)$ is maximised. The likelihood and negative log-likelihood functions directly follow:

$$L(\vec{k}) = (2\pi)^{\frac{-n}{2}} \prod_{i=1}^{N} \sigma_i^{-1} e^{\frac{-(d_i-\mu_i)^2}{\sigma_i^2}}$$

$$\text{-log}(L) = \frac{n}{2}\log(2\pi) + \sum_{i=1}^{N} \log(\sigma_i) + \frac{1}{2}\sum_{i=1}^{N} \frac{(d_i - \mu_i)^2}{\sigma_i^2} \tag{2.59}$$

The gradient with respect to the parameters is required for optimisation of the likelihood function. Assuming the mean is known (i.e. does not depend on any parameters), and an arbitrary functional

form of the variance $\sigma^2 = \sigma^2(\vec{k})$, the gradient of the negative log-likelihood with respect to the j^{th} parameter is given by:

$$\frac{-\partial log(L)}{\partial k_j} = \frac{1}{2}\sum_{i=1}^{N} \frac{1}{\sigma_i^2}\frac{\partial \sigma_i^2}{\partial k_j}\left(1 - \frac{(d_i - r_i)^2}{\sigma_i^2}\right) \tag{2.60}$$

Finally, the differential of the variance with respect to the parameter k for constant estimated sigmas is given by:

$$\frac{\partial \sigma_i^2}{\partial k} = 1 \tag{2.61}$$

and for distance-dependent sigmas with parameters $\vec{k} = (k_1, k_2)$:

$$\frac{\partial \sigma_i^2}{\partial \vec{k}} = \begin{bmatrix} 1 \\ r_i \end{bmatrix} \tag{2.62}$$

in accordance with Equations (2.56) and (2.57).

The optimisation problem amounts to searching for parameter values $\vec{k}$ such that the constraints $\frac{-\partial log(L)}{\partial k_j} = 0$ are satisfied for all j, within some acceptable error margin, so that the likelihood function $L(\vec{k})$ is sufficiently maximised. Note that other (non-Normal) distributional forms could be considered and handled using this method. Minimisation of the negative log-likelihood function is performed using a quasi-Newton method, in which an approximation of the Hessian matrix is improved/updated after each procedural iteration. Specifically, we use the BFGS formula for updating the (inverse) Hessian approximation, and a line search algorithm for selecting trial parameter values. See Nocedal and Wright (1999) for details of the method.

Chapter 3

Software Implementation and Output

3.1 Design and Implementation of ProSMART

3.1.1 Overall Procedural Design

The developed software *ProSMART* is a command line tool written in C++, designed for use on Unix-based systems. Further to the main application, *ProSMART*, the software comprises two separate modules, *ProSMART ALIGN* and *ProSMART RESTRAIN*, which are effectively hidden from the end user. The purpose of the main application is to handle input, execution of the modules (allowing parallel batch processing), and reporting final results, as appropriate. *ProSMART ALIGN* is used for the pairwise alignment of protein chains, and related functionalities. Subsequently, the resultant output may be utilised by *ProSMART RESTRAIN* for the generation of external restraints for use in crystallographic refinement. Either one or both of these modules may be executed (iteratively) in any given session.

As a minimum requirement, one target PDB file must be provided. However, multiple PDB files may be specified as target or secondary (reference) inputs. Specifying only target PDB file(s) results in an all-on-all comparison, using the target chains also as secondary chains. Chains do not have to be explicitly specified; if chains are not specified then the input PDB files are inspected to obtain the list of all valid chains. Specification of other arguments and parameters is optional.

In cases where multiple chains are considered, whether as target or secondary, individual instances of *ProSMART ALIGN* and/or *ProSMART RESTRAIN* may be executed in parallel. In addition, *REFMAC* may be executed in parallel with *ProSMART ALIGN* when used to generate bond lists. The number of simultaneous *ProSMART* child processes running at any given time may be limited, as desired. Specifically, process IDs are monitored; whenever a process terminates, new processes are launched providing the maximum allowed number of child processes is not exceeded.

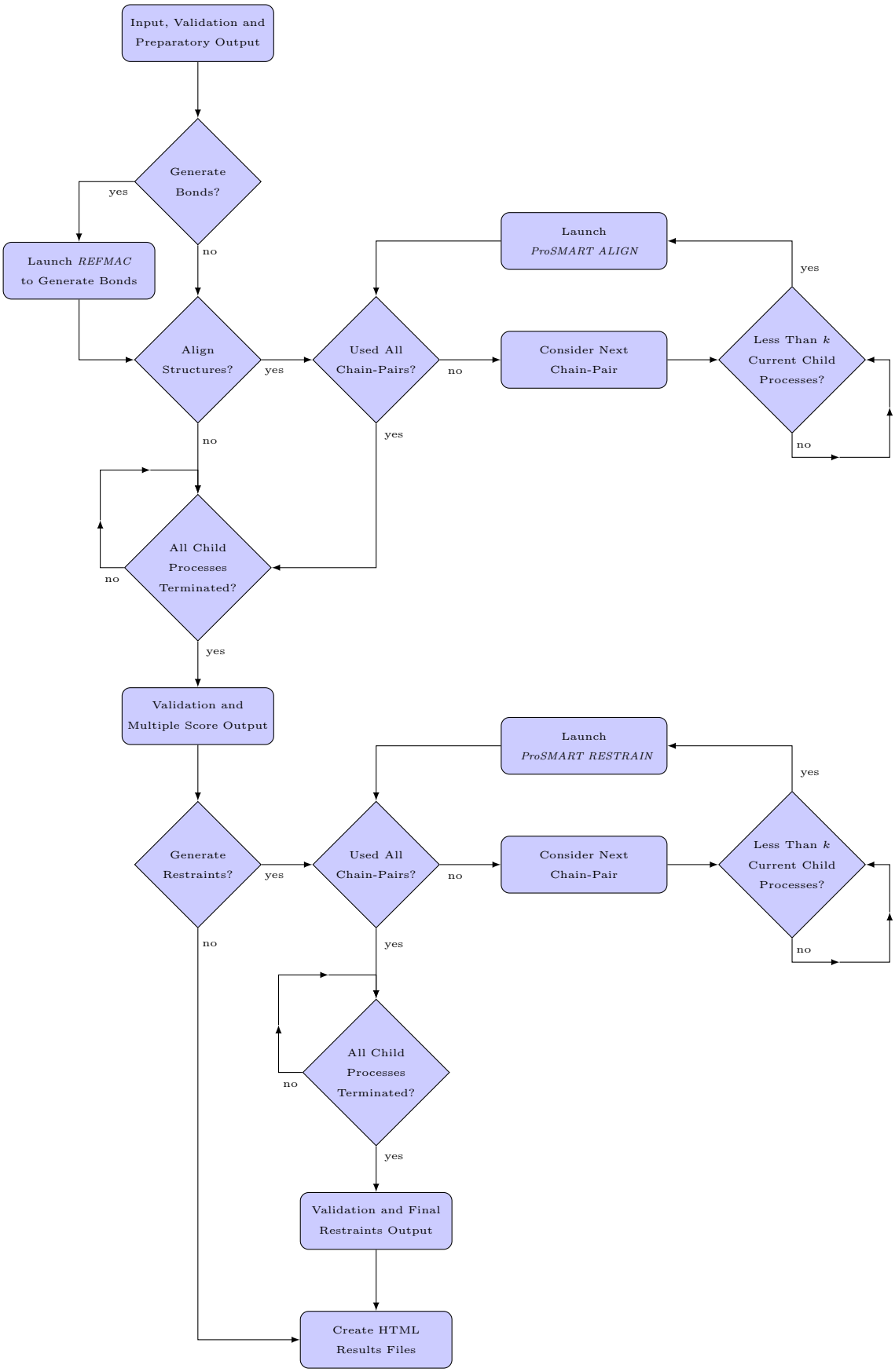


Figure 20: Flow chart illustrating the general procedure involved in *ProSMART* execution. The parameter k specifies the chosen maximum number of simultaneous child processes, which may include instances of *ProSMART ALIGN*, *ProSMART RESTRAIN*, and *REFMAC*.

An overview of the main *ProSMART* procedure is shown in Figure 20. The various stages involved may be chronologically summarised as follows:

- *Interpret input arguments*, which are used to determine the operations to be performed, and optionally alter parameter values. Keywords may be specified as command line arguments, or provided in an external file. The session output directory is created; name and location may be specified. If required, an XML format file is created that contains runtime information, and also acts as an error/warning log.

- *Input validation*, to ensure that all arguments were interpreted correctly, and that parameters satisfy certain criteria. Check that input PDB format files are provided, and are unambiguously specified according to the design.

- *Directory structure initialisation*, creating subdirectories within the session output directory. Actual directory structure may vary depending on program options.

- *Installation verification*, ensuring that *ProSMART* is setup correctly. This involves confirming that the *ProSMART ALIGN* and *RESTRAIN* binaries and the *ProSMART* library exist in the expected locations, and that permissions allow read/write access to the directory used for storing temporary files.

- *Determine jobs to be performed*, which involves identification of the list of files/chains to be compared, and confirm file existence. If individual chains are not specified (independently for target and secondary input files), then the files are parsed and all valid chains identified. If the fragment library is to be used (instead of any secondary chains) then the library configuration file is read. This file provides the locations of the fragment PDB files, and associated parameter values.

- *Process input PDB files*, parsing each file and removing any chains from the job list that do not contain at least one valid structural fragment of appropriate length. Coordinate entries corresponding to each input chain are interpreted and converted into a bespoke format. This involves renumbering residues[1] (as specified in §2.1.1), reducing the amount of unutilised information stored, converting to a more efficient storage method, and removing hydrogen atoms, secondary alternative conformations, and invalid residues (i.e. those without all four main chain atoms present). The reformatted chains are then written to file, allowing faster file reading upon subsequent iterations of *ProSMART ALIGN* and *RESTRAIN*. Since the initial file processing only has to be performed once, this results in a more efficient procedure overall, particularly for large batch jobs.

- *Final preparatory output*. Further subdirectories are created, based on the filenames and chain IDs of the input chains. Also, program log files and HTML format results files are

[1] note that a reference to original residue nomenclature is maintained.

initialised/reinitialised.

- *Generate list of bonded atom-pairs*, if appropriate. If external restraints are to be generated, atoms separated by one or two chemical bonds may be removed from the list of restraints. If side chain restraints are to be generated, the list of bonded atom-pairs may be obtained using *REFMAC*. In this situation, *REFMAC* is launched as a child process prior to execution of, and in parallel with, *ProSMART ALIGN*. One instance of *REFMAC* is launched for each target PDB file, whilst ensuring that the maximum number of allowed simultaneous child processes is not exceeded.

- *Launch ProSMART ALIGN*, if appropriate. For each chain-pair to be aligned, a script is written that launches *ProSMART ALIGN* with appropriate input arguments. These scripts are iteratively executed for each chain-pair. Each instance of *ProSMART ALIGN* is a separate process. Multiple instances are allowed to execute in parallel, subject to ensuring that no more than a prespecified number of child processes are running at any given time (including any instances of *REFMAC*).

- *Wait for all child processes to terminate*, whether *ProSMART ALIGN* or *REFMAC*.

- *Validation*. Existence of expected resultant alignment files is confirmed, and these files are inspected to ensure they satisfy validation criteria. Similarly for any bonds files generated by *REFMAC*.

- *Create multiple-chain output files*. The pairwise global score files are read (this also involves validation by existence), combined into multiple score matrices, and output to file. If the fragment library is used, fragment type alignment files are constructed by combining results from individual alignment files. If structures are presumed sequence-identical, further alignment files and colour scripts are output, corresponding to the worst residue-based alignment scores over all chain-pairs, on a per-residue basis (this is particularly useful for structure ensembles).

- *Generate list of chain-pairs to be used for restraint generation*, depending on selection option. Specifically, for each chain from the list of targets, restraints may be generated using either: all reference (secondary) chains; or only one chain. In the latter case, the chains are selected that have the best global alignment score, as specified by the *ProSMART ALIGN* alignment. By default, if a given chain is present in both target and secondary chain lists, restraints are not generated for this self-pairing unless specified.

- *Launch ProSMART RESTRAIN*, if appropriate. For each chain-pair, a script is written that launches *ProSMART RESTRAIN* with appropriate input arguments. These scripts are iteratively executed for each chain-pair. Each instance of *ProSMART RESTRAIN* is a separate process. Multiple instances are allowed to execute in parallel, subject to ensuring that no more than a prespecified number of child processes are running at any given time.

- *Wait for all child processes to terminate.*
- *Validation.* Existence of expected resultant restraints files is confirmed, and these files are inspected to ensure they satisfy validation criteria.
- *Create multiple-chain restraints files.* Individual pairwise restraints files are concatenated into final restraints files, as appropriate. Each target PDB file, which may contain multiple chains, has its own final restraints file. These final restraints files comprise all generated restraints for all considered chains in that PDB file.
- *Create final results files*, which are generated in HTML/JavaScript format. Actual output depends on chosen program options and input arguments.

System Requirements and Dependencies

ProSMART has various dependencies as a result of design and implementation, although attempts have been made to ensure a reasonable degree of platform-independence on modern Unix-based systems. Being written in C++, an appropriate compiler is required (GNU g++ by default). Standard Unix utilities (e.g. 'cat', 'make', 'mkdir', 'rm'), shells (specifically '/bin/bash'), and common C/C++ libraries are required, and their presence is assumed.

ProSMART uses data structures from the TNT package (Pozo, 1997) and a C++ translation (Pozo, 2003) of the JAMA package (Hicklin et al., 2000) to calculate the singular value decomposition of a matrix. Consequently, these public domain packages are also required, which are non-standard libraries; the appropriate header files will be distributed with *ProSMART*.

For displaying the output HTML results files, a modern web browser is required, with JavaScript enabled. Generated external restraints are designed for use with *REFMAC* (Murshudov et al., 2011, 1997), and colour scripts for use with *PyMOL* (Schrödinger, LLC, 2010; DeLano, 2007).

3.1.2 Presentation of Output

The three *ProSMART* applications output results in the form of various files, which should be subsequently manually inspected or interpreted/used by external software applications. Actual output files depend on the operations to be performed; for example, different files are output depending on whether a secondary chain or a fragment library is to be used. Potential output files include:

- *ProSMART ALIGN*
  - Log file;
  - Residue alignment and scores;
  - Fragment categorisation, according to the library;
  - Fragment scores, for each library entry;

- Global scores and statistics;
- Superposition transformation matrices;
- Global superposed PDB files;
- Rigid substructure superposed PDB files;
- Pairwise chain colour scripts;
- Individual chain fragment-based colour scripts.

- *ProSMART RESTRAIN*
  - Log file;
  - External structure atomic distance restraints;
  - Fragment-based atomic distance restraints.

- *ProSMART*
  - XML log file;
  - Multiple-chain global score matrices;
  - Multiple-chain *PyMOL* loader scripts;
  - Colour scripts for structure ensembles;
  - Multiple-chain external restraints files;
  - Sequence files;
  - HTML-format results pages.

ProSMART has the potential to produce a large number of output files, especially when running batch jobs with multiple target and/or secondary chains. Consequently, navigation of output directories and identification of the desired output files may be (and has previously proved) overwhelming for unfamiliar users. This issue is addressed by the provision of a HTML-format results page that aims to provide a more intuitive interface for viewing results, simplifying navigation of the output directories. Furthermore, this output ignores any previous executions of *ProSMART*, displaying only the results from the most recent execution using the particular output directory (if the same output directory is used for multiple *ProSMART* executions, then any previous output files are not deleted, but are replaced if necessary). The HTML-format results pages (and other required hidden files) are created during runtime, bespoke for each *ProSMART* execution.

Figure 21 displays an example of a *ProSMART* results page. All categories are displayed on the left; clickable items are displayed in blue. Categories are enabled or disabled depending on the options chosen upon execution of *ProSMART*. For the example in Figure 21, it is evident that both *ProSMART ALIGN* and *RESTRAIN* were executed, the fragment library was not used, and an XML-format log was not created. Coherently displaying all enabled categories (blue) in this

ProSMART Results

Completed: 03/08/2011 15:55:32

Procrustes Structural Matching Alignment & Restraints Tool

Runtime Info
- Input Parameters
- ProSMART Align Log
- ProSMART Restrain Log
- XML Log

Residue Alignment/Scores
- Pairwise
- Fragment Type
- Fragment Scores

Global Scores
- Pairwise
- Multiple

Superpositions
- Transformations
- PDB Files

Colour Scripts
- Pairwise
- Fragment Scores

Atomic Bond Restraints
- External Structure
- Fragment

Other
- Sequence
- External Links

Chain Pair

	2cey_A
2cex_A	2cey_A
2cex_A	2v4c_A
2cex_A	3b50_A
2cex_B	2cey_A
2cex_B	2v4c_A
2cex_B	3b50_A
2cex_C	2cey_A
2cex_C	2v4c_A
2cex_C	3b50_A
2cex_D	2cey_A
2cex_D	2v4c_A
2cex_D	3b50_A

[Navigate Directory]

Source: ProSMART_Output/Output_Files/Residue_Alignment_Scores/2cex_A/2cex_A_2cey_A.txt

# ProSMART Alignment File									
Res1	Res2	AA1	AA2	Type1	Type2	SideRMS	SideAV	Min	Central	Rotate
2	2	ASP	ASP	S	S	1.498	0.4954	0.3253	NA	NA
3	3	TYR	TYR	S	S	0.4401	0.3836	0.2102	NA	NA
4	4	ASP	ASP	S	S	0.2334	0.11	0.2083	NA	NA
5	5	LEU	LEU	S	S	0.1574	0.09957	0.2083	NA	NA
6	6	LYS	LYS	S	S	0.7165	0.4819	0.2083	0.3253	0.006522
7	7	PHE	PHE	S	S	0.2747	0.2405	0.2083	0.2102	0.003331
8	8	GLY	GLY	S	S	0.1924	0.1924	0.2083	0.2083	0.0003299
9	9	MET	MET	S	S	0.2125	0.05506	0.2083	0.2213	0.003228
10	10	ASN	ASN	S	S	0.3699	0.3474	0.202	0.2312	0.006902
11	11	ALA	ALA	S	S	0.3392	0.2993	0.202	0.2278	0.007742
12	12	GLY	GLY	NA	NA	0.1538	0.1538	0.202	0.231	0.006026
13	13	THR	THR	NA	NA	0.349	0.2589	0.1904	0.2111	0.002049
14	14	SER	SER	NA	NA	1.675	1.085	0.1895	0.202	0.001035
15	15	SER	SER	H	H	0.2375	0.2129	0.1791	0.2058	0.003694
16	16	ASN	ASN	H	H	0.3804	0.3456	0.1791	0.2035	0.003085
17	17	GLU	GLU	H	H	0.3006	0.2276	0.1676	0.1904	0.002963
18	18	TYR	TYR	H	H	0.2323	0.166	0.1676	0.1895	0.004272
19	19	LYS	LYS	H	H	2.683	1.881	0.1676	0.1791	0.003377
20	20	ALA	ALA	H	H	0.3017	0.2941	0.1606	0.182	0.002929
21	21	ALA	ALA	H	H	0.2702	0.2547	0.1435	0.1676	0.001546
22	22	GLU	GLU	H	H	0.2576	0.1984	0.1435	0.1679	0.001328
23	23	MET	MET	H	H	0.4523	0.2614	0.1435	0.1733	0.003759
24	24	PHE	PHE	H	H	0.1728	0.1417	0.1435	0.1606	0.00252
25	25	ALA	ALA	H	H	0.1658	0.1569	0.1435	0.1435	0.001487
26	26	LYS	LYS	H	H	4.657	3.381	0.1435	0.1532	0.0003969
27	27	GLU	GLU	H	H	0.2571	0.1466	0.1435	0.1614	6.412e-05
28	28	VAL	VAL	H	H	0.09101	0.06455	0.1435	0.1593	0.001779
29	29	LYS	LYS	H	H	0.337	0.2001	0.1435	0.1672	0.001241
30	30	GLU	GLU	H	H	0.1872	0.1648	0.1532	0.1819	0.001055
31	31	LYS	LYS	H	H	1.014	0.4437	0.1593	0.173	0.000539
32	32	SER	SER	H	H	0.423	0.3724	0.1593	0.1708	0.0006595
33	33	GLN	GLN	H	H	0.2854	1.294	0.1672	0.1788	0.002389
34	34	GLY	GLY	S	S	0.1035	0.1035	0.1708	0.1918	0.001707
35	35	LYS	LYS	S	S	1.833	1.302	0.1708	0.1827	0.001366
36	36	ILE	ILE	S	S	0.2543	0.1983	0.1708	0.1989	0.001718
37	37	GLU	GLU	S	S	0.4271	0.2203	0.1788	0.2013	0.002167
38	38	ILE	ILE	S	S	0.3329	0.2296	0.1827	0.198	0.003944
39	39	SER	SER	S	S	0.3018	0.275	0.1827	0.1868	0.005546
40	40	LEU	LEU	S	S	0.2388	0.2137	0.1868	0.2108	0.001635
41	41	TYR	TYR	S	S	0.2344	0.1809	0.1868	0.1967	0.002446
42	42	PRO	PRO	S	S	0.08259	0.04281	0.1868	0.2094	0.001598
43	43	SER	SER	S	S	0.1778	0.147	0.1868	0.2191	0.001697

Figure 21: Screenshot of an example *ProSMART* results page, displayed using Mozilla Firefox5. The example features the comparison of 2cex (chains A, B, C and D) against 2cey, 2v4c, and 3b50 (this selection of chains is also considered in §3.3.4). The image illustrates inspection of the output alignment/scores file corresponding to the pairwise comparison of 2cex(A) and 2cey(A).

way provides the user with a relatively comprehensive list of all types of output generated during runtime. This aims to inform the inexperienced user of some of the available functionalities. The presence of greyed-out categories subtly informs the user that other functionalities are available in *ProSMART*, which would require the specification of other input arguments.

To the right of the list of categories is a dynamic context-dependent frame, which changes content depending on the selected category. In this case, the selection box is used to specify the chain-pair of interest; clicking on a chain-pair will result in the content of the frame on the right to change accordingly. The frame on the right is used to display the content of the *ProSMART* output files as they appear on disk, in text or HTML format. The location of the file displayed in the frame is automatically displayed above, allowing the user to know where the corresponding source file is located. Clicking the 'navigate directory' button results in the frame on the right listing the contents of the directory where the selected source file is located; this graphical interface can then be used to navigate through the file system, providing manual control as desired.

The *ProSMART* package includes user documentation, which may also be currently found online (http://www.ysbl.york.ac.uk/mxstat/Rob/User_Documentation.html), that provides information regarding usage. This includes compilation and installation instructions, and examples of basic usage and protocol. A list of input arguments is also provided, forming a reference to available functionality.

3.1.3 Internal Batch Processing Performance

The issue of computational performance is generally of greatest concern when running large batch jobs, e.g. performing an all-on-all alignment of structures in a database, or scanning a target structure against a large database. In contrast, external restraints would typically only be generated for a few chain-pairs in any given execution. We deduce that computational performance has the potential to be an issue of great importance for *ProSMART ALIGN*, whilst being less important for *ProSMART RESTRAIN*. Consequently, we focus only on the performance of *ProSMART ALIGN*.

A small dataset of PDB files was created in order to test computational performance. This dataset comprised 30 structures (chains) of various types and lengths, all of which were solved using X-ray crystallography. A range of different structures was selected, so that results might be deemed representative of dissimilar structures. However, it is important to clarify that the dataset was not designed with any intent of reliably representing the vast range of structures in the PDB. Rather, the main aim was to achieve a small sample of structures with a wide range of chain lengths.

Chains selected for inclusion in the dataset are listed in Table 1. In creation of the dataset, 3 chains from each of 10 bins were selected, where each bin represents a 100-residue chain length interval (i.e. 3 chains with length in $[100, 200]$, 3 with length in $[200, 300]$, etc.). Within each bin, selected chains had at most 10% sequence similarity; this condition was enforced using *PDBselect* (Griep and Hobohm, 2010). Subject to these criteria, chains were arbitrarily manually selected for

PDB ID	N_{res}	PDB ID	N_{res}	PDB ID	N_{res}	PDB ID	N_{res}	PDB ID	N_{res}
7fd1 A	106	1us0 A	313	2vqr A	512	2e26 A	705	2wyh A	905
2wyt A	153	2dej A	346	1q6z A	524	1w7c A	737	2ivf A	912
1x8q A	184	3k01 A	391	3c8y A	574	1h16 A	759	3og2 A	986
2hlc A	230	3lov A	451	3moe A	618	3ahc A	802	3kll A	1007
2vxn A	249	3o6w A	467	2dy1 A	660	3b34 A	866	1k32 A	1023
2v8b A	279	1gwe A	498	3ju4 A	670	2xos A	899	3gjx A	1041

Table 1: List of the 30 chains comprising the dataset used for performance testing. PDB codes, chain identifiers, and the number of residues are displayed.

inclusion.

All performance tests were performed on a modern desktop computer running Ubuntu 10.04. Specifically, unless otherwise stated, on a quad-core Intel Core i7 3Ghz CPU (Bloomfield processor; Nehalem microarchitecture), with 6GB 1600Mhz DDR3 RAM, and a standard 7200rpm hard disk. *ProSMART* does not use a considerable amount of RAM, and RAM speed was not found to have a noticeable effect on performance, within standard DDR3 specifications. For consistency, the Intel Turbo-Boost feature was disabled, insuring that any performance inconsistencies may be attributed to fluctuations in base clock speed and system resource allocation. Measurement accuracy is a potential source of error, although this was not deemed to be a major issue, given the strength of observed trends.

Multi-Chain Performance Scaling

As previously mentioned, *ProSMART* allows the parallel processing of batch jobs, by means of simultaneous execution of multiple child processes. Since each thread has to share resources, such as hard drive access for performing read and write file operations, multithreading performance scaling is considered in order to investigate the efficiency of batch processing within *ProSMART*.

Multi-chain performance was tested by means of an all-on-all (half-matrix) comparison of the chains in the test dataset, varying the parameter controlling the maximum number of allowed *ProSMART* child processes. This resulted in a total of 435 ($30 \times 29 \div 2$) pairwise comparisons by instances of *ProSMART ALIGN*. Being mainly interested in the performance of *ProSMART ALIGN*, and since it does not make sense to generate external restraints for random dissimilar structures, *ProSMART RESTRAIN* was not executed. Execution time was measured as the difference between system times at launch and termination of *ProSMART*[2].

Results of the batch processing performance scaling tests are shown in Figure 22. Due to the embarrassingly parallel (Foster, 1995) nature of the design and implementation of the parallel portion of *ProSMART*, the negative effects of Amdahl's law (Amdahl, 1967) are not observed,

[2]Measurements were recorded from within *ProSMART*, measured in microseconds using the 'gettimeofday' function from the C POSIX library.

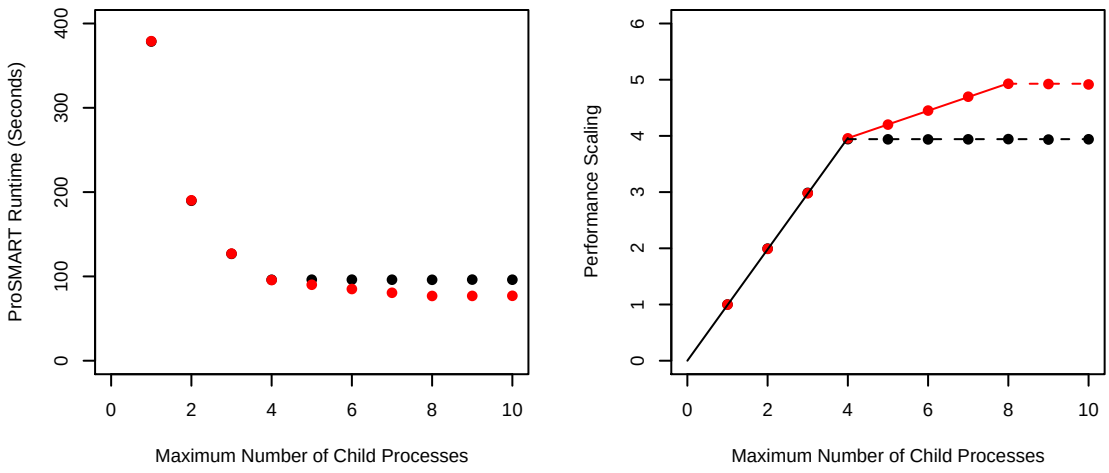


Figure 22: The left image shows the relationship between observed overall computation time of *ProSMART* and the maximum number (x) of simultaneous instances of *ProSMART ALIGN*. Performance scaling is shown on the right, relative to the observed computation time when $x = 1$ (i.e. 378.6/seconds). In both graphs, black and red points (and lines, in the graph on the right) correspond to observed times when Intel Hyper-Threading is disabled and enabled, respectively. The solid black line corresponds to the regression line based on observations for $x = 1 \ldots 4$ with Hyper-Threading disabled; the dotted black line corresponds to the value at $x = 4$ (i.e. 3.94). The solid red line corresponds to the regression line based on observations for $x = 4 \ldots 8$ with Hyper-Threading enabled; the dotted red line corresponds to the value at $x = 8$ (i.e. 4.93).

with scaling behaving more in line with Gustafson's law (Gustafson, 1988). Note, however, that Amdahl's law may take effect in cases where a very large number of cores are available (this could be investigated). Since the serial components of *ProSMART* are generally extremely fast relative to the parallel components, we might surmise that this effect may not be noticeable in practical application, providing the number of pairwise comparisons greatly exceeds the number of cores (e.g. database scanning; all-on-all comparisons). Specific results may depend on the particular platform.

Here, approximately linear scaling is observed up to the execution of four *ProSMART ALIGN* threads, with a gradient of 0.982, not dissimilarly to results from some other applications (Barker et al., 2008). A gradient below unity is expected, not only due to resource allocation, but also due to the operations performed in *ProSMART* that are executed using only one thread, both before and after the iterative execution of *ProSMART ALIGN*. Furthermore, linear scaling could only possibly be achieved if the problem could be perfectly factorised, which would not quite occur in practice. With Hyper-Threading disabled, approximately constant performance is observed when executing more than four simultaneous threads, in the range considered; specifically, performance is approximately 3.94 times better than when allowing only one simultaneous *ProSMART ALIGN* thread.

Enabling Hyper-Threading allows the simultaneous execution of up to eight threads, despite having only four physical CPU cores (by utilising unused memory cycles). This results in increased

performance when running more than four threads, with maximum performance being realised when allowing eight *ProSMART ALIGN* threads to execute simultaneously. This further increase in performance is also linear, although the relative performance gain is greatly reduced (likely due to resource sharing), having an observed gradient of 0.244. Again, approximately constant performance is observed when executing more than eight simultaneous threads, in the range considered. Specifically, performance is observed to be 4.93 times better than when allowing only one simultaneous *ProSMART ALIGN* thread. Note, however, that resource availability and allocation may reduce performance if a very large number of threads are simultaneously executed.

In summary, almost-relative linear scaling can be observed when using *ProSMART*'s batch processing capabilities, up to the number of available CPU cores. Further performance improvements may be seen when the cores are capable of handling multiple simultaneous threads. For optimal performance, the number of allowed simultaneous *ProSMART* child processes (threads) should be chosen to be equal to the number of threads that the CPU can simultaneously handle (higher values are not advised).

CPU Speed Performance Scaling

It is hypothesised that the performance of *ProSMART* is largely dependent on CPU speed. The alternative is that other factors (e.g. RAM speed/capacity, hard drive speed, and allocation of resources) act as a bottleneck, limiting the potential for increased performance at higher CPU speeds. To test this, the CPU multiplier was varied, whilst keeping other factors (e.g. RAM speed, base clock speed) constant. In order to maximally utilise the processor, Hyper-Threading was enabled and the maximum number of allowed *ProSMART* child processes chosen to be eight, in accordance with the above results.

Results suggest the number of aligned structures per second to scale linearly with increased CPU speed, as can be seen in Figure 23, indicating that *ProSMART* greatly benefits from faster CPU speeds. There is no indication that the performance increase diminishes for high CPU speeds, suggesting there to be no major bottleneck inhibiting performance. We conclude that performance is highly dependent on CPU speed as well as the number of available cores/threads.

The results demonstrate how an all-on-all comparison of the 435 chains in the test dataset can be performed in less than 60 seconds on a modern single-CPU desktop computer (results may depend on system configuration). Average runtimes per chain-pair were observed as low as 0.13 seconds. However, chain length has a considerable impact on performance, as observed in §3.2. Computation times would be expected to be much quicker for short chain-pairs, and much slower for long chain-pairs. Note also that the distribution of chain lengths in the test dataset is not representative of structures in the PDB. Specifically, the observed average runtime of around 0.13 seconds would be expected to be realised for a dataset of chains each comprising approximately 600 residues. For comparison, given the same test setup, average runtimes of around 0.04 seconds

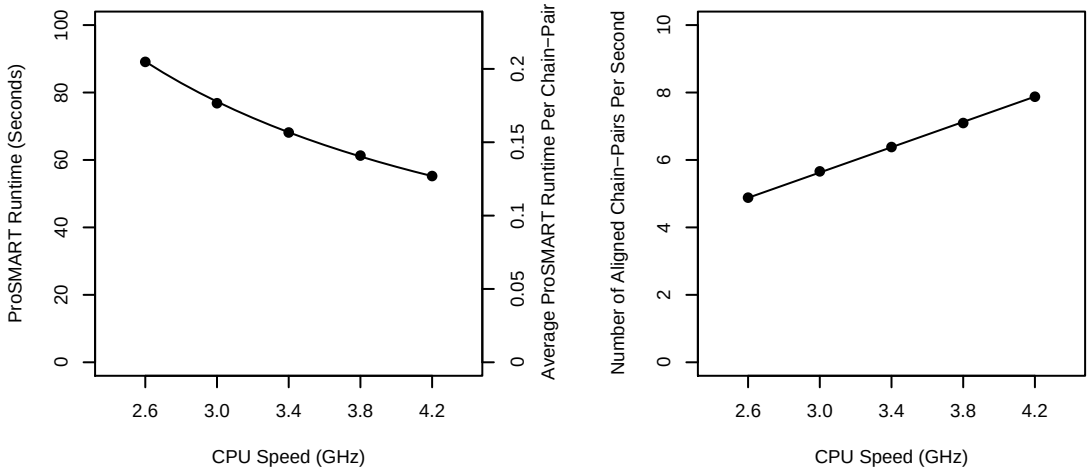


Figure 23: The left image shows the relationship between overall computation time of *ProSMART* and CPU speed. The average frequency of *ProSMART ALIGN* executions per second is shown on the right. Black points correspond to observations, and black lines to the linear regression model of average frequency of *ProSMART ALIGN* executions per second against CPU speed. The intercept parameter was not significantly different from zero, and thus was not included in the model. The gradient parameter was estimated as 1.88.

would be expected to be realised for a dataset of chains all comprising approximately 300 residues (data not shown).

These results suggest that the use of *ProSMART* in computationally expensive applications, such as database scanning, may be feasible. Future improvements in computing technologies will continue to increase the relevance and potential of this application as CPU compute power increases. Indeed, modern CPUs are currently available that exhibit more cores, and higher frequencies, than that of our test setup.

3.2 Performance of ProSMART ALIGN

3.2.1 Speed of ProSMART ALIGN Components

The *ProSMART* process was split into several major components, in order to investigate the degree to which they influence overall computation time. This allowed empirical identification of dependencies on certain variables (most notably chain lengths), as expected according to the algorithms' orders of complexity. Considered variables included only those that could be inferred directly from the two chains separately.

Model intercept parameters were only included where significant. Variance-stabilising transformations, or similar alternatives, were used where appropriate. For simplicity, all models use a Normal error structure (given a transformation, where appropriate). Coefficients of determination (R^2 values) were calculated directly from the observed and fitted values, not from the regression models; specifically, the square of the correlation between the observed and predicted values is

reported. For clarity, this allows comparability for a variety of different models with/without an intercept parameter.

Due to being interested in the performance of individual instances of *ProSMART ALIGN*, the comparison of each chain-pair was executed separately and consecutively. Consequently, Hyper-Threading was disabled, and the parallel processing capabilities of *ProSMART* were not used for any performance tests in this section; times reported correspond to the utilisation of only one thread. Indeed, performance would be expected to scale in accordance with the results of §3.1.3.

Unless otherwise stated, each of the tests involved an all-on-all (half-matrix) comparison of structures in the test dataset introduced in §3.1.3, excluding the diagonal elements representing the alignment of a chain with itself (due to being only interested in the alignment of dissimilar structures). Tests were performed with seven different fragment lengths ($n = 3, 5, 7, 9, 11, 13, 15$), the central value $n = 9$ being the default (note that fragment length must be odd, and has minimum value 3). Consequently, each test involved 3045 ($30 \times 29 \times 7 \div 2$) pairwise executions of *ProSMART ALIGN*.

File Input

Reading and interpretation of the input files is handled by *ProSMART* externally to the *ProSMART ALIGN* process. Re-formatted input files are written, which are subsequently read by *ProSMART ALIGN*. The file input stage comprises the reading and interpretation of these re-formatted files, and the creation of 'Residue' and 'Fragment' objects.

As expected, computation time was found to heavily depend on the lengths of the two chains, and also slightly (but significantly) on the chosen fragment length (n), resulting in the model:

$$t = \alpha n + (|F_1| + |F_2|)(\beta + \gamma n + \varepsilon) \tag{3.1}$$

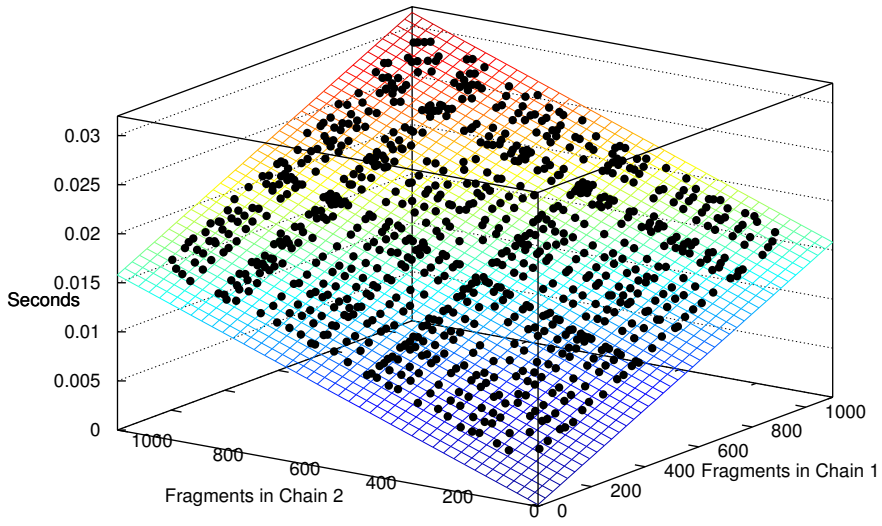


Figure 24: Computation time of the file input stage against the numbers of fragments in chains 1 and 2, $|F_1|$ and $|F_2|$, respectively. Black points correspond to observed times when using fragment length 9. The surface depicts the expectation of Equation (3.1), with $n = 9$.

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 2.189 \times 10^{-5}$, $\beta = 1.400 \times 10^{-5}$, $\gamma = 2.050 \times 10^{-8}$, and $\sigma = 3.775 \times 10^{-7}$, with $R^2 = 0.9948$. The model for $n = 9$ is shown in Figure 24.

Calculation of the Distance Matrix

Computation time (t seconds) was found to depend on the product of the number of fragments in the two chains, $|F_1||F_2|$, as can be seen in Figure 25. Whilst the employed calculation of the distance matrix does involve linear components in $|F_1|$ and $|F_2|$ (i.e. computation of the trace of the self-covariance matrices $\mathrm{tr}(\hat{\mathbf{F}}_{\mathbf{1i}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{1i}})$ and $\mathrm{tr}(\hat{\mathbf{F}}_{\mathbf{2j}}^{\mathbf{T}}\hat{\mathbf{F}}_{\mathbf{2j}})$), these linear terms were not found to be significant. This is because the quadratic component, calculation of individual elements of the distance matrix, dominates computation time.

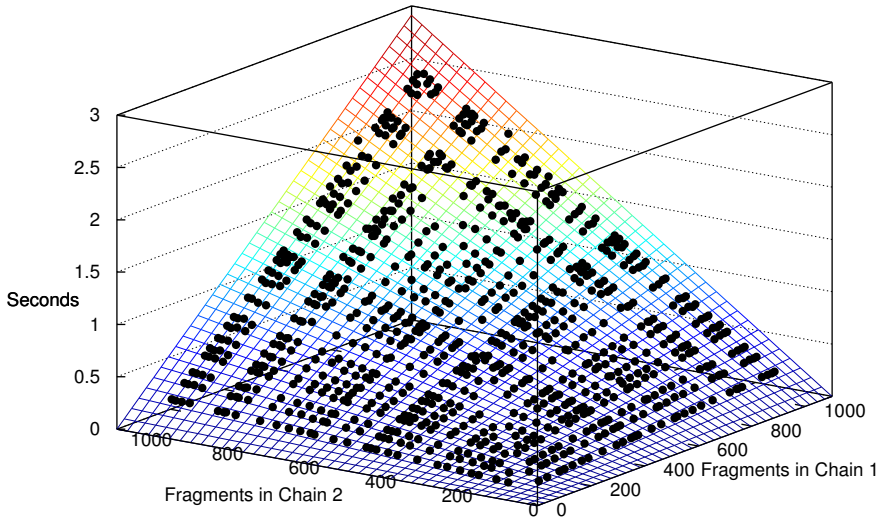


Figure 25: Computation time of the fragment distance matrix against the numbers of fragments in chains 1 and 2, $|F_1|$ and $|F_2|$, respectively. Black points correspond to observed times when using fragment length 9. The surface depicts the expectation of Equation (3.2), with $n = 9$.

Computation of an element (i, j) of the distance matrix involves a constant and a linear term in fragment length, as can be seen in Figure 26. This is due to the calculation comprising three steps: calculation of the scaled interfragment covariance matrix (number of operations: linear in n); calculation of the singular values (number of operations: constant); and calculation of the Procrustes score (number of operations: constant).

Since the variance of t depends on $|F_1||F_2|$, parameters were estimated by performing simple linear regression on the ratio of these two terms, resulting in the model:

$$t = |F_1||F_2|\,(\alpha + \beta n + \varepsilon) \tag{3.2}$$

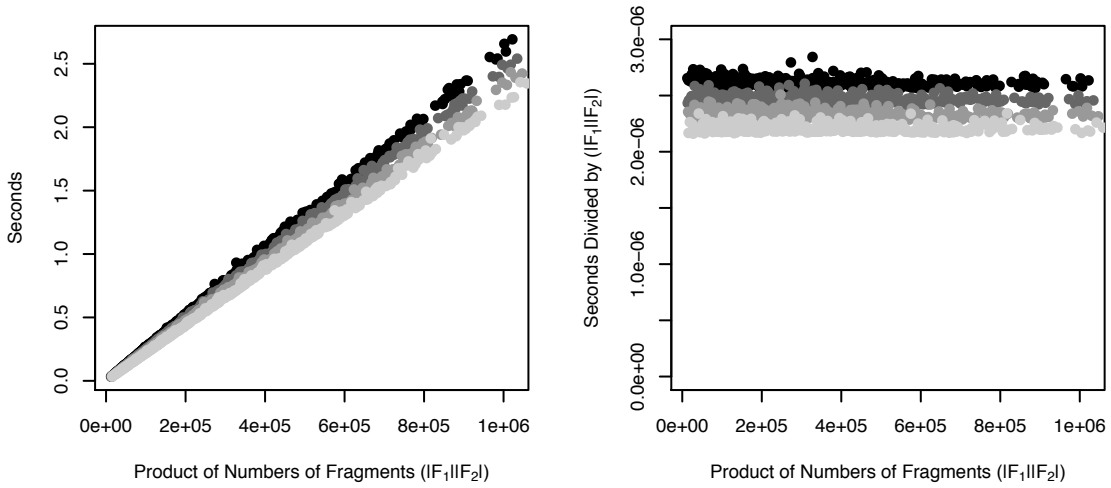


Figure 26: Relationship between computation time (seconds) and the product of the numbers of fragments in chains 1 and 2, $|F_1||F_2|$. Computation time is show in the graph on the left; ratio of time and $|F_1||F_2|$ is shown on the right. Results with various fragment lengths are shown, depicted by varying greyscale intensities. For clarity, results are shown for $n = 3, 7, 11, 15$, with darker intensity indicating higher fragment length.

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 2.104 \times 10^{-6}$, $\beta = 3.353 \times 10^{-8}$, and $\sigma = 3.387 \times 10^{-8}$, with $R^2 = 0.9995$. To give a loose interpretation of parameters, ignoring other factors, α represents the average time taken to calculate the Procrustes score (including calculation of the singular value decomposition of a 3×3 matrix), and β represents the average time taken to calculate the scaled covariance matrix per residue (comprising four atoms) in a fragment.

Dynamic Programming Algorithm

The dynamic programming algorithm includes calculation of the gap penalty matrix, the cost matrix, backward tracing to find the optimal path, and alignment filtering to ensure a one-to-one fragment correspondence. This suggests that computation time should have linear and quadratic dependencies on the numbers of fragments in the chains. Calculation of the vectors of helix scores $\vec{D}_i^{\text{helix}}$ (see §2.2.1), required in order to construct the gap penalty matrix, takes time proportional to $n|F_i|$. This can be seen in Figure 27.

Noting that the variance of t was found to depend on the average of the numbers of fragments, computation time is modelled:

$$t = \alpha + \beta|F_1||F_2| + (|F_1| + |F_2|)(\gamma + \delta n + \varepsilon) \tag{3.3}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 7.972 \times 10^{-5}$, $\beta = 7.407 \times 10^{-9}$, $\gamma = 3.019 \times 10^{-6}$, $\delta = 1.951 \times 10^{-7}$, and $\sigma = 1.618 \times 10^{-7}$, with $R^2 = 0.9971$. The model for $n = 9$ is shown in Figure 28.

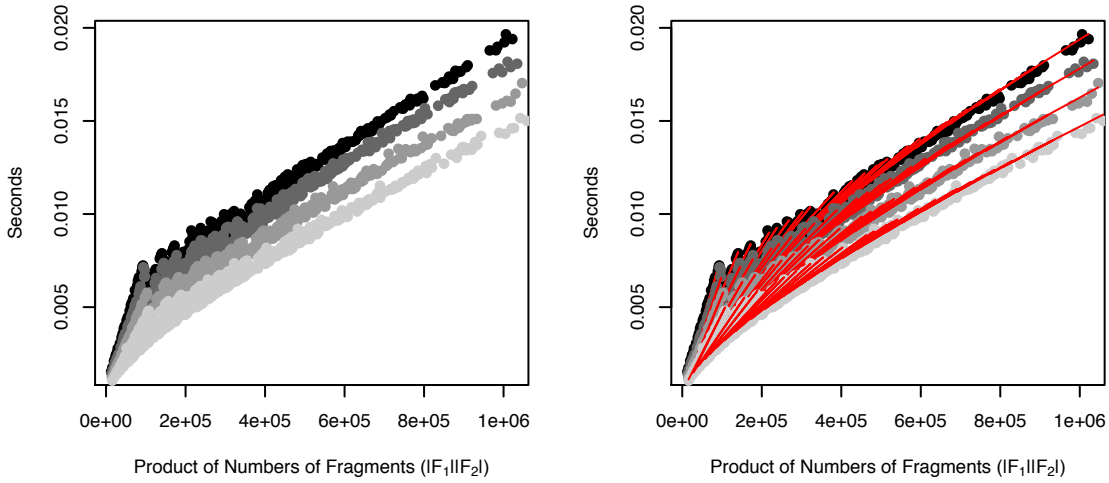


Figure 27: Relationship between computation time of the dynamic programming stage and the product of the numbers of fragments in chains 1 and 2. Results with various fragment lengths are shown on the left, depicted by varying greyscale intensities. Results are shown for $n = 3, 7, 11, 15$, with darker intensity indicating higher fragment length. On the right, red lines represent the model according to the expectation of Equation (3.3).

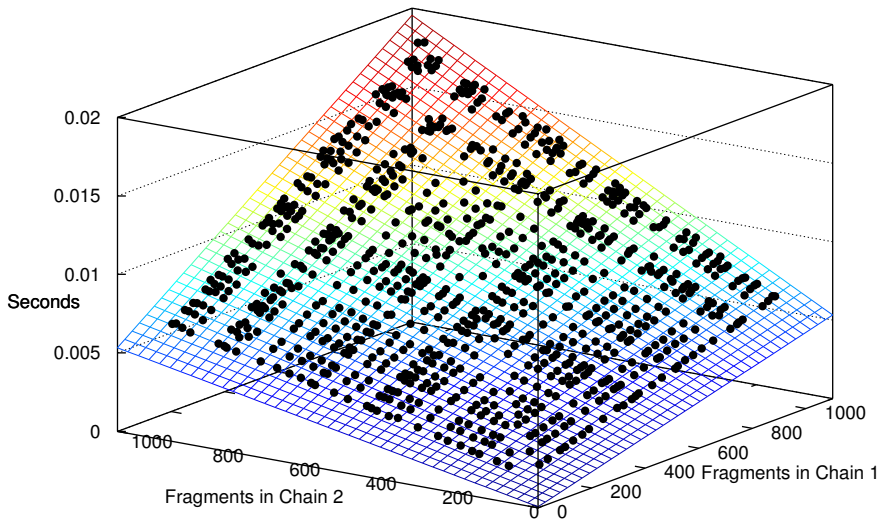


Figure 28: Computation time of the dynamic programming stage against the numbers of fragments in chains 1 and 2, $|F_1|$ and $|F_2|$, respectively. Black points correspond to observed times when using fragment length 9. The surface depicts the expectation of Equation (3.3), with $n = 9$.

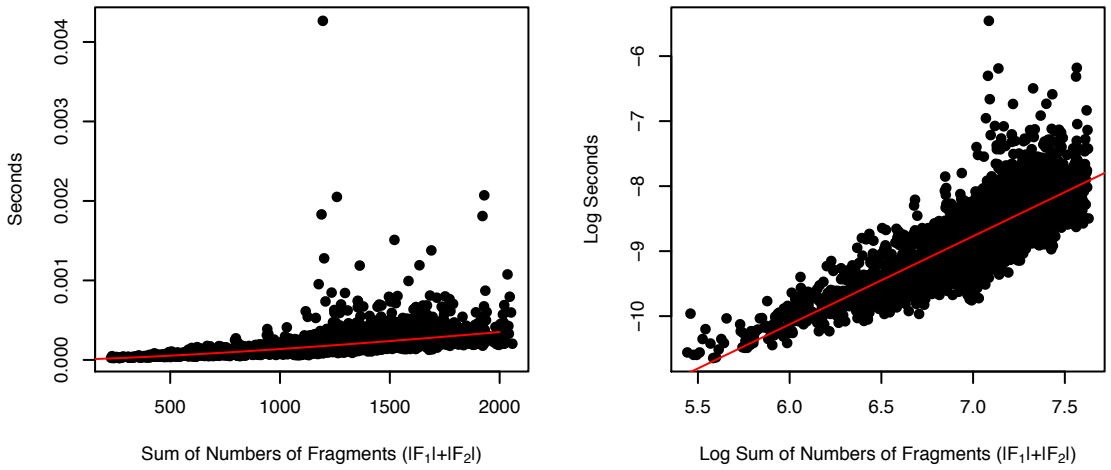


Figure 29: Relationship between computation time of the segment-based alignment refinement stage and the sum of the numbers of fragments in chains 1 and 2. A log-log plot is shown on the right. Black dots correspond to observed times, for all trialled fragment lengths. Red lines represent the model according to the expectation of Equation (3.4).

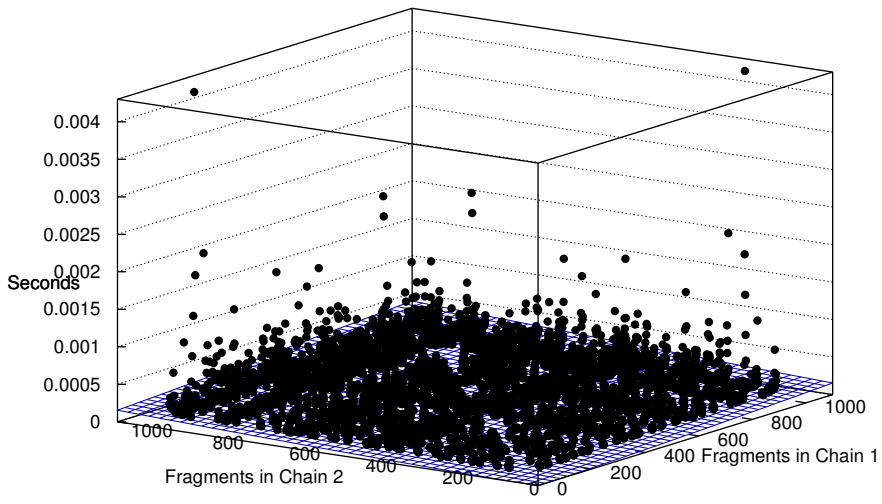


Figure 30: Computation time of the segment-based alignment refinement stage against the numbers of fragments in chains 1 and 2. Black points correspond to observed times, for all trialled fragment lengths. The surface depicts the expectation of Equation (3.4).

Segment-Based Alignment Refinement

The segment-based alignment refinement stage involves conversion from residue-based to segment-based storage format, iterative alignment refinement, and conversion back to the residue-based format.

Due to the heterogenous nature of the iterative refinement process, it is not possible to analytically determine the order of complexity with a reasonable degree of confidence. It is reasonable to surmise that the number of iterations, and the time it takes to perform each iteration, will depend largely on the particular chain pair being compared. Computation time was found to depend on the average of number of fragments (see Figure 29), and was modelled assuming a power relation:

$$t = (|F_1| + |F_2|)^{\alpha}\, e^{\beta+\varepsilon} \tag{3.4}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 1.356$, $\beta = -18.26$, and $\sigma = 0.3524$, with $R^2 = 0.3008$ (the log-log linear regression model had $R^2 = 0.6882$).

As can be seen in Figure 30, the errors are large compared with the fitted values, as expected. This clarifies that computation time is more dependent on structural properties of the particular chain pair than on the numbers of fragments in the chains. However, even the slowest trials are orders of magnitude faster than some of the other stages, meaning that this lack of determinism should not dramatically impact overall computation time.

Residue-Based Alignment Optimisation

The residue-based alignment optimisation stage involves removing clashing aligned fragment-pairs, iterative alignment optimisation, and error checking. Again, due to the nature of the iterative optimisation process, no attempt is made to predict the order of complexity.

Due to the employed alignment maximisation criteria, alignment length is highly correlated with the length of the shorter of the two chains (see Figure 31). Consequently, computation time was modelled:

$$t = \min(|F_1|, |F_2|)^{\alpha+\beta n}\, e^{\gamma+\delta n+\varepsilon} \tag{3.5}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 1.404$, $\beta = 1.298 \times 10^{-2}$, $\gamma = -15.99$, $\delta = -3.851 \times 10^{-2}$, and $\sigma = 0.1554$, with $R^2 = 0.9495$ (the log-log linear regression model had $R^2 = 0.9738$). For the test dataset, the observed time complexity of the optimisation stage was better than quadratic, with order varying from approximately 1.4 to 1.6 for fragment lengths 3 to 15, as seen in Figure 32.

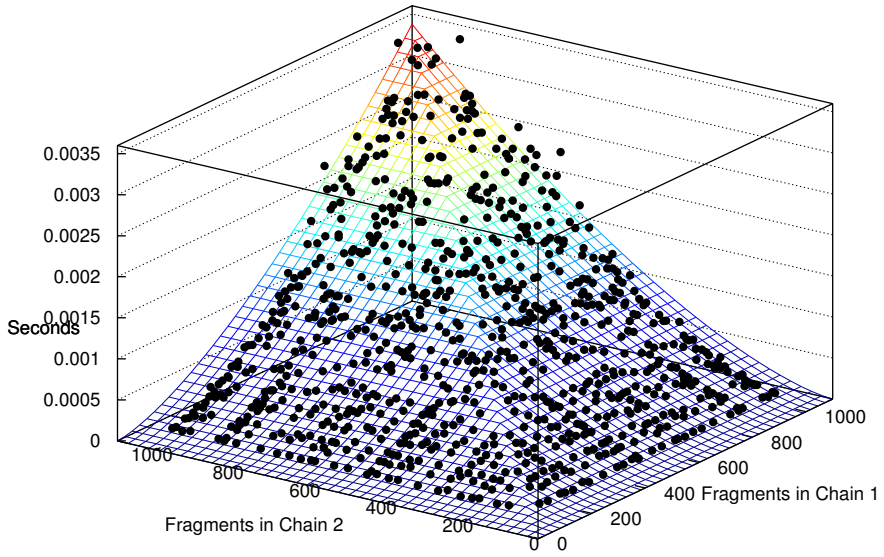


Figure 31: Computation time of the optimisation stage against the numbers of fragments in chains 1 and 2, $|F_1|$ and $|F_2|$, respectively. Black points correspond to observed times when using fragment length 9. The surface depicts the expectation of Equation (3.5), with $n = 9$.

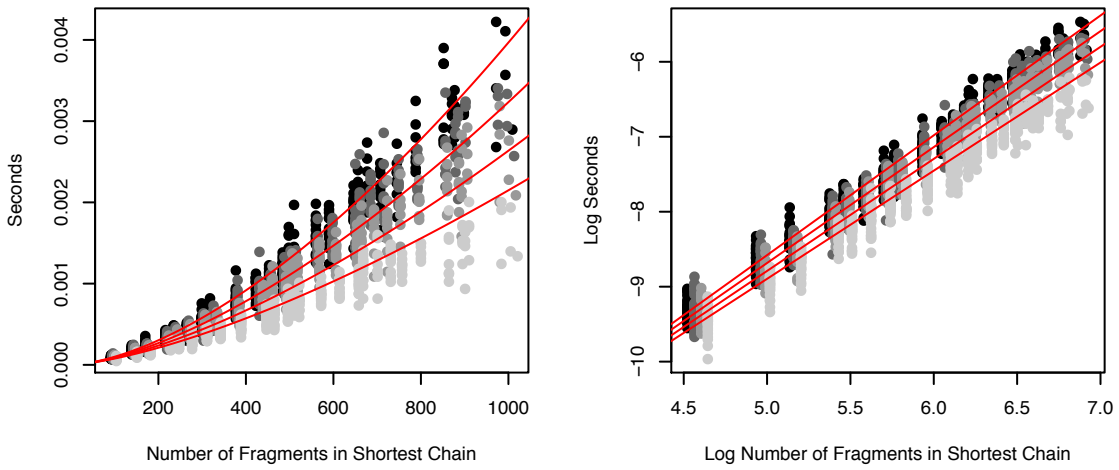


Figure 32: Relationship between computation time of the optimisation stage and the number of fragments in the shorter chain, with a log-log plot on the right. Results with various fragment lengths are shown, depicted by varying greyscale intensities. Results are shown for $n = 3, 7, 11, 15$, with darker intensity indicating higher fragment length. Red lines represent the model according to the expectation of Equation (3.5).

Alignment Scoring

The alignment scoring stage involves inferring the residue alignment from the fragment alignment, calculating the fragment-based central and intrafragment rotational dissimilarity scores, calculating the residue-based minimum score, and categorising residues according to the fragment library. These procedures are highly dependent on the alignment length, and are also dependent on the lengths of the two chains and the fragment length.

The length of the shorter of the two chains was used to represent the alignment length. Noting that the variance of t was found to depend on the length of the longer of the chains, computation time was modelled:

$$t = \alpha + \beta\left(|F_1| + |F_2|\right) + n\gamma \min\left(|F_1|, |F_2|\right) + \max\left(|F_1|, |F_2|\right)\varepsilon \tag{3.6}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 4.549 \times 10^{-4}$, $\beta = 1.355 \times 10^{-5}$, $\gamma = 1.068 \times 10^{-6}$, and $\sigma = 1.465 \times 10^{-6}$, with $R^2 = 0.9797$. The model for $n = 9$ is shown in Figure 33.

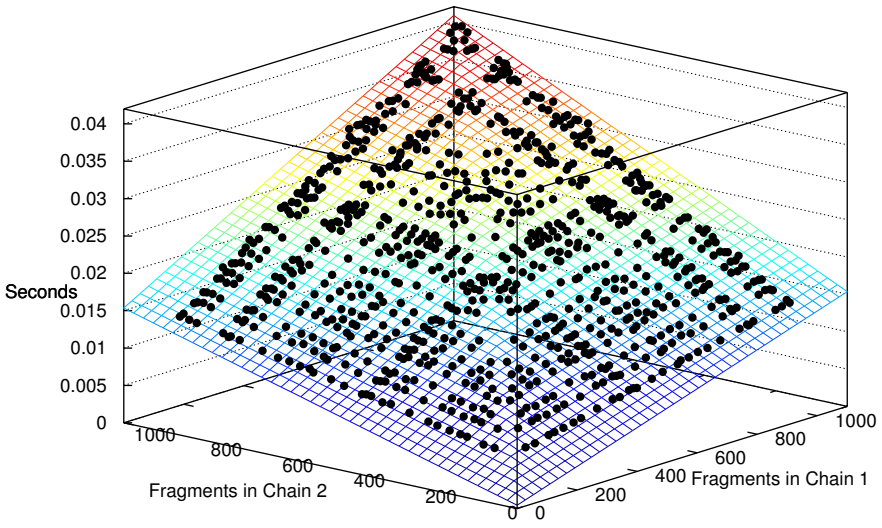


Figure 33: Computation time of the alignment scoring stage against the numbers of fragments in chains 1 and 2. Black points correspond to observed times, for fragment length 9. The surface depicts the expectation of Equation (3.6) with $n = 9$.

File Output

Output files include those comprising the global scores, the residue alignment (including residue-based scores), and a transformation file (which required calculation of global structure superposition). Superposed PDB-format files and *PyMOL* colour scripts were not written to file, since they are not output by default; a keyword must be specified in order to activate this functionality.

Computation time is linearly dependent on the alignment length, which is represented by the number of fragments in the shorter chain. Twelve extreme outliers were observed, as can be seen in Figure 34. These were largely attributed to non-deterministic errors; it was assumed that the increased computation times were largely due to heterogeneities in system resource allocation and performance (surmisably mainly attributed to the hard disk). Consequently, these outliers were removed for the estimation of model parameters, using an arbitrarily-selected threshold of 0.005 seconds, resulting in the model:

$$t = \alpha + \beta \min\left(|F_1|, |F_2|\right) + \varepsilon \tag{3.7}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 3.160 \times 10^{-4}$, $\beta = 3.950 \times 10^{-6}$, and $\sigma = 1.125 \times 10^{-4}$, with $R^2 = 0.9845$. The model is shown in Figures 34 and 35.

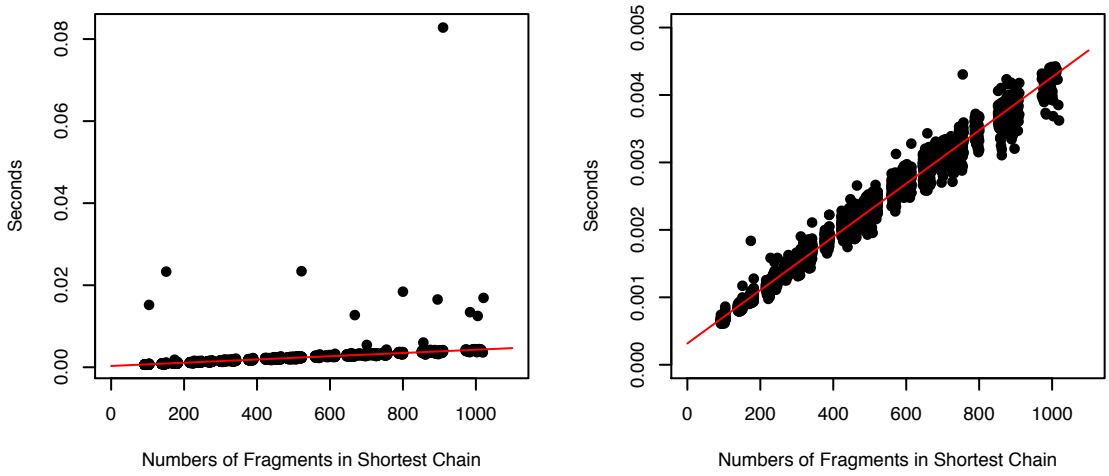


Figure 34: Relationship between computation time of the file output stage and the number of fragments in the shorter chain. Both graphs show the same data; extreme outliers are visible in the graph on the left, but not in the graph on the right (representing removal of outliers). Red lines represent the model according to the expectation of Equation (3.7).

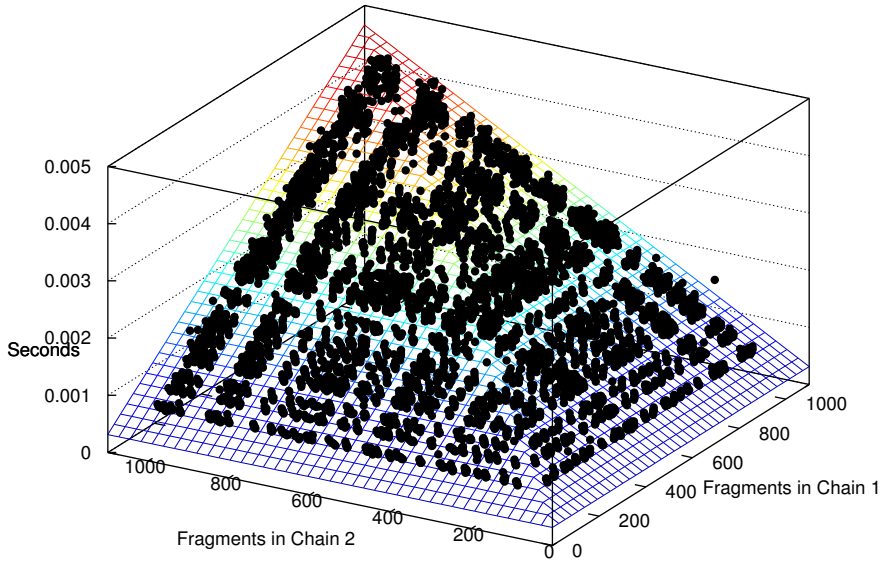


Figure 35: Computation time of the file output stage against the numbers of fragments in chains 1 and 2. Black points correspond to observed times. The surface corresponds to the expectation of Equation (3.7).

Identification of Rigid Substructures

The identification of rigid substructures involves: fragment alignment filtering (by Procrustes and intrafragment rotation scores), calculation of interchain fragment rotations, calculation of pairwise differential rotations, performing single linkage clustering, removal of small clusters, identification of preliminary cluster transformations, cluster rigidity filtering, identification of final cluster transformations, and calculation of fragment-based cluster scores.

This process involves algorithms that are linear in alignment length, quadratic in filtered alignment length, and linear in the number of identified clusters, given cluster sizes. Since the structures in the dataset are presumed dissimilar, the initial alignment filtering stage will tend to dramatically reduce the number of fragment-pairs considered. However, note that pairs of repetitive secondary structure fragments (particularly helices) might be expected to get through the initial filtering, despite potentially originating from unrelated structures. Consequently, given this heterogeneity, the order of the relationship between computation time and alignment length is not obvious, and extreme outliers may be expected. Subsequent stages will filter the alignment further, and no clusters would be expected to be identified in this case.

Given that constant filtering parameters were maintained for all tests (Procrustes score ≤ 1.0; intrafragment rotation angle $\leq 15°$), the size of the filtered alignment will depend highly on fragment length, since Procrustes score correlates with fragment length. Given the same parameters, increasing fragment length results in fewer aligned fragment pairs passing the initial filtering. This means that on average the maximum computational expense, and maximum variability, would be

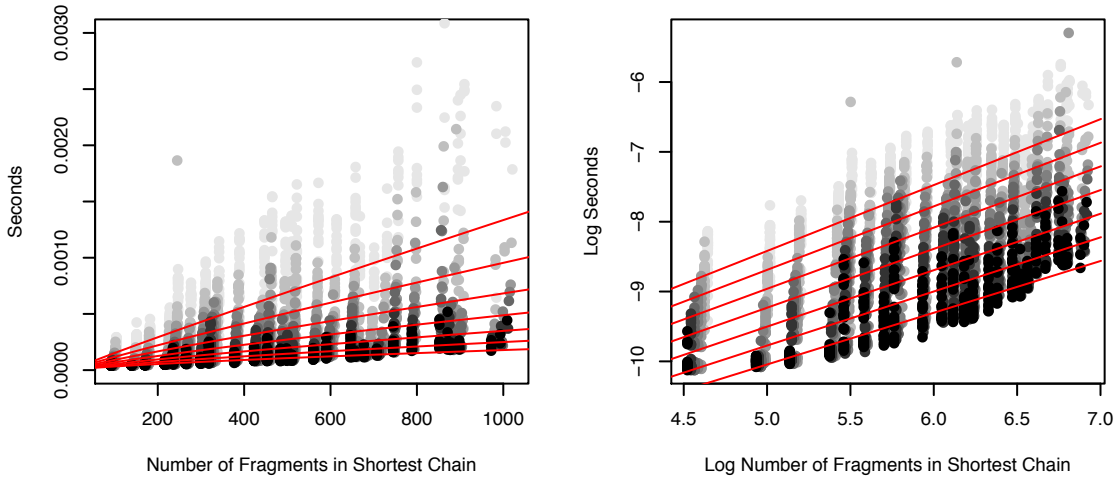


Figure 36: Relationship between computation time of the rigid substructure identification stage and the number of fragments in the shortest chain. A log-log plot is shown on the right. Results with various fragment lengths are shown, depicted by varying greyscale intensities. Results are shown for $n = 3, 5, 7, 9, 11, 13, 15$, with darker intensity indicating higher fragment length. Red lines represent the model according to the expectation of Equation (3.8).

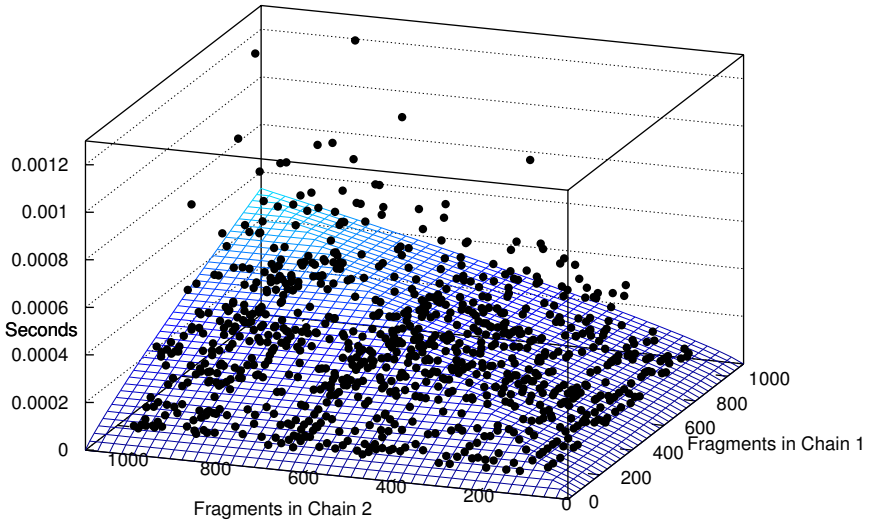


Figure 37: Computation time of the rigid substructure identification stage against the numbers of fragments in chains 1 and 2, for fragment length 9. Black points correspond to observed times. The surface corresponds to the expectation of Equation (3.8), with $n = 9$.

realised at the minimum fragment length ($n = 3$), as seen in Figure 36. Computation time was modelled:

$$t = \min\left(|F_1| + |F_2|\right)^{\alpha+\beta n} e^{\gamma+\delta n+\varepsilon} \tag{3.8}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 0.9942$, $\beta = -1.689 \times 10^{-2}$, $\gamma = -12.98$, $\delta = -5.102 \times 10^{-2}$, and $\sigma = 0.4889$, with $R^2 = 0.6716$ (the log-log linear regression model had $R^2 = 0.7303$). The model is shown in Figure 37. Note that these results will not reflect the computation time for structure-pairs with a sufficient degree of similarity. Rather, these results are of more relevance to prospective large database scanning, in which the majority of structure pairs would be dissimilar. In such 'null' cases, the time complexity has been observed to be on average faster than linear in minimum chain length (order varying from 0.94 to 0.74 for fragment lengths 3 to 15).

3.2.2 Alternative Methods of Fragment Scoring

There are three methods of scoring implemented in *ProSMART*. The first method (the default) calculates the Procrustes score using the four mainchain atoms from each residue, resulting in a matrix of $4n$ atomic coordinates. The second method uses only the C^α atoms, resulting in a matrix of just n atoms. Whilst this method utilises less information, it should be faster. Specifically, the calculation of the scaled covariance matrices should be four times faster than the default method, resulting in reduce relative performance loss at higher fragment lengths. The third method takes a hybrid approach, using all mainchain atoms for calculation of the rotation required to superpose the fragments, but then using only the C^α atoms for calculation of the score. Each of these methods uses Procrustes analysis for the calculation of the RMSD score, under the premise that this approach should be faster than physically superposing the fragments and calculating the RMSD in the traditional fashion. It is appropriate to test this presumption.

The parameters α and β of the model for distance matrix calculation (Equation (3.2)), given by $t = |F_1||F_2|\left(\alpha + \beta n + \varepsilon\right)$, were estimated for each of four approaches. These were: default method ($\alpha = 2.105 \times 10^{-6}$, $\beta = 3.375 \times 10^{-8}$); the second method ($\alpha = 2.120 \times 10^{-6}$, $\beta = 5.063 \times 10^{-9}$); the third method ($\alpha = 3.121 \times 10^{-6}$, $\beta = 9.133 \times 10^{-8}$); and traditional RMSD calculation using all main chain atoms ($\alpha = 3.168 \times 10^{-6}$, $\beta = 2.491 \times 10^{-7}$). As expected, the average coefficient of $|F_1||F_2|$ was very similar for the first and second methods, the second having a weaker response to fragment length. The third and fourth methods had much higher coefficients, indicating increased computational expense. As would be expected, the fourth method (traditional calculation of the RMSD) had a much stronger response to n, due to the number of operations actually being quadratic in fragment length (note that a linear model was not appropriate, but was nevertheless used so that parameter values were comparable).

Figure 38 displays the performance of the four considered approaches for the default fragment length 9. The second method performs slightly faster than the default, whilst the third method is

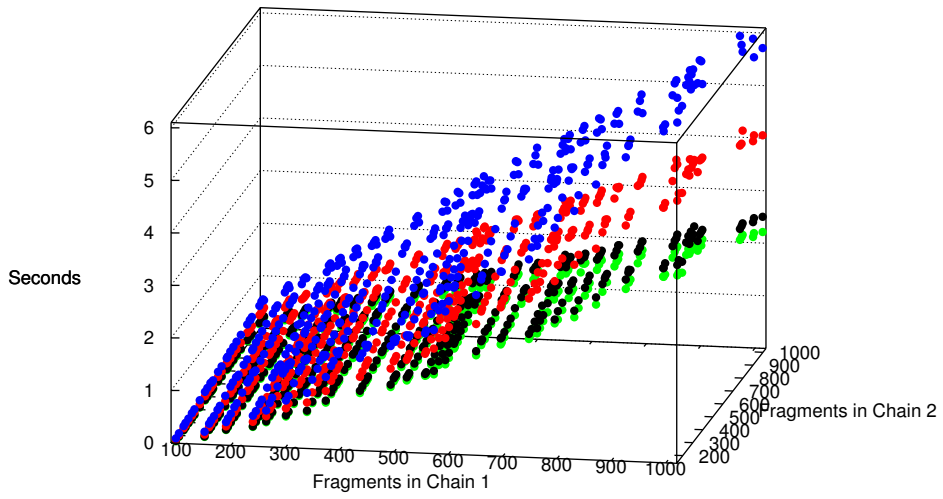


Figure 38: Computation time of the fragment distance matrix against the numbers of fragments in the two chains, for fragment length 9. Black points correspond to the default scoring method, green points to the second method, red points to the third method, and blue points to calculation of RMSD by physically superposing coordinates.

noticeably slower, as expected. Calculation of the RMSD without using Procrustes analysis was found to be noticeably slower than the implemented methods, taking on average 2.4 times longer (for $n = 9$) than the default Procrustes method, for which its results are theoretically equivalent.

3.2.3 Overall Speed of ProSMART ALIGN

Having determined the empirical, and where possible theoretical, orders of average-case complexity of the more notable components of *ProSMART ALIGN*, it is now possible to determine the overall algorithmic complexity of the program. By considering the individual components, we are able to understand where terms in the final model originate from, and to identify which procedures incur the most computational expense. This allows analysis of the implemented techniques, and also the identification of candidate algorithms for further computational optimisation in future. At the same time, we are able to identify which algorithms are sufficiently fast, indicating that further optimisation would not be overly beneficial. Note that we represent chain length using the number of fragments, due to the strong correlation with number of residues ($R^2 = 0.9998$ for test dataset).

Consideration of the separate components indicates average-case complexity (for dissimilar structures) to be quadratic in chain length and linear in fragment length $O(|F_1||F_2|n)$, this order being present in calculation of the distance matrix and the dynamic programming algorithm. The file input and alignment scoring stages are linear in fragment length. File input, alignment

scoring and file output are linear in chain length. Non-integer power relationships between time and chain length were observed for segment-based refinement, residue-based optimisation, and rigid substructure identification. The observed power relationships were better than quadratic in the considered fragment length range, and better than linear in the case of rigid substructure identification. Being non-deterministic due to the iterative heuristics involved, the computation time of these three stages is very variable. However, for dissimilar structures, this increased variability does not impact on overall computation time, since these stages have been observed to be orders of magnitude faster than some of the other components. Computation times of the other five components were found to be highly deterministic.

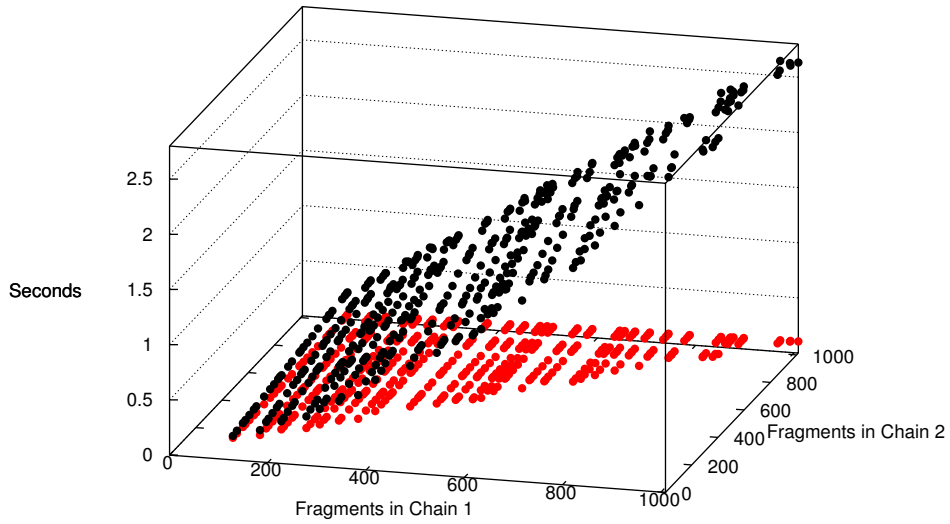
Program Decomposition

Computation time of *ProSMART ALIGN* is dominated by calculation of the distance matrix, especially at higher chain lengths, as seen in Figure 39. This preliminarily suggests the program to be $O(|F_1||F_2|n)$ on average. Distance matrix calculation is far more computationally expensive than the sum of other components, for all considered fragment lengths. Increasing fragment length causes noticeably increased distance matrix computation time, whilst at a glance seemingly not dramatically affecting the sum of other components.

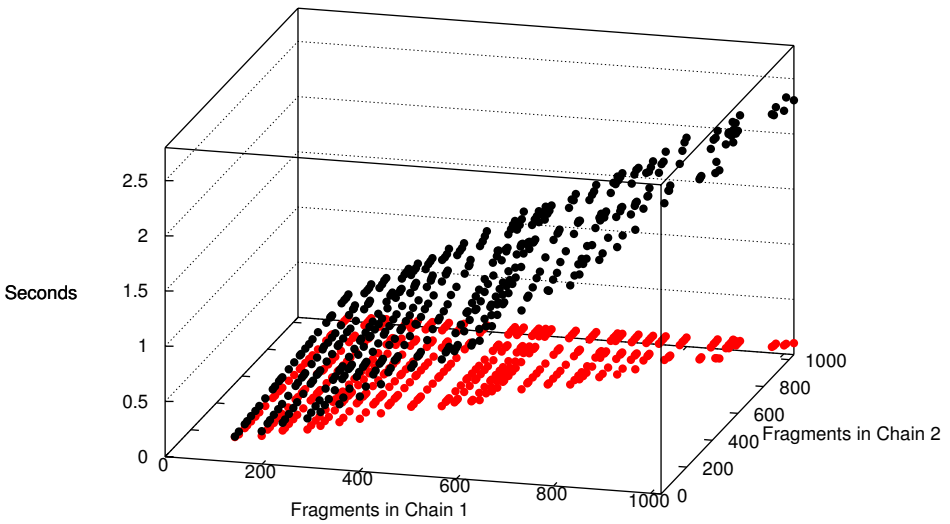
The computation times of the segment-based refinement, residue-based optimisation and rigid substructure identification algorithms were found to be negligible in comparison with the other algorithms, being orders of magnitude faster than total computation time. Consequently, these processes were excluded from any further analysis. This suggests the successful development of fast heuristic algorithms for refinement, optimisation and rotation clustering, at least in terms of speed. However, it is important to note that these algorithms may prove more computationally significant when executed on similar structures, in accordance with their design. Nevertheless, the achievement of negligible computational expense in this context is relevant and important for large database scanning where the majority of compared chain-pairs will be dissimilar.

Figure 40 displays the proportion of overall computation time attributed to distance matrix calculation. This component strongly dominates relative computation time at all chain lengths, but especially for higher chain lengths. When the length of the shorter chain is small, the effect of other components becomes more important; this is expected, given their comparative orders of complexity. Changing fragment length did not noticeably change the qualitative nor quantitative nature of this effect (data not shown).

Figure 41 displays the proportion of overall computation time attributed to file output, dynamic programming, alignment scoring, and file output. The most computationally expensive of these were file input and alignment scoring, with file output being almost negligible. It is important to mention that outputting the non-default PDB files and *PyMOL* colour scripts is computationally expensive, and would result in the file output stage taking time comparable to that of the file input stage



(a) Computation time using fragment length $n = 15$.



(b) Computation time using fragment length $n = 3$.

Figure 39: Relationship between computation time and the number of fragments in the two chains. Black dots represent computation time of the distance matrix; red dots represent total computation time of everything else in *ProSMART ALIGN*. Subfigure (a) corresponds to fragment length 15, and subfigure (b) to fragment length 3.

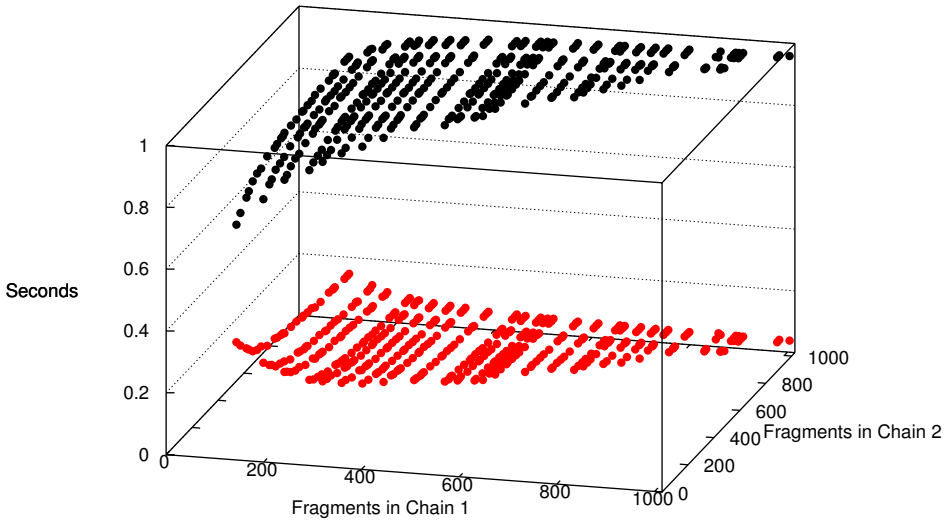


Figure 40: Black dots represent the ratio between computation time of calculating the distance matrix and overall time of *ProSMART ALIGN*. Red dots represent the complement (sum of all other processes). Data corresponding to fragment length 9 are shown.

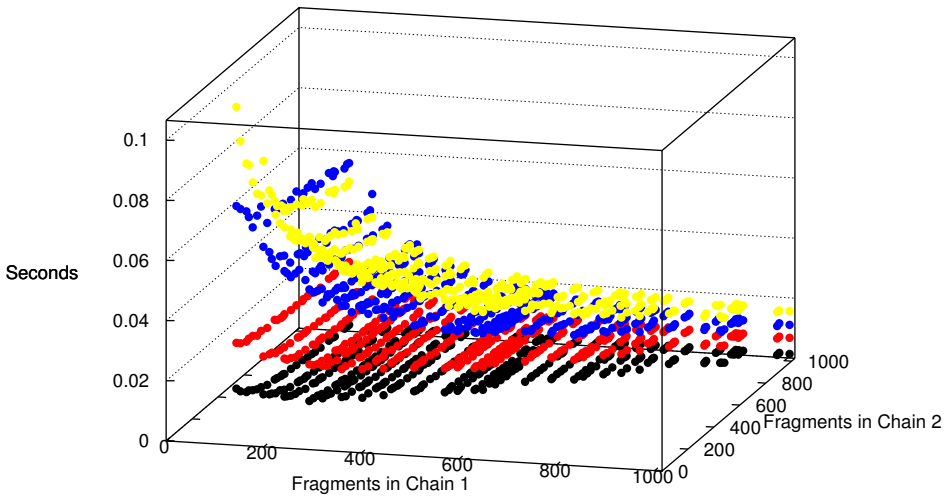


Figure 41: Ratio between computation time of calculating individual components and overall time of *ProSMART ALIGN*, for fragment length 9. Black dots correspond to file output, red to dynamic programming, blue to file input, and yellow to alignment scoring.

(data not shown). Again, varying fragment length had little impact on observed trends (data not shown).

We conclude that calculation of the distance matrix is the most computationally expensive component of *ProSMART ALIGN*, and thus further optimisation of this algorithm would result in the greatest performance benefit. Any optimisation of other components would be of comparably little benefit, at least for dissimilar structures. However, the greatest relative improvement would be observed when the shorter of the two chains has a small chain length, e.g. less than around 200 residues.

Computational Complexity

By modelling the individual components, it has been possible to gain insight regarding the sources and functional forms of computational expense. It is now possible to use this knowledge of expected model terms in order to model the net runtime of *ProSMART ALIGN*, and thus identify the average order of complexity observed in practice.

Total runtime was calculated externally, from launch to termination of the process. Consequently, the observed times include all components considered above, as well as other sources of computational expense, such as launching and terminating the process.

The chosen model for overall runtime included all terms from the distance matrix model, including the form of error structure. In addition, terms were inherited from the models for file input, dynamic programming, and alignment scoring. For parsimony through simplicity and maintenance of parameter interpretation, other terms (including those from the file output model) were not used; the inclusion of extra terms did not dramatically improve the model. All included parameters were highly significant, and their values agree (within reason) with those observed in the source individual component models, thus they maintain sensible interpretation. Total runtime was modelled:

$$t = \alpha + \beta n \min\left(|F_1|, |F_2|\right) + \left(|F_1| + |F_2|\right)\left(\gamma + \delta n\right) + |F_1||F_2|\left(\epsilon + \zeta n + \varepsilon\right) \tag{3.9}$$

where $\varepsilon \sim N(0, \sigma^2)$, and parameters were estimated as $\alpha = 7.599 \times 10^{-3}$, $\beta = 1.075 \times 10^{-6}$, $\gamma = 2.618 \times 10^{-5}$, $\delta = 3.319 \times 10^{-7}$, $\epsilon = 2.130 \times 10^{-6}$, $\zeta = 3.272 \times 10^{-8}$, and $\sigma = 3.732 \times 10^{-8}$, with $R^2 = 0.9995$. This model suggests that runtime is highly deterministic, at least for dissimilar structures such as those in the dataset. As expected, the observed average complexity of total runtime is $O(|F_1||F_2|n)$. Due to the strong correlation between number of fragments and number of residues in a chain ($R^2 = 0.9998$ for test dataset), it may be deduced that *ProSMART ALIGN* is on average $O(N_1 N_2 n)$. The model for $n = 9$ is shown in Figures 42 and 43.

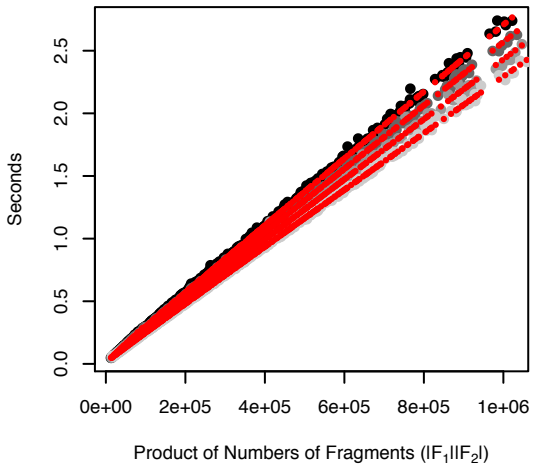


Figure 42: Computation time of *ProSMART ALIGN* against the product of the numbers of fragments in chains 1 and 2. Results with various fragment lengths are shown, depicted by varying greyscale intensities. Results are shown for $n = 3, 7, 11, 15$, with darker intensity indicating higher fragment length. Red dots represent fitted values of the model according to the expectation of Equation (3.9).

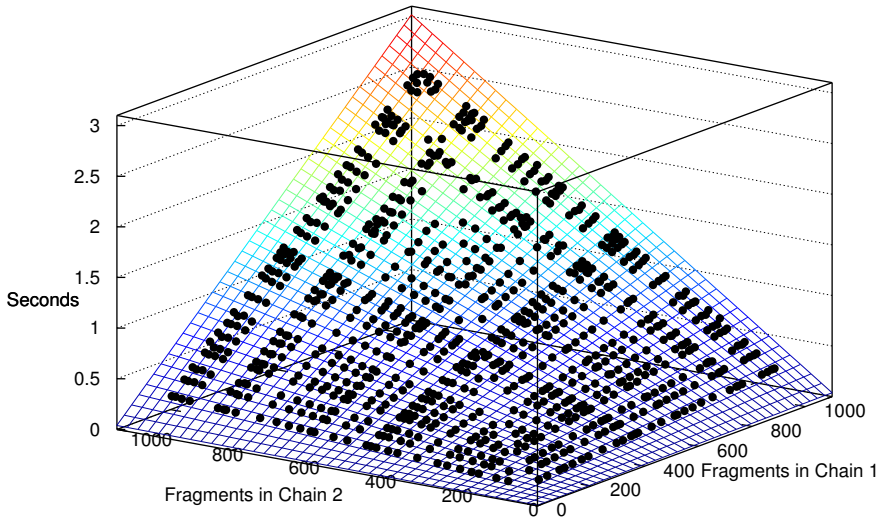


Figure 43: Computation time of *ProSMART ALIGN* against the numbers of fragments in chains 1 and 2. Black points correspond to observed times when using fragment length 9. The surface depicts the expectation of Equation (3.9), with $n = 9$.

3.2.4 Primary Source of Computational Expense

Since the calculation of the Procrustes distance matrix is by far the most computationally expensive component, it is of interest to identify the major sources of this expense. Such sources would be prime targets for any future optimisation, should performance become of major import.

Calculation of the distance matrix involves two steps: calculation of the trace of the scaled self-covariance matrices, which takes linear time $O(|F_1| + |F_2|)$; and calculation of the matrix elements, which is quadratic $O(|F_1||F_2|)$. The quadratic component involves three calculations (for each element): the scaled interfragment covariance matrix; the singular value decomposition (SVD); and finally the Procrustes score.

The linear component was found to be negligible, being orders of magnitude faster than total computation time (maximum computation time for $n = 15$ was approximately 6×10^{-3} seconds). Of the quadratic time components, calculation of the SVD was by far the most expensive; calculation of the scaled covariance matrices was secondary, whilst calculation of the Procrustes score was almost negligible. These are shown in Figure 44, for fragment length 9.

As expected, calculation of the scaled covariance matrices was strongly linearly dependent on fragment length, whilst calculation of the SVD was not, as can be seen in Figure 45. We may conclude that calculation of the matrix of SVDs is $O(|F_1||F_2|)$, and is the most computationally expensive component of *ProSMART ALIGN*, at least in the considered fragment length range (although note that a single SVD calculation took approximately 2×10^{-6} seconds on average).

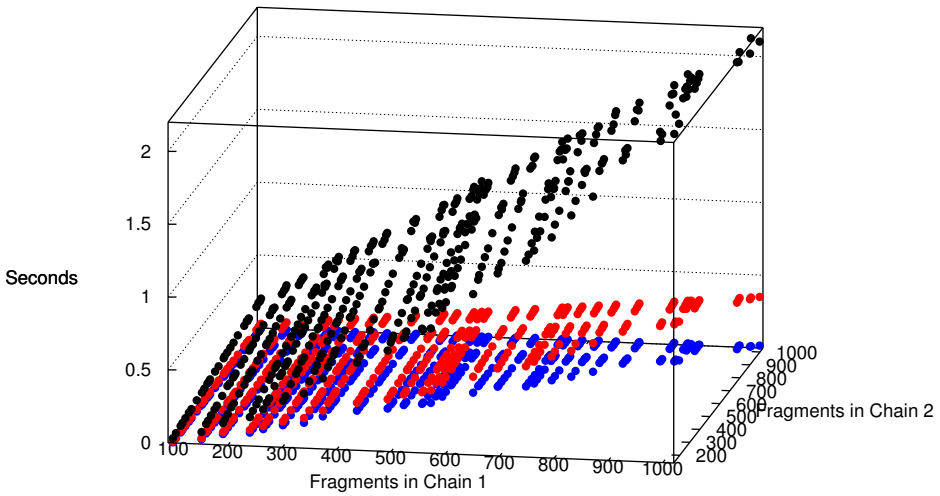


Figure 44: Sum of computation times of the various components involved in the calculation of elements of the distance matrix against the numbers of fragments in chains 1 and 2, when using fragment length 9. Red points correspond to calculation of the scaled covariance matrices, black points to the calculation of the singular value decompositions, and blue points to the consequent calculation of the Procrustes scores.

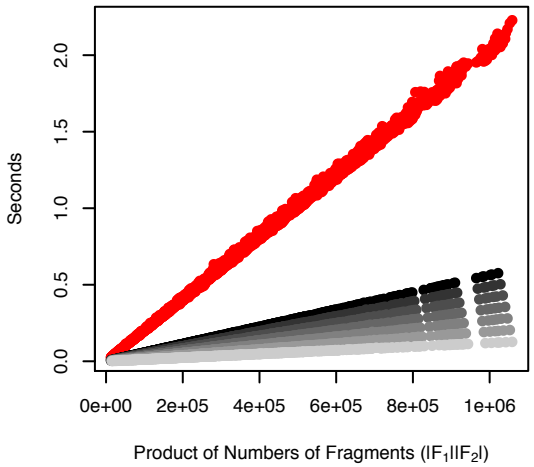


Figure 45: Sum of computation times of the major components involved in the calculation of elements of the distance matrix against the product of numbers of fragments in the two chains. Points corresponding to the calculation of the scaled covariance matrices are shown in greyscale for fragment lengths $n = 3, 5, 7, 9, 11, 13, 15$, with darker intensity indicating higher fragment length. Red points correspond to the calculation of the singular value decompositions of the trace of the matrix products (for all fragment lengths).

Calculation of the scaled covariance matrices is the secondary source of computational expense, and is majorly responsible for the presence of the fragment length n in *ProSMART ALIGN*'s observed average time complexity $O(|F_1||F_2|n)$. Consequently, calculation of the SVD, and to a lesser degree the trace of the matrix products, should be prime targets for any future algorithmic optimisation, should performance improvement become a priority.

3.3 Examples Demonstrating Functionality of ProSMART ALIGN

Further to outputting text files containing residue-based alignment scores, *ProSMART* provides HTML-format global alignment scores, superposition results in the form of PDB files, and *PyMOL* (Schrödinger, LLC, 2010; DeLano, 2007) scripts that colour residues according to achieved scores. Most functionalities use an intuitive colour gradient (colours and gradients may be chosen) representing similarity/dissimilarity, offering a powerful way of visualising results.

3.3.1 Main Chain Scores

A unique residue alignment A_R may be directly inferred from the final fragment alignment:

$$A_R = \bigcup_{a_k \in A} \{(f_i + x, f_j + x) : (i, j) = a_k, x = 0 \dots n-1\} \tag{3.10}$$

where n is the number of residues in a fragment, using the notation of Chapter 2. We have knowledge of the Procrustes scores $\mathbf{D}_{a_k}$ corresponding to each of the aligned fragment-pairs $a_k \in A$. However, the fact that a particular residue may belong to multiple aligned fragments causes ambiguity in assigning a score to an aligned residue pair. We are able to exploit this, aiming to extract as much information as possible by considering multiple scores. These scores provide different, complementary, information about the structural similarity of a residue's local structural environment, which may be used in concert when performing a conformation-independent structural analysis. Specifically, these scores are termed:

- Central score;
- Minimum score;
- Intrafragment rotational dissimilarity score.

The central and minimum scores of an aligned residue-pair are directly inherited from fragment Procrustes scores. In contrast, the intrafragment rotational dissimilarity score describes backbone differential geometry.

Central and Minimum Scores

The central score measures the structural dissimilarity of the immediate local structural environment of a residue, whereas the minimum score corresponds to the best-scoring aligned fragment that the residue belongs to. The structural resolution, which is defined by the (odd) fragment length n, determines what is considered to be the local structural environment.

The central score of a residue is inherited from the fragment centred on that residue, and may be defined:

$$s_{\text{central}}(k) = \mathbf{D}_{ij} \qquad \text{s.t. } [A_R]_k = (f_i + \tfrac{n-1}{2}, f_j + \tfrac{n-1}{2}), (i,j) \in A \tag{3.11}$$

This represents an injective map between aligned fragments and residues; only residues located at the centre of an aligned fragment have a central score. Note that there is not always an aligned fragment-pair $(i,j) \in A$ that satisfies these conditions. Indeed, the first and last $\frac{n-1}{2}$ residue pairs in a segment of consecutively aligned residues do not have a central score. This is justified, since it is not sensible to assign residues a central score if the corresponding fragments are unaligned. Consequently, a central score is only assigned to residues that exist in the centre of an aligned fragment-pair.

The minimum score may be defined:

$$s_{\min}(k) = \min \{\mathbf{D}_{ij} : [A_R]_k = (f_i + x, f_j + x), (i,j) \in A, x = 0 \ldots n-1\} \tag{3.12}$$

for k in $1 \ldots |A_R|$. A residue may belong to up to n aligned fragments; a residue inherits its minimum score from the most favourable of these fragments. This score is often of most interest for

visualisation of residue-based structural comparisons of non-identical structures, and consequently might generally be considered *ProSMART*'s archetypal residue-based score. Unlike the central score, there is potential for every residue to be assigned a minimum score, depending on the particular fragment alignment.

The central score is a measure of the conservation of a residue's immediate structural environment, whereas the minimum score describes whether or not the residue belongs to a conserved region. For example, if a residue is on the cusp between a rigidly conserved region and a dissimilar region (or at least, a point where there is a conformational change), it would be expected for the minimum score to be very low (well-scoring) relative to the central score.

Note that the minimum score is inherently never greater than the central score:

$$s_{\min}(k) \leq s_{\text{central}}(k) \qquad \forall k = 1 \ldots |A_R|. \tag{3.13}$$

Consequently, the implication of similarity from the central score is a stronger condition than from the minimum score. Residues with a sufficiently low minimum score are expected to belong to conserved regions between the two structures. In contrast, those with a sufficiently low central score are expected to be embedded within conserved regions. This distinction may be useful when attempting to identify larger conserved substructures (e.g. domains) and regions that remain rigidly conserved during global conformational change.

Intrafragment Rotational Dissimilarity Score

Main chain conformational changes (e.g. hinging motions), such as those observed for different binding modes, cause increased local structural dissimilarity in similar structures. This will often cause some residues' central scores to increase. However, using the Procrustes score alone, there is no way to distinguish between residues involved in conformational change and those that merely have relatively dissimilar local environments. We may wish to ask whether the dissimilarity implied by an observed Procrustes score is due to inherent structural dissimilarity, random variations in atomic positions, or whether the compared fragments would be very similar but for the distortion of the main chain (e.g. consider longer hinge regions, where the main chain is gradually bent or twisted).

One particular situation that we want to be able to identify is when a particular residue is responsible for, or involved in, a relatively large conformational change. This can occur when local structure is well-conserved on both sides of the target residue but the fragment centred on the target residue has a relatively poor Procrustes score. This situation sometimes occurs when there is a point insertion or sharp hinge.

We aim to be able to deal with such situations by considering the 'intrafragment rotational dissimilarity' score, which represents the amount of conformational change within a fragment, about the central residue. This score exists for each residue that is central to an aligned fragment, similarly to the central score.

The intrafragment rotational dissimilarity score is achieved by splitting a fragment into two sections, and considering the difference between the rotations required to superpose these sections between the two structures. Specifically, an n-residue fragment from chain x is split into two sections, left and right, with corresponding main chain atomic coordinate matrices denoted $\mathbf{C}^{\mathbf{x}}_{\text{left}}$ and $\mathbf{C}^{\mathbf{x}}_{\text{right}}$. Each of these sections comprise $\frac{n-1}{2}$ residues, noting that the central residue is excluded. One of the scenarios we aim to identify is when the central residue is an effective singular point of dissimilarity in an otherwise similar alignment. Therefore, the central residue is excluded in an attempt to improve the superposition of the separate sections, and increase sensitivity to conformational change.

Changes in main chain conformation may be described by the transformational dissimilarity between adjacent aligned sections of structure. Due to chemical restraints restricting the possible conformations of local structure, we surmise that the translational component of this transformation does not provide much more information than can be achieved by considering the rotational component alone. Hence, in order to minimise the number of descriptors, we represent the transformational dissimilarity by considering only the rotational component.

For each aligned fragment pair $a_k \in A$, the intrafragment rotational dissimilarity score is given by the cosine distance:

$$s_{\text{rot}}(k) = 1 - \cos(\theta_k) \tag{3.14}$$

where θ_k represents main chain conformational change about the central residue of the k^{th} aligned fragment-pair. Specifically, θ_k is the angle of rotation required to optimally superpose the right fragment sections, relative to the superposition of the left sections, allowing for translational invariance. These procedural concepts and methods of scoring are very similar to that used for rigid substructure identification (see §2.3).

Note that θ_k is an Euler angle between rotation matrices in $SO(3)$, and does not correspond to a simple intuitive angle in Euclidean space. Rather than attempting to separate the rotational component into two descriptors, representing curvature and torsion, information regarding differential geometry of the backbone is captured by the single parameter θ_k. Consequently, the score $s_{\text{rot}}(k)$ is penalised by both twisting and bending of the main chain. This score is such that identical fragments score zero, with increasing scores indicating increasing conformational dissimilarity. The maximum possible value of the score is 2, although in practice high scores are very unlikely to be observed due to the effect of chemical restraints limiting potential fragment conformations.

The elements of a rotation matrix $\mathbf{R} \in SO(3)$ are related to the corresponding Euler angle θ by the identity (Shoemake, 1985):

$$\text{tr}\,(\mathbf{R}) = 1 + 2\cos(\theta) \tag{3.15}$$

Consequently, the intrafragment rotational dissimilarity score may be calculated in the more con-

venient form (compare with Equation (2.32)):

$$s_{\text{rot}}(k) = \frac{3 - \text{tr}\left(\mathbf{R}_{\text{left}}\mathbf{R}_{\text{right}}^{\mathbf{T}}\right)}{2} \tag{3.16}$$

where $\mathbf{R}_{\text{left}}$ and $\mathbf{R}_{\text{right}}$ are the rotation matrices that superpose the left and right sections of the aligned fragment pair, respectively, after the coordinates are transformed so that they are centred at zero, allowing for translational invariance. The matrix $\mathbf{R}_{\text{left}}\mathbf{R}_{\text{right}}^{\mathbf{T}} \in SO(3)$ represents the differential rotation between the two halves of the fragment. The rotation matrices $\mathbf{R}_{\text{left}}$ and $\mathbf{R}_{\text{right}}$ are calculated using the singular value decompositions of the mean-normalised covariance matrices $\hat{\mathbf{C}}_{\text{left}}^{\mathbf{2T}}\hat{\mathbf{C}}_{\text{left}}^{\mathbf{1}}$ and $\hat{\mathbf{C}}_{\text{right}}^{\mathbf{2T}}\hat{\mathbf{C}}_{\text{right}}^{\mathbf{1}}$, similarly to the procedure used for superposing fragments (see §2.1.4).

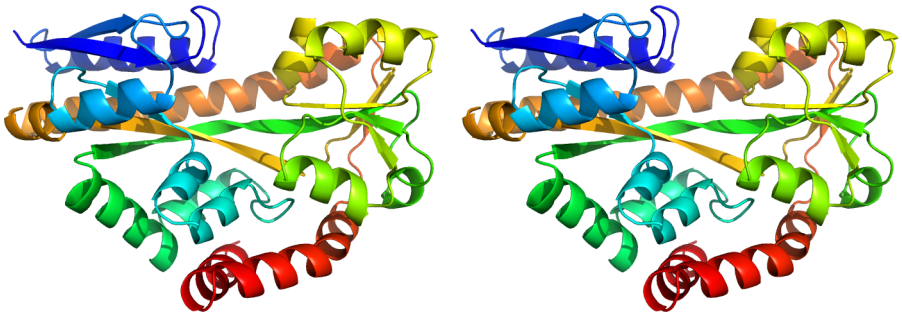
The score is approximately zero when there is no main chain conformational change about the fragment-pair's central residue, ignoring details about the local structural dissimilarity. However, the more similar the structures, the more robust the superposition. Consequently, the score is relatively insensitive to the Procrustes score. This means that this score provides complementary information to the minimum and central main chain scores. Considering all scores together allows for a greater knowledge and understanding of the dissimilarities in an aligned residue's local structural environment between two protein structures. In particular, the intrafragment rotational dissimilarity score allows for the easy identification of the residues that are responsible for, or involved in, main chain conformational change in similar structures. Such a change may or may not have a large impact on global conformation.

Since the calculation of this score involves the superposition of just $\frac{n-1}{2}$ residues, the robustness of the intrafragment rotational dissimilarity is largely dependent on the choice of fragment length n. The robustness of this score is also highly dependent on the similarity of the structures being compared; poor similarity can misleadingly lead to low or high scores, which have no sensible interpretation. Therefore, this score only makes sense if there is evidence to suggest that the surrounding local structures are sufficiently similar. This score should be used in conjunction with other scores, particularly the central score, in order to ensure a meaningful interpretation.

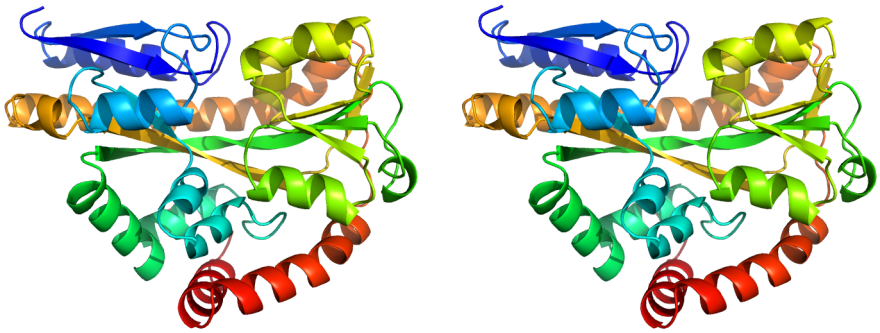
Examples

Further to tabular form, residue-based scores are communicated by colouring residues according to the corresponding score, allowing intuitive visualisation of results. In the current implementation, *ProSMART* achieves this by outputting *PyMOL* format colour scripts that automatically colour residues appropriately. This approach is deemed powerful, since it provides different and complementary information to a superposition alone.

Figures 46, 47, 48 and 49 show an example of the comparison of two sequence-identical chains adopting different global conformations. Consideration of the robust minimum score (Figure 47) immediately identifies that the structures are locally similar throughout the chain, indicating that

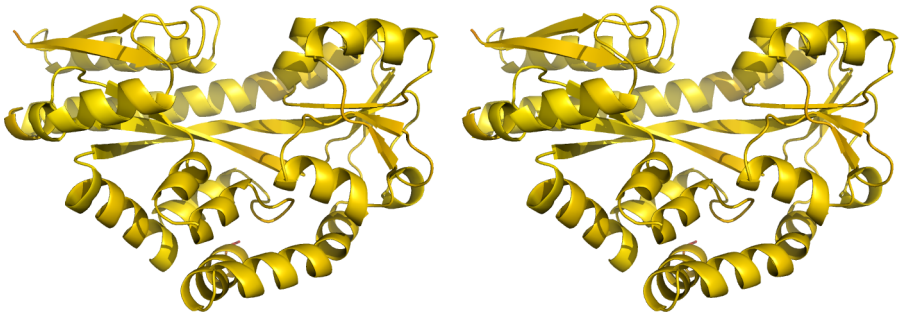


(a) Stereo view of 2cex(A).

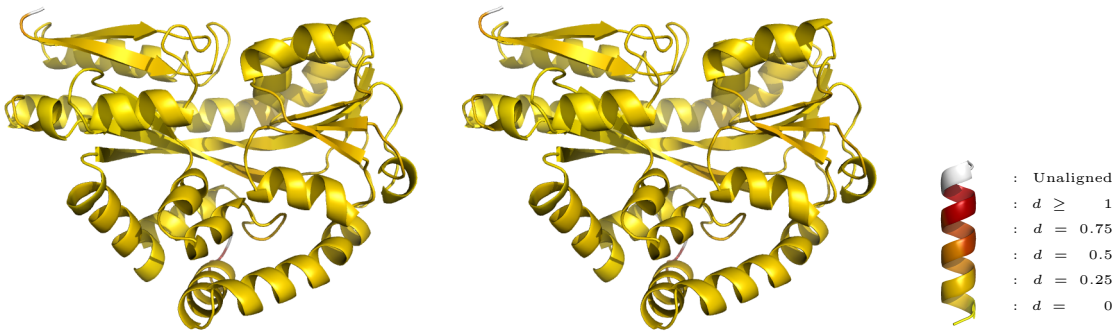


(b) Stereo view of 3b50(A).

Figure 46: Stereo views of two sequence-identical sialic acid binding protein structures (a) 2cex(A) (Muller et al., 2006) and (b) 3b50(A) (Johnston et al., 2008), which are unbound and bound forms, respectively. Structures are oriented so their coordinate frames correspond to the global superposition, and are shown rainbow-coloured along the chain from blue (N-termini) to red (C-termini).

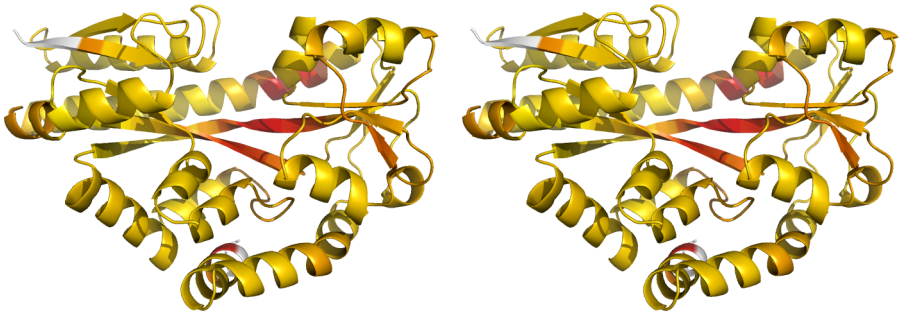


(a) Stereo view of 2cex(A).

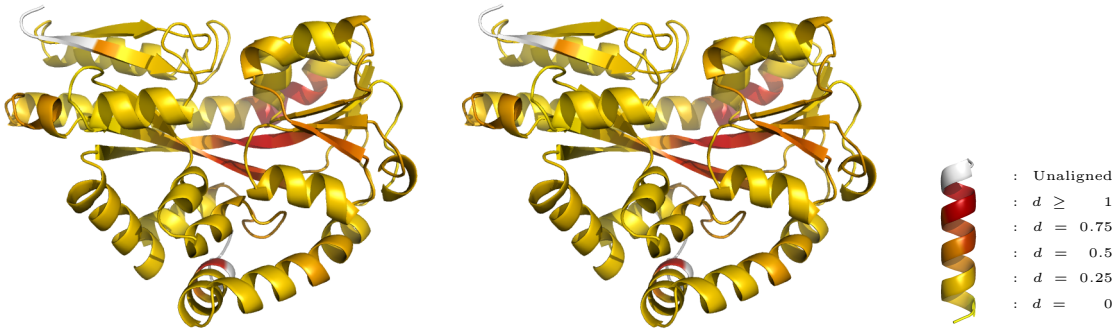


(b) Stereo view of 3b50(A).

Figure 47: Stereo views of two sequence-identical sialic acid binding protein structures 2cex(A) and 3b50(A), coloured by the minimum main chain scores arising from their comparison (with $n = 9$) .

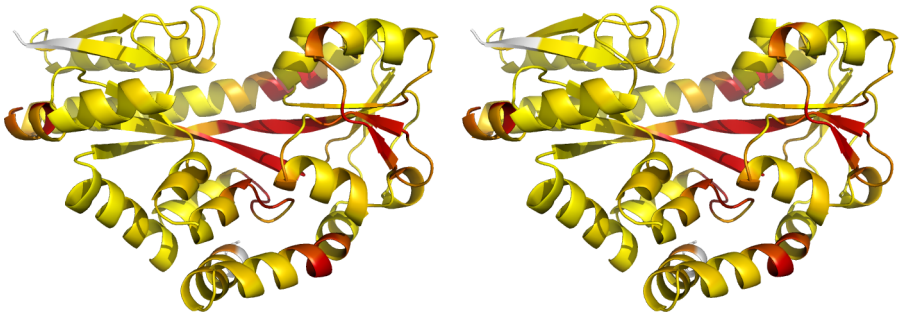


(a) Stereo view of 2cex(A).

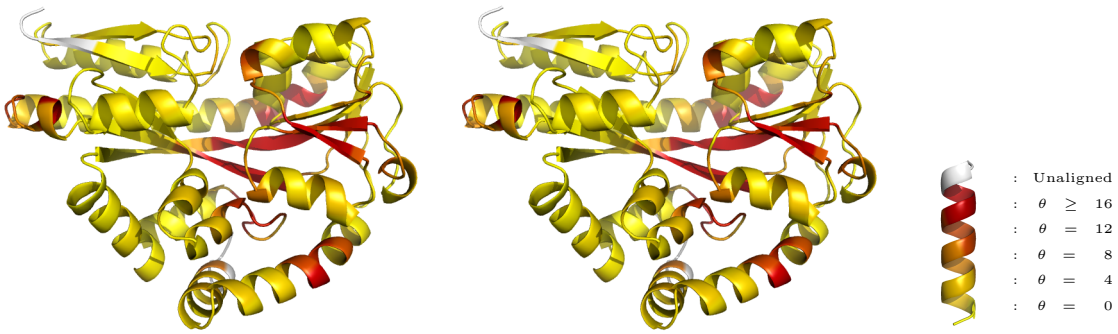


(b) Stereo view of 3b50(A).

Figure 48: Stereo views of two sequence-identical sialic acid binding protein structures 2cex(A) and 3b50(A), coloured by the central main chain scores arising from their comparison (with $n = 9$) .



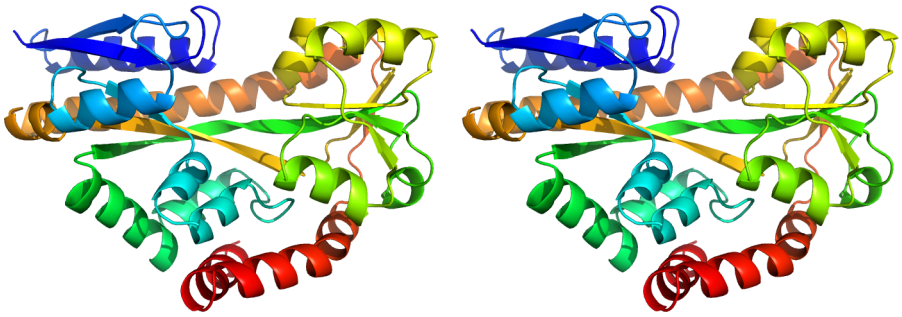
(a) Stereo view of 2cex(A).



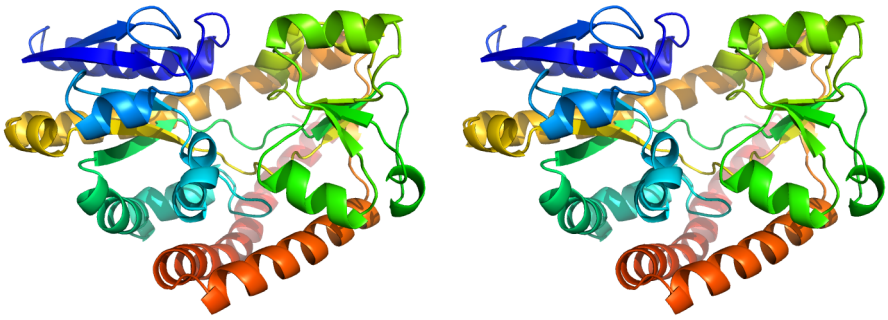
(b) Stereo view of 3b50(A).

Figure 49: Stereo views of two sequence-identical sialic acid binding protein structures 2cex(A) and 3b50(A), coloured by the intrafragment rotational dissimilarity main chain scores arising from their comparison (with $n = 9$) .

a high degree of local conformational conservation is maintained everywhere despite the global conformational change. However, the central score (Figure 48) identifies that there are residues about which the local structure environment diverges. This suggests that these residues are those involved in the conformational change, such as the hinge region. The intrafragment rotational dissimilarity score (Figure 49) identifies these same regions as being relatively dissimilar, indicating that subtle local conformational change of the backbone does indeed occur in these regions, and that the observed relatively high central scores are not simply due to local dissimilarity alone. Furthermore, local conformational change is observed to occur within the interior of the domain displayed on the right. This suggests that the hinging domain motion is not the only conformational change, and that this domain on the right undergoes structural change during the binding process (although, in general, such differences may also be due to other factors, such as crystal packing). Note that the intrafragment rotational dissimilarity score is a stronger indicator of local conformational change than the central score. For example, a few residues are identified as red in a bent helix at the bottom of the lower images. These few residues, which are only identified by the intrafragment rotational dissimilarity score, are indeed part of the 'hinge'.

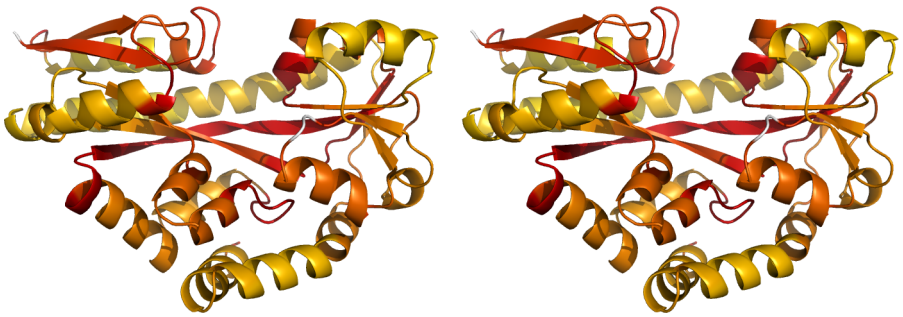


(a) Stereo view of 2cex(A).

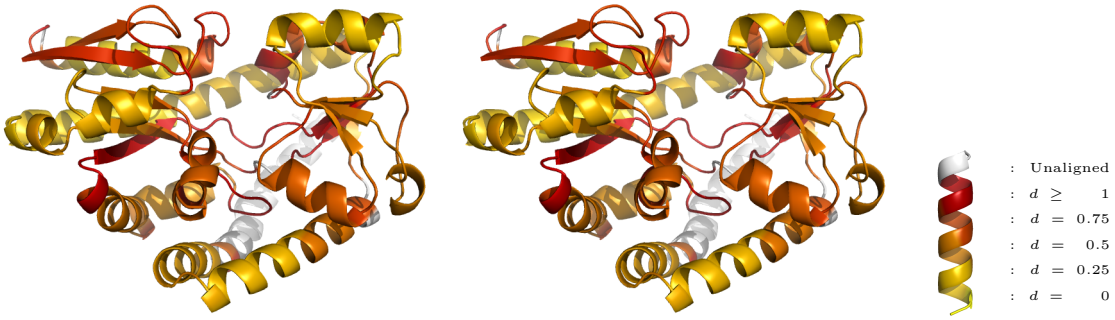


(b) Stereo view of 2hzk(A).

Figure 50: Stereo views of sialic acid binding protein 2cex(A) (Muller et al., 2006) and sodium-alpha-keto acid binding protein 2hzk(A) (Gonin et al., 2007), which share 14% sequence identity (on aligned residues). Structures are oriented so their coordinate frames correspond to the global superposition, and are shown rainbow-coloured along the chain from blue (N-termini) to red (C-termini).

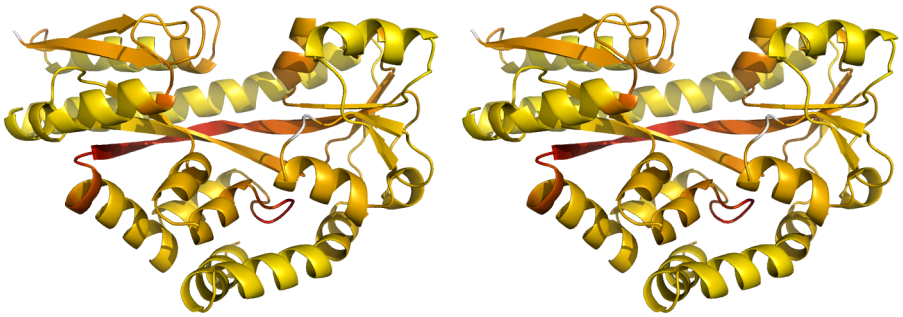


(a) Stereo view of 2cex(A).

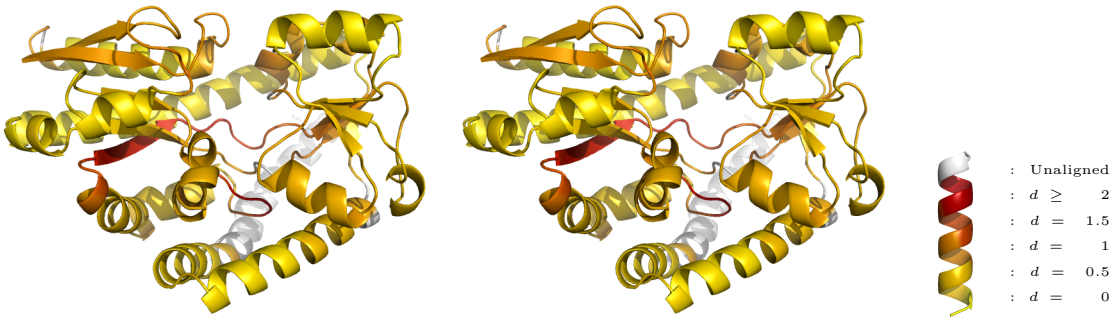


(b) Stereo view of 2hzk(A).

Figure 51: Stereo views of a sialic acid binding protein 2cex(A) with a sodium-alpha-keto acid binding protein 2hzk(A), coloured by the minimum main chain scores arising from their comparison (with $n = 9$), shown with a steep colour gradient.



(a) Stereo view of 2cex(A).



(b) Stereo view of 2hzk(A).

Figure 52: Stereo views of a sialic acid binding protein 2cex(A) with a sodium-alpha-keto acid binding protein 2hzk(A), coloured by the minimum main chain scores arising from their comparison (with $n = 9$), shown with a gradual colour gradient.

Figures 50, 51 and 52 show the comparison of two structures (the target structure is the same as in the previous example) that are globally similar, sharing the same topology and fold. However, these structures are divergent in sequence, having only 14% sequence identity. Nevertheless, the high degree of structural conservation allows *ProSMART* to appropriately align the two chains. It is evident that there is far less local structural similarity between these chains than observed between the sequence-identical pair previously considered (Figure 51 displays the minimum score on the same colour gradient used in Figure 47), although local backbone structure is still preserved in many regions. Altering the colour gradient (Figure 52) provides different information, allowing more intuitive identification of regions that are least locally conserved between the two chains.

3.3.2 Side Chain Scores

Further to scores describing the structural dissimilarity of the main chain, *ProSMART* also provides measures of the conformational conservation of side chains, relative to their assumed local coordinate frames. Two scores are provided: the side chain RMSD, and the side chain average position. The former is calculated only for aligned residue-pairs that share the same amino acid type; the latter may be calculated even if the amino acid type is different.

In order to calculate the side chain scores, the two aligned residues are placed into a common coordinate frame, so that their atomic coordinates may be directly compared. Specifically, the atomic positions are transformed so that the best scoring fragment-pair that the residue-pair belongs to is optimally superposed (note that this best scoring fragment-pair was also used to calculate the residues' minimum main chain score). This method of superposition may be more robust than using the fragment-pair centred on the target residue-pair. The side chain score does not make sense for residues with sufficiently large (unfavourable) minimum scores. Side chain scores are nevertheless reported regardless of minimum score, so the user should take care to interpret results sensibly.

Side Chain RMSD

The side chain RMSD is calculated as the average distance between corresponding side chain atoms in the target residue pair, after being transformed into the common fragment-based coordinate frame. The C^{α} atoms are included as part of the side chain, so that all residues may be scored (i.e. the score is still meaningful for glycine). If atoms are missing from the side chains, then the score is still calculated, ignoring the missing atoms. Side chain atoms are aligned according to atom names. Consequently, similar side chains are identified as dissimilar if their nomenclature is inconsistent; this is considered to be a feature rather than a limitation.

For any aligned residue pair $(x, y) \in A_R$, let $(i, j) \in A$ specify the fragments associated with the residue pair's minimum score, in accordance with Equation (3.12). The coordinates of the side chain atoms may be placed into the same coordinate frame as the superposed fragments by applying the same transformation used to superpose the fragments. Denoting the coordinate matrices of the

side chain atoms by $\mathbf{C_{1x}}$ and $\mathbf{C_{2y}}$, the transformed coordinates are given by:

$$\hat{\mathbf{C}}_{\mathbf{1x}} = \mathbf{C_{1x}} - \vec{\mu}_{\mathbf{F_{1i}}} \tag{3.17}$$

$$\hat{\mathbf{C}}_{\mathbf{2y}} = \left(\mathbf{C_{2y}} - \vec{\mu}_{\mathbf{F_{2j}}}\right) \mathbf{R_{ij}} \tag{3.18}$$

where $\vec{\mu}_{\mathbf{F_{1i}}}$ and $\vec{\mu}_{\mathbf{F_{2j}}}$ are the mean positions of fragment i from protein 1 and j from protein 2, respectively, and $\mathbf{R_{ij}}$ is the rotation required to superpose fragment-pair (i, j). The side chain RMSD may then be calculated as the average distance between corresponding transformed coordinates:

$$s_{\mathrm{RMSD}}(k) = \sqrt{\frac{1}{M}\left(\mathrm{tr}(\hat{\mathbf{C}}_{\mathbf{1x}}^{\mathbf{T}}\hat{\mathbf{C}}_{\mathbf{1x}}) + \mathrm{tr}(\hat{\mathbf{C}}_{\mathbf{2y}}^{\mathbf{T}}\hat{\mathbf{C}}_{\mathbf{2y}}) - 2\mathrm{tr}(\hat{\mathbf{C}}_{\mathbf{1x}}^{\mathbf{T}}\hat{\mathbf{C}}_{\mathbf{2y}})\right)} \tag{3.19}$$

for $k = 1 \ldots |A_R|$.

Side Chain Average Position

The sidechain average position is the distance between the average positions of the side chain atoms after superposition, and is given by:

$$s_{\mathrm{AV}}(k) = \sqrt{\sum_{j=1}^{3}\left(\frac{1}{M_1}\sum_{m=1}^{M_1}[\hat{\mathbf{C}}_{\mathbf{1x}}]_{mj} - \frac{1}{M_2}\sum_{m=1}^{M_2}[\hat{\mathbf{C}}_{\mathbf{2y}}]_{mj}\right)^2} \tag{3.20}$$

where M_1 and M_2 are the numbers of atoms in the two side chains. This score may be calculated for residues with different amino acid types, so may be useful when comparing chains with very high structural similarity but non-identical sequences. However, results should be interpreted contextually, remembering that side chains from different amino acids will have different score distributions.

Examples

This functionality may be used for various purposes, including investigation and visualisation of changes in sites of interest, effects of crystal packing, etc. Achieved information may also be used in model refinement (e.g. inferring suitability of restraints) and identification of target residues for consideration during manual refinement. In suitable cases, this functionality allows the user to quickly visually pinpoint side chains that could be manually inspected to check for model correctness. In other cases, trends in side chain conservation might be visualised (e.g. for investigation of signalling). This method also allows identification of inconsistent nomenclature, which may be of interest.

An example of visualising side chain conformational change using the side chain RMSD score is given in Figure 53, in which six very similar NCS-related chains are compared. It is possible to quickly identify which side chains are in different conformations, and which are rigidly conserved. The ability to easily identify such subtle differences may have application when refining structures, i.e. consideration of whether observed differences are actual/significant.

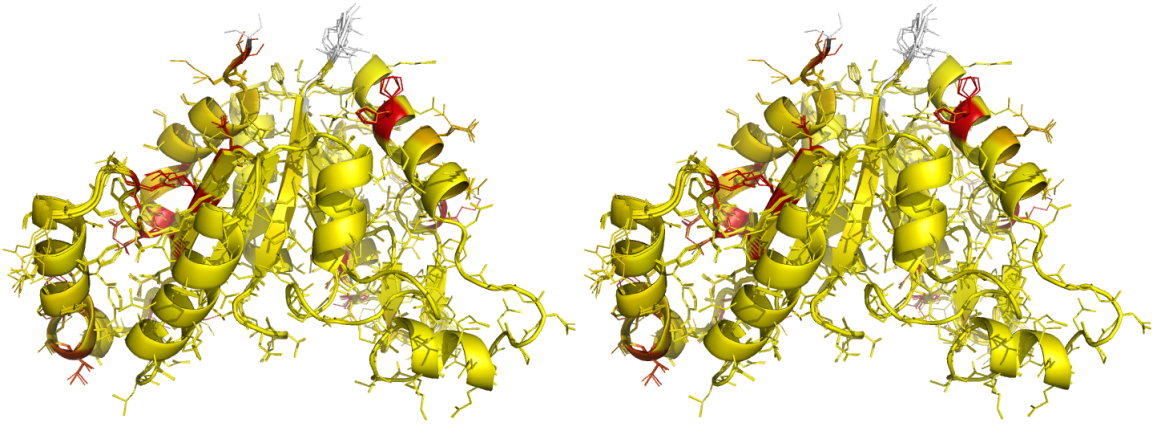


Figure 53: Comparison of the six NCS-related chains in the dethiobiotin synthetase protein with PDB ID 3mle (Nicholls et al., 2010), with chain A as the target. Compared chains are shown globally superposed in stereo. Residues are coloured according to side chain RMSD score, with yellow residues indicating strict conservation, red indicating different conformations ($s_{\mathrm{RMSD}} > 1$), and white for unaligned residues.

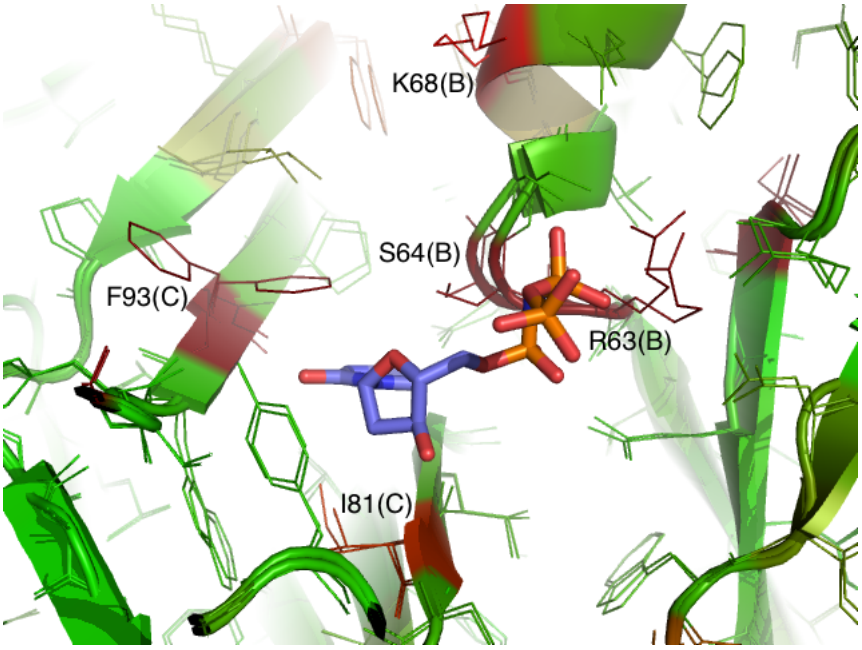


Figure 54: Depiction of the active site between chains B and C of a dUTPase protein (Garcia-Nafria et al., 2010). Specifically, the structures with PDB codes 2xcd (apo) and 2xce (bound) are shown, including the ligand from 2xce. The coordinate frame of 2xce is used; each chain from 2xcd is superposed onto the corresponding chain in 2xce. Some residue labels are added for illustration. Residues are coloured using a green-red gradient, according to the side chain RMSD score. Residues with conserved side chains are coloured green, those with $s_{\mathrm{RMSD}} > 1$ are coloured red.

Figure 54 shows an example of the sensitive conformation-independent identification of subtle differences between identical structures that is possible using *ProSMART*. Being structurally very similar, the chains globally superpose reasonably well. However, the small degree of global conformational change present results in the side chains not superposing perfectly. By colouring residues by the side chain RMSD score, it is possible to easily distinguish between side chains that have changed (red), and those that have maintained (green), conformation relative to their local coordinate frame, irrespective of how well superposed they are. Definition of the local coordinate frame depends on choice of structural resolution (in this case, $n = 9$). This example demonstrates the usefulness of simultaneously combining different types of visual information, i.e. colour and spatial (superposition). The colour scheme makes it immediately apparent that the biologically relevant F93(C) changes rotameric state, and I81(C), R63(B), S64(B) and the surface residue K68(B) exhibit conformational change in the presence of the ligand (although it is not implied that the presence of the ligand is necessarily responsible for all of these differences).

3.3.3 Superposition

Further to outputting text files for residue-based alignment scores, *ProSMART* provides superposition results in the form of PDB files and transformation matrices. Furthermore, various *PyMOL* (Schrödinger, LLC, 2010; DeLano, 2007) scripts that colour residues according to scores are output. Most functionalities use a colour gradient (colours and gradients may be chosen) representing similar to dissimilar, providing a powerful way of visualising results.

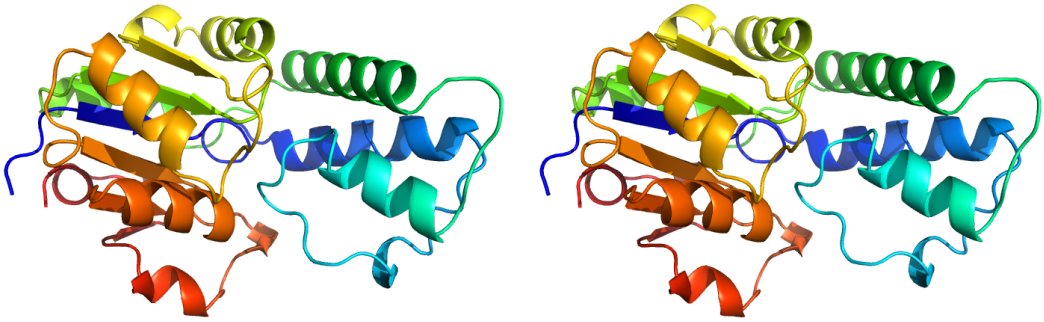
By default, all aligned residues are used to calculate the global superposition. However, *ProSMART* allows the alignment to be filtered using a Procrustes score threshold, allowing the global superposition to be calculated using only residues that have sufficiently well-conserved local structural environments. Such a threshold is not applied in any examples considered here.

Example of Global and Rigid Substructure Superposition

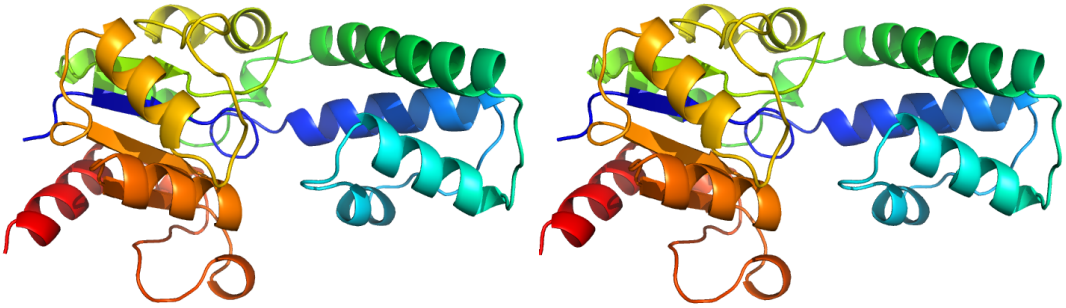
Figure 56 depicts the minimum scores resulting from the comparison of two structurally similar chains with relatively low sequence identity (24% of the 209 aligned residues were identical in sequence). The two chains are classified in *SCOP* as belonging to the same superfamily but different families, phosphatase YniC and beta-phosphoglucomutase, respectively. It is apparent that overall fold and topology of these structures is the same. Some regions of local structure are very well conserved, whilst some surface loops display more conformational flexibility.

The global superposition of these chains (Figure 57a) results in a reasonable overlay of both domains, allowing easy visual identification of secondary structure element correspondences. However, this superposition is not particularly good anywhere, providing no information of clarity regarding the maintenance of any internal structural rigidity.

One rigid substructure is identified using *ProSMART*'s rigid substructure identification func-

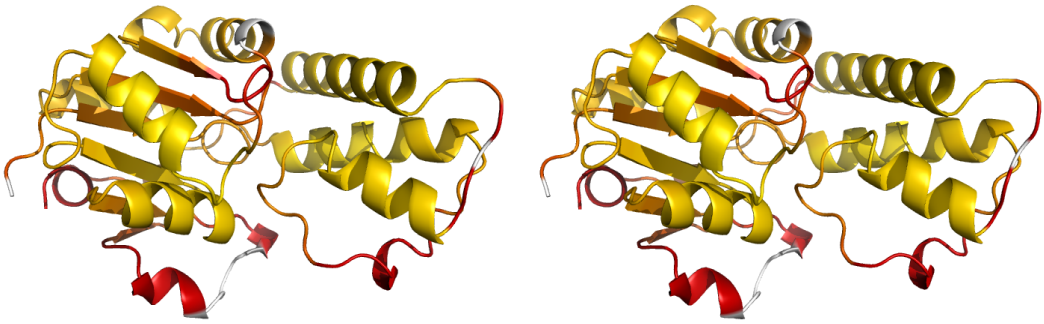


(a) Stereo view of 1te2(A).

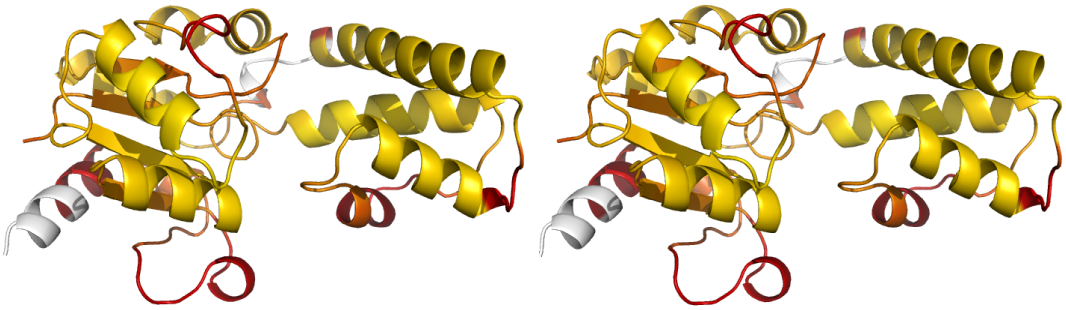


(b) Stereo view of 1zol.

Figure 55: Depictions of (a) the putative phosphatase 1te2(A) (Kim et al., 2004) and (b) the beta-phosphoglucomutase 1zol (Zhang et al., 2005), which share 24% sequence identity. Illustrations of the compared structures are rainbow coloured along the chain from blue (N-termini) to red (C-termini).

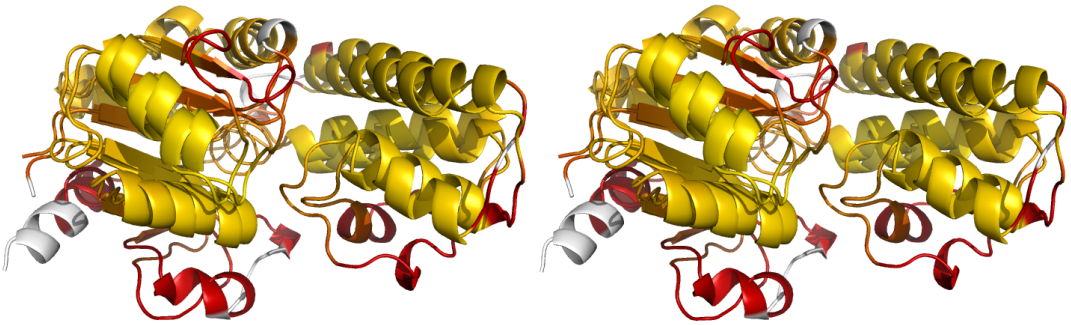


(a) Stereo view of 1te2(A).

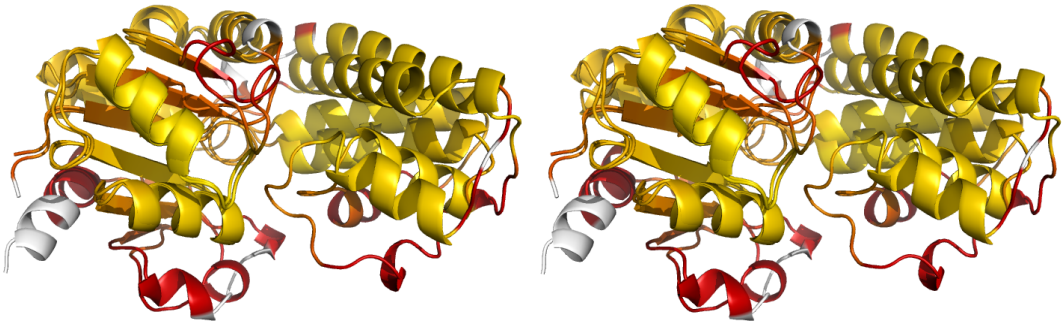


(b) Stereo view of 1zol.

Figure 56: Depictions of (a) the putative phosphatase 1te2(A) (Kim et al., 2004) and (b) the beta-phosphoglucomutase 1zol (Zhang et al., 2005), with residues coloured according to the minimum score resulting from their comparison. Yellow indicates locally conserved structure, those with minimum scores $d > 1$ are coloured red, and those unaligned are coloured white.



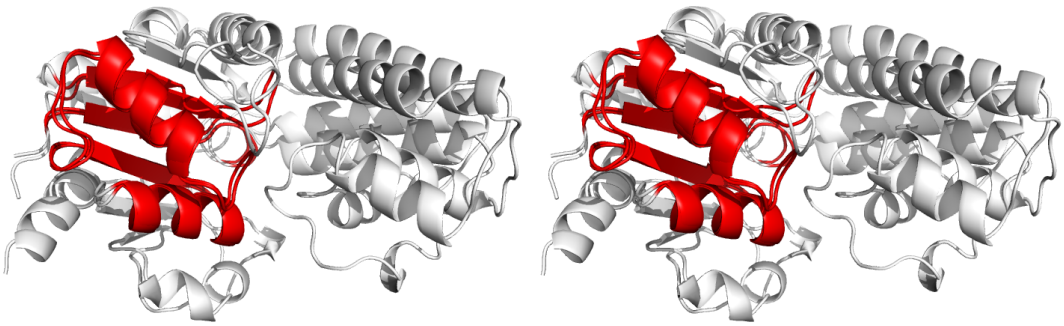
(a) Stereo global superposition of 1te2(A) and 1zol, with residues coloured by minimum score.



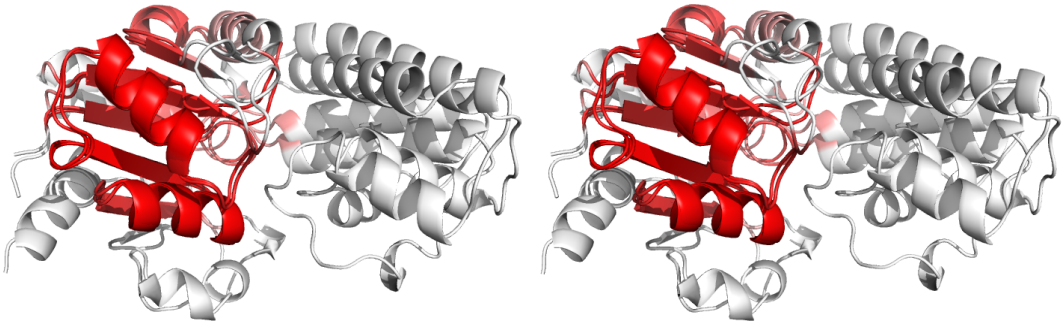
(b) Stereo substructure superposition of 1te2(A) and 1zol, with residues coloured by minimum score.

Figure 57: Superpositions of the putative phosphatase 1te2(A) (Kim et al., 2004) and the beta-phosphoglucomutase 1zol (Zhang et al., 2005). Structures are coloured by minimum score (yellow for conserved structure, red for high minimum scores $d > 1$, and white for unaligned residues). Subfigure (a) shows the global superposition using all aligned residues. Subfigure (b) shows the structures superposed according to the rigid substructure identification functionality; one rigid substructure is identified.

tionality, which is used for superposition of these chains in Figures 57b and 58. This view improves superposition of the domain shown on the left, at the expense of the domain on the right (which is less rigidly structurally conserved). Using this view, it is easy to see that the core of the superposed domain is reasonably rigidly structurally conserved. It is also possible to visually identify which surface loops are not conserved. The well-conserved region used for definition of the rigid substructure's coordinate frame is displayed in Figure 58. A depiction of the degree of orientational belongingness to this substructure is also provided (Figure 58b), allowing for intuitive visualisation of the residues considered reasonably close to the substructure's coordinate frame.



(a) Stereo substructure superposition of 1te2(A) and 1zol, with residues very close to the rigid core coloured.



(b) Stereo substructure superposition of 1te2(A) and 1zol, coloured by substructure belongingness score.

Figure 58: Superpositions of the putative phosphatase 1te2(A) (Kim et al., 2004) and the beta-phosphoglucomutase 1zol (Zhang et al., 2005). Structures are superposed according to the rigid substructure identification functionality; one rigid substructure is identified. Subfigure (a) shows residues coloured red if they are part of the final cluster used for definition of the rigid substructure. Colour intensity in subfigure (b) represents orientational agreement with the substructure's coordinate frame. Intense colour indicates strong orientational agreement, gradually fading to white at $\hat{d}^{\theta}_{\hat{c}_i k} = 0.005$ ($\approx 6°$. See §2.3.3 for details). Residues not belonging to any aligned fragment-pairs are also coloured white.

Example of Superposition with Multiple Rigid Substructures

The rigid substructure identification functionality may identify multiple distinct substructures. In such cases, superpositions are identified and output (in the form of PDB files and transformations) for each of the substructures. One *PyMOL* script is output that colours residues if they belong to the 'definition' (final cluster) of any substructure. In this case, each substructure is assigned a different colour, and residues are coloured accordingly. For continuity and ease of interpretation, substructures also maintain this colour scheme in the individual scripts that colour residues according to substructure belongingness.

For example, the global superposition of the sequence-identical binding proteins displayed in Figure 59 results in a poor structural overlay, due to the conformational change. As seen in Figure

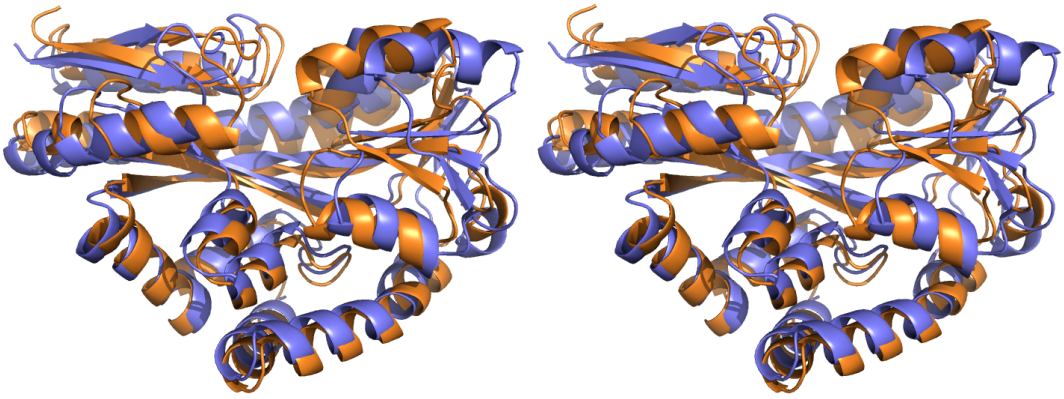
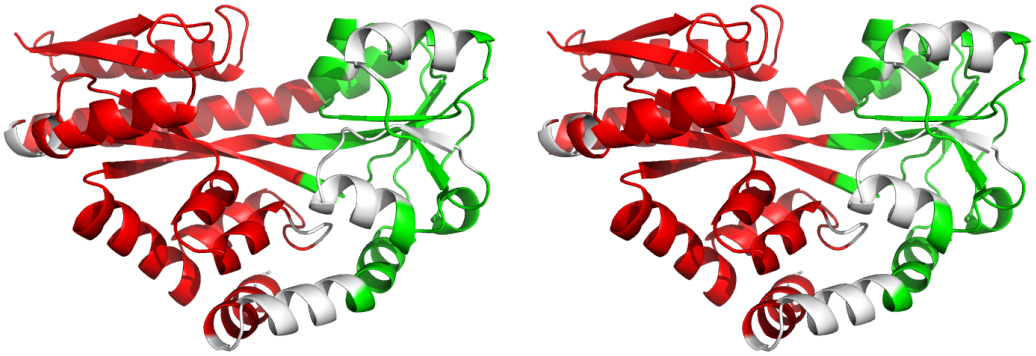
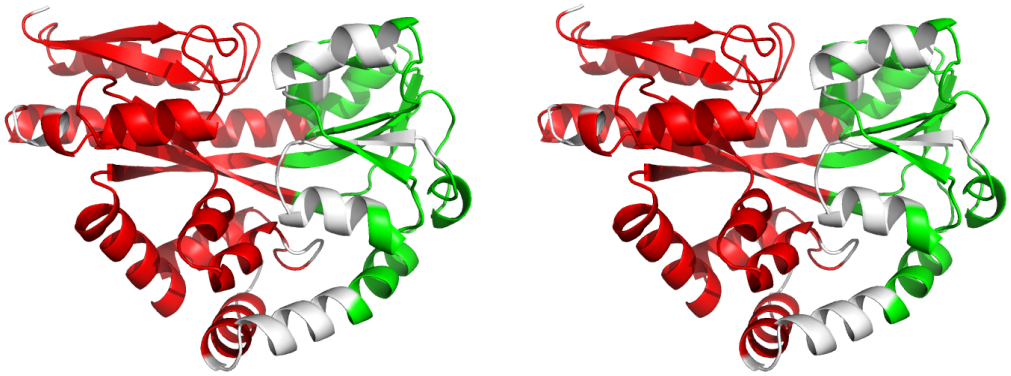


Figure 59: Stereo global superposition of sequence-identical structures 2cex(A) (blue) and 3b50(A) (orange), resulting from a structural comparison using fragment length $n = 9$.



(a) Stereo view of 2cex(A).



(b) Stereo view of 3b50(A).

Figure 60: Results of rigid substructure identification for the comparison of 2cex(A) and 3b50(A), with fragment length $n = 9$. Two clusters are identified, which correspond to the regions coloured in red and green, respectively. The figure shows (a) 2cex(A) and (b) 3b50(A) oriented so that their coordinate frames correspond to the global superposition. Residues are coloured (red or green) according to the closest substructure, providing they are used as part of the substructure definition; those coloured white are not used for definition of the final clusters.

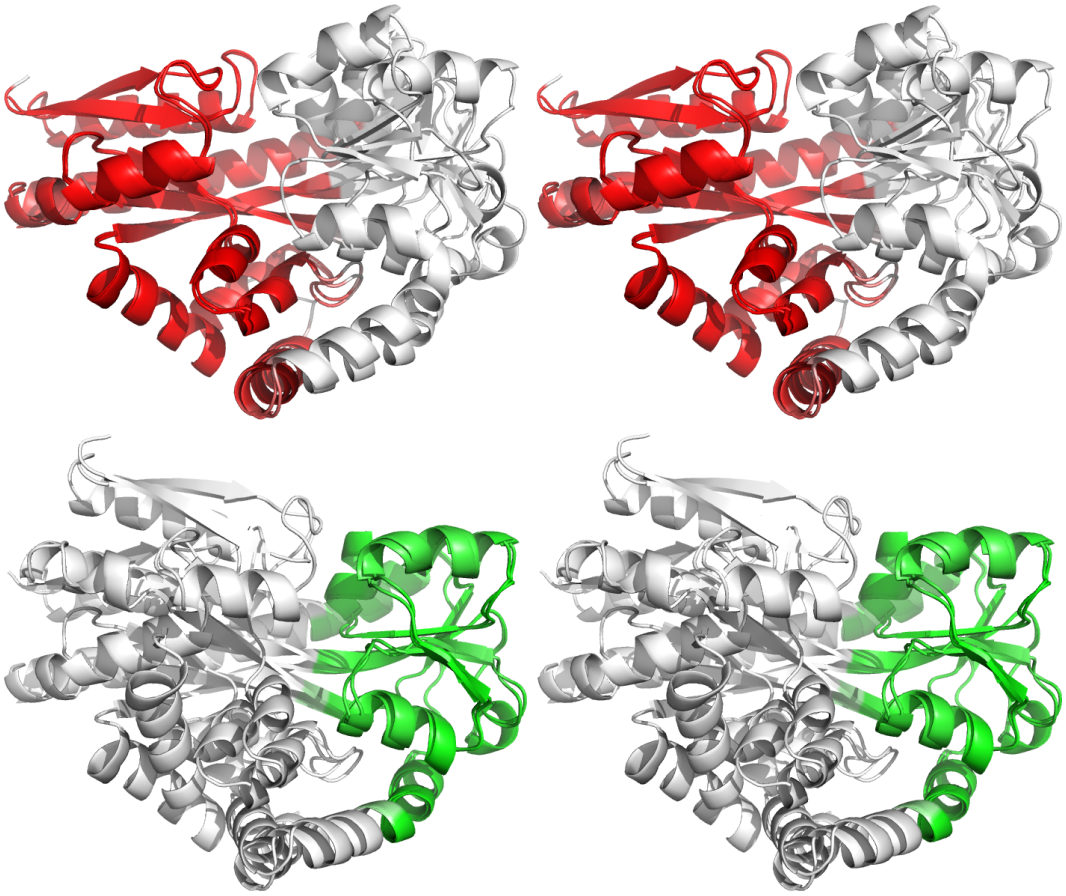
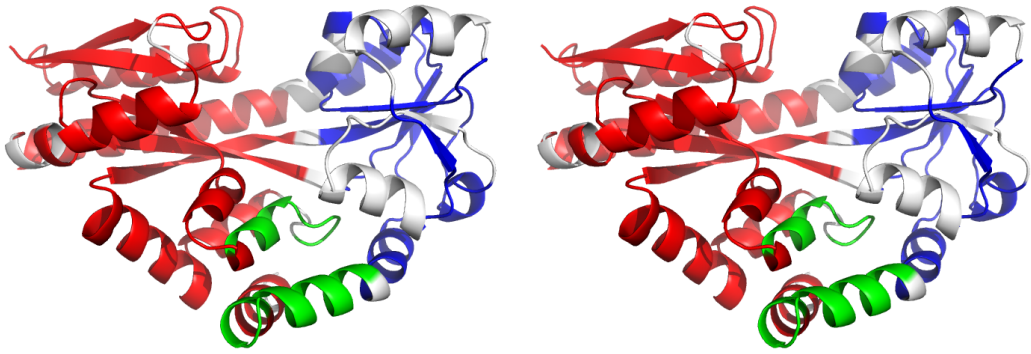


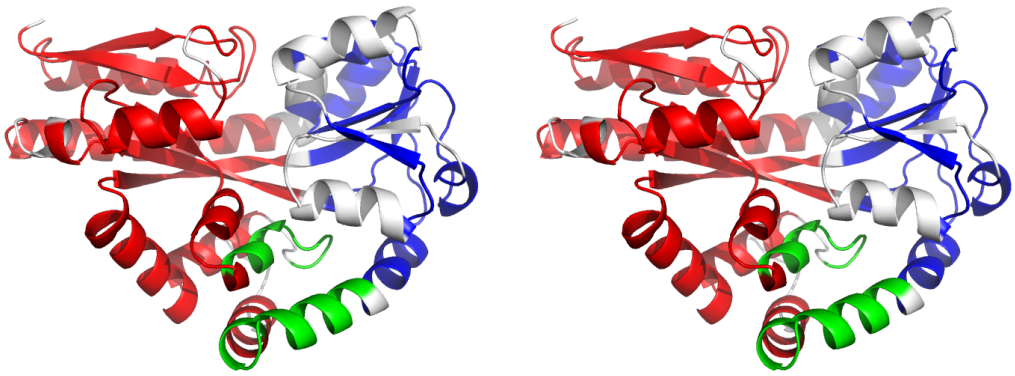
Figure 61: Stereo superpositions of the first (red, upper) and second (green, lower) identified rigid substructures resulting from the comparison of 2cex(A) and 3b50(A), with fragment length $n = 9$. The images are shown in the coordinate frame of 2cex(A); the secondary chain 3b50(A) is transformed according to the substructures' normalised average quaternions. Colour intensity indicates orientational agreement with the substructure's coordinate frame. Intense colour indicates strong orientational agreement, gradually fading to white at $\hat{d}^{\theta}_{\hat{c}_i k} = 0.005$ ($\approx 6°$. See §2.3.3 for details). Residues not belonging to any aligned fragment-pairs are also coloured white.

60, two rigid substructures are identified, which correspond to the domains on either side of the hinge. The resultant superpositions shown in Figure 61 provide a much better overlay, indicating that both domains display rigid conservation between the two structures. Interestingly, two helices in the domain shown on the right are not used as part of the substructure definition (see Figure 60). In fact, these two helices are brought closer together, in a clamping motion, when the ligand is bound in 3b50(A). Further to the global hinge motion, this observation suggests binding to also incur biologically-relevant intradomain motion in this case. The subtle nature of this observation demonstrates a potential application of this functionality of *ProSMART*.

Note that part of one helix (located at the bottom of the images) is not identified as sufficiently 'close' to either rigid substructure. This indicates that this small region does not belong to either domain, and that the kink in the helix is relevant to the domain motion (this can also be observed using the intrafragment rotational dissimilarity score, previously considered in Figure 49).



(a) Stereo view of 2cex(A).



(b) Stereo view of 3b50(A).

Figure 62: Results of rigid substructure identification for the comparison of 2cex(A) and 3b50(A), with fragment length $n = 7$, in contrast with Figure 60. Three clusters are identified, which correspond to the regions coloured in red, green, and blue, respectively.

It is important to acknowledge that choice of parameter values can have a considerable effect on the qualitative and quantitative nature of results. For example, the substructures displayed in Figures 60 and 61 were realised using the default choice of fragment length $n = 9$. In contrast, those identified in Figures 62 and 63 were realised using fragment length $n = 7$ (other parameters were unchanged). This simple parameter alteration results in three substructures being identified. Two of these correspond to the two domains previously identified. The other substructure corresponds to the hinge region, including the helix that was not previously identified as belonging to either domain. Whilst only two small regions are used in the definition of this substructure, other regions in the hinge are identified as having a high degree of belongingness; this indicates that these regions have an orientationally similar coordinate frame to the substructure. Specifically, these regions include the kink in the long helix (located at the back of the images) and, to a lesser degree[3], the twisted strand in the centre of the hinge. Note that these regions are spatially separated from the core of the rigid substructure; the employed method is independent of spatial location, being interested only in orientational agreement of residues' local structural environments.

[3]the paler green indicates that this region is not as 'close' to the substructure as regions with more intense colour.

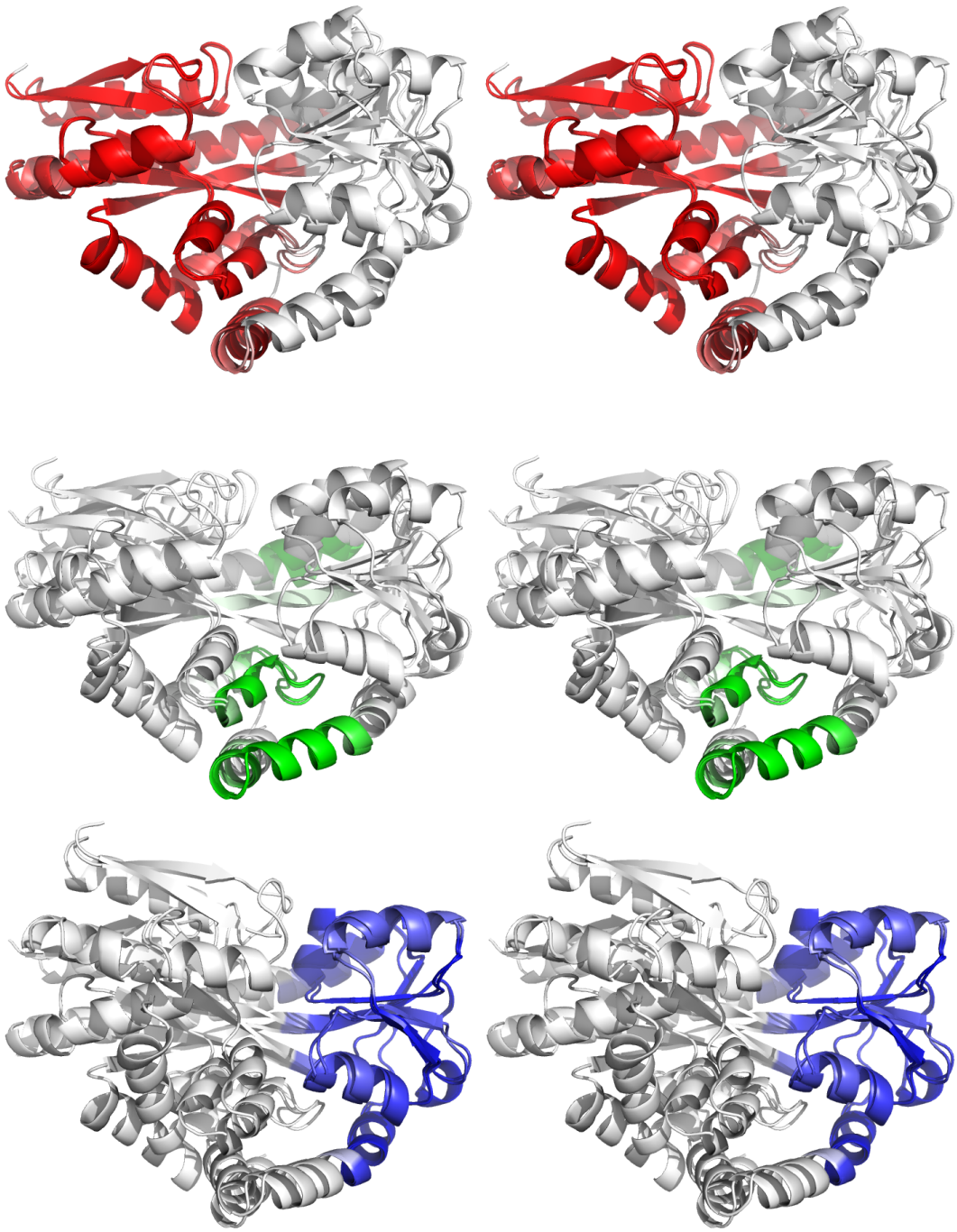


Figure 63: Stereo superpositions of the three rigid substructures identifed by the comparison of 2cex(A) and 3b50(A) with fragment length $n = 7$, in contrast with Figure 61. Superposition and colouring methods used for the three identified substructures are similar to that used in Figure 61.

Interdisciplinary Usage

ProSMART may be used with structures from various experimental/theoretical methods, including X-ray crystallography, electron microscopy, nuclear magnetic resonance (NMR) spectroscopy and molecular dynamics (MD) simulations. However, models comprising only the C^{α} trace are not currently supported, since the presence of all main chain atoms is required for default definition of a structural fragment. For NMR and MD structure ensembles, each of the states in the ensemble are pairwise compared with the first state. *ProSMART* allows residues to be coloured by the worst score[4] over all states, allowing depiction of maximum local structural dissimilarity (although pairwise chain colouring is also possible).

For example, Figure 64 displays the global superposition of an NMR structure ensemble. The addition of colour intuitively provides information regarding conformational flexibility that would be more difficult to visually discern otherwise. Figure 65 displays superposed results from an MD simulation. Superposition of both the global alignment and rigid substructures are shown, coloured by maximum dissimilar minimum score, demonstrating how various functionalities of *ProSMART* might be combined to produce the desired results.

[4]Maximum dissimilarity scoring for structure ensembles is currently supported for the central, minimum and intrafragment rotational dissimilarity main chain scores, and the side chain RMSD and average scores.

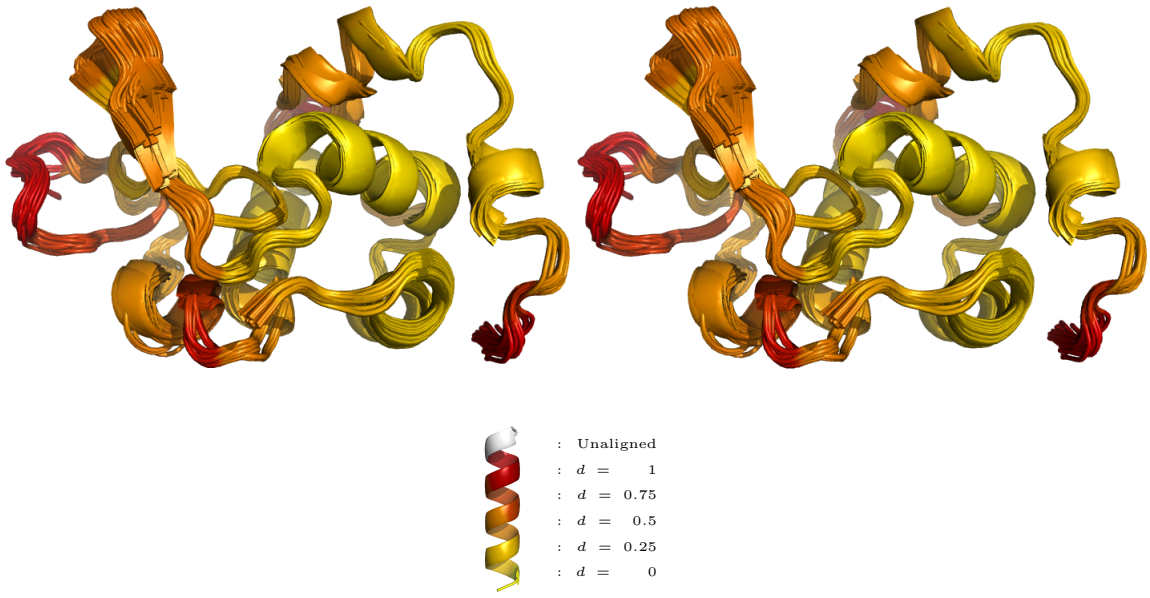
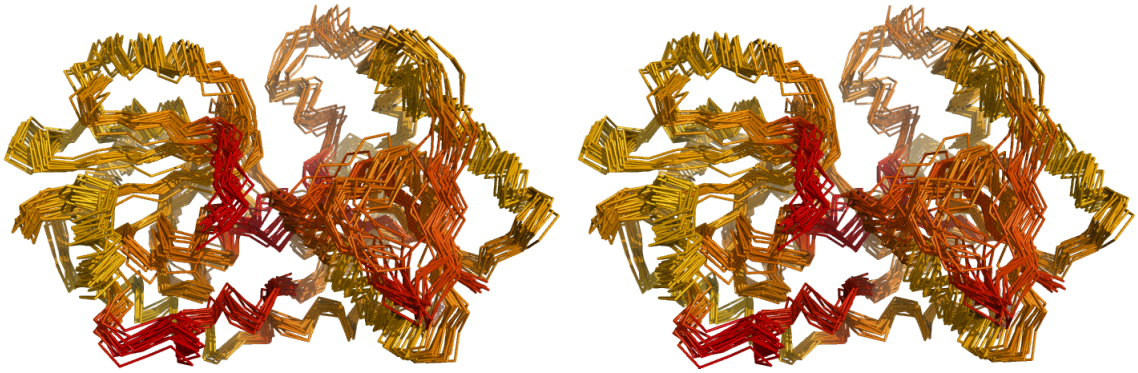
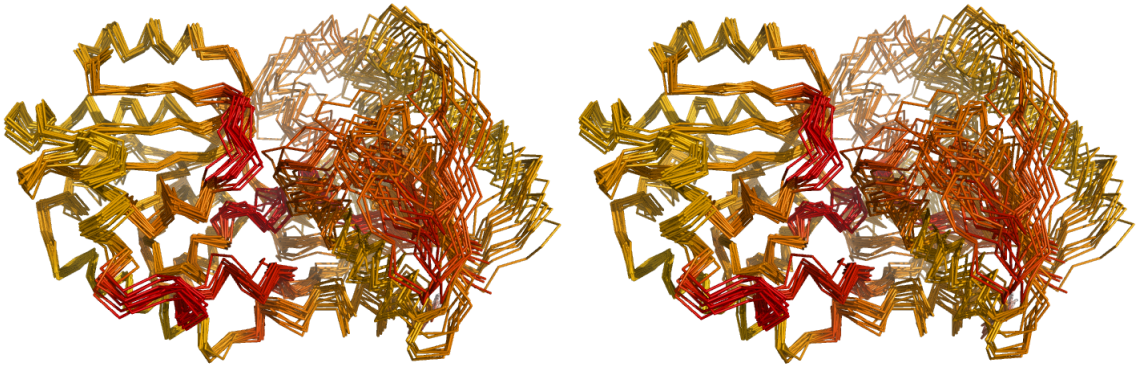


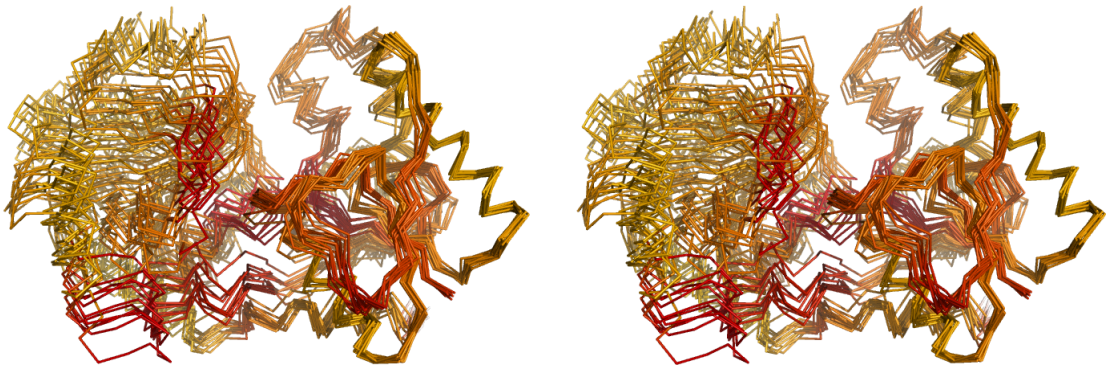
Figure 64: Stereo global superposition of the NMR structure ensemble 1e8l of hen lysozyme (Schwalbe et al., 2001), comprising 50 states. Residues are coloured by worst minimum score over all states.



(a) Stereo global superposition of a WaaG ensemble.



(b) Stereo superposition of the first identified rigid substructure in a WaaG ensemble.



(c) Stereo superposition of the second identified rigid substructure in a WaaG ensemble.

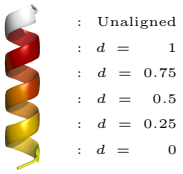


Figure 65: Superpositions of the structure ensemble from a 55ns MD simulation of WaaG, which comprises 19 states (PDB file and permission to use this example courtesy of Dr Jens Landström). The upper image shows the global superposition. The lower images display superpositions of the identified rigid substructures. Residues are coloured by worst minimum score over all states.

3.3.4 Global Alignment Statistics

ProSMART supplies various statistics providing information about how well the two structures are aligned. These include the percentage of aligned residues, the percentage strict sequence identity of aligned residues, and the global RMSD of main chain atoms after superposition of all aligned residues. Other statistics derived from the main chain scores are also provided, specifically the average of minimum and central scores.

The overall minimum score of the global alignment is given by the average of the aligned residues' minimum scores:

$$D_{\text{min}} = \frac{1}{|A_R|} \sum_{k=1}^{A_R} s_{\text{min}}(k) \tag{3.21}$$

using the same notation as in §3.3.1.

The central scores represent a one-to-one mapping between aligned fragments and residues. Consequently, unlike with the minimum scores, individual fragments have equal weighting in the distribution of central scores (the score of each fragment is only counted once). The overall central score of the global alignment, which is given by the average of the aligned residues' central scores, may thus be calculated directly from the fragment alignment:

$$D_{\text{central}} = \frac{1}{|A|} \sum_{k=1}^{|A|} \mathbf{D}_{a_k} \tag{3.22}$$

Presentation of Results

The global statistics are provided as text files for each compared chain-pair. Also, for each *ProSMART* execution, HTML-format matrices of global scores are provided (one for each statistic), allowing results from multiple pairwise comparisons to be viewed concurrently. Text-format versions of these matrices are also output, for easier parsing by external applications.

	2cexA	2cexB	2cexC	2cexD	2ceyA	2v4cA	3b50A
2cexA		2.9	0.721	0.466	0.568	3.25	3.24
2cexB			3.12	3.09	2.51	0.583	0.55
2cexC				0.734	0.943	3.46	3.45
2cexD					0.751	3.5	3.49
2ceyA						2.91	2.91
2v4cA							0.431
3b50A							

	2ceyA	2v4cA	3b50A
2cexA	0.568	3.25	3.24
2cexB	2.51	0.583	0.55
2cexC	0.943	3.46	3.45
2cexD	0.751	3.5	3.49

Figure 66: Screenshots of output HTML-format global score matrices, corresponding to the global RMSD of all aligned residues after superposition. Images correspond to the all-on-all alignment of all (sequence-identical) chains in PDB IDs 2cex, 2cey, 2v4c, and 3b50 (left), and the alignment of 2cex against 2cey, 2v4c and 3b50 (right). Yellow text represents global conformational similarity, gradually changing to red indicating relative dissimilarity.

In general, the comparison of x target chains with y secondary chains would result in an $x \times y$ matrix of scores, for each statistic. Intuitively, an all-on-all comparison of x chains would result in an upper-triangular matrix of dimensionality x. Where appropriate, the numeric text elements of the HTML-format matrices are coloured according to a notion of similarity/dissimilarity (actual colours may be chosen).

An example of this functionality is shown in Figure 66, which displays matrices of global RMSD scores for seven sequence-identical chains. It is possible to identify that there are two clusters of globally conserved chains. These clusters comprise: 2cex(A), 2cex(C), 2cex(D) and 2cey; and 2cex(B), 2v4c and 3b50. These clusters correspond to open and closed conformations, respectively.

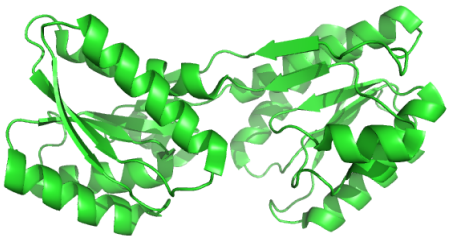
Example Demonstrating Issues with Global Alignment Scoring

Figure 67 shows an example of the comparison of 20 chains from 4 different families, using the average central score. The displayed matrix depicts both intra-class and inter-class average Procrustes scores. Families A and B are both classified by *SCOP* as α/β, family C as all-β, and family D as all-α. Five chains from each family were arbitrarily selected for inclusion, subject to having reasonably low sequence identity with other chains in the family (as can be seen in Figure 67). It is not implied that this example represents a suitable comparative analysis of these families; the example was chosen merely to exhibit certain phenomena.

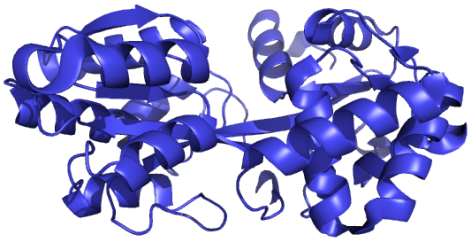
As might be expected, the intra-family average central scores of all four groups display evidence of internal similarity. However, not all intra-family scores are lower than inter-family scores, indicating that there is no clear threshold defining similarity. The score penalises any chain-pair that does not exhibit strong conservation of local structure; note that groups A and B are relatively dissimilar despite belonging to the same *SCOP* class.

All-α group D exhibits some of the lowest dissimilarity scores. Dissimilarity scores between all-α and α/β structures are generally much lower than between other groups, due to the alignment of many α-helical fragment-pairs randomly scoring well thus reducing the global score. In contrast, scores between groups C and D are the largest, with all scores being greater than 3Å, as would be expected for the comparison of all-α and all-β structures.

Scores within group C are higher than for other groups. However, this does not imply that intra-group similarity for group C should necessarily be considered lower than for other groups. Fragments in these structures (mainly β-strands) are generally more flexible than those in the other groups, thus the distribution of scores is inherently higher. This is reflected in the scores between group C and other groups – these scores (effectively between random structures) are approximately 1.5–2Å for all-β versus α/β, 2.5–3Å for all-β versus all-α, whilst being lower (1–1.5Å) for other groups. This demonstrates that the conceptual threshold defining similarity/dissimilarity depends on structures' internal and relative properties. Note that such properties should be considered to be continuous, and should not simply be discretised as class α and/or β.

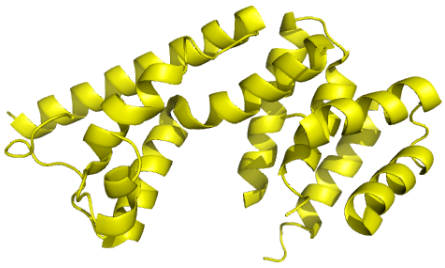


(a) 1gub(A) from family A (α/β class)



(b) 1o7t(A) from family B (α/β class)

(c) 3bp5(A) from family C (all-β class)



(d) 3dto(A) from family D (all-α class)

	1gubA	1gcgA	1tjyA	1jx6A	2vk2A	1o7tA	2pt1A	2v84A	1sbpA	2qryA	3bp5A	2ok0L	2aw2A	1xedA	1q8mA	3dtoA	3djbA	3b57A	2pq7A	2pjqA
1gubA		0.79	0.821	1.1	0.799	2.05	2.09	1.76	2.29	2.01	2.4	2.65	2.35	2.2	1.91	1.64	1.73	1.66	1.08	1.64
1gcgA	18.7		0.913	0.965	0.673	1.94	2.23	1.92	1.81	1.85	2.36	2.8	2.83	2.83	2.12	1.64	1.72	1.42	1.26	1.67
1tjyA	17.5	12		1.23	0.828	2.06	2.01	1.84	1.61	1.71	2.21	2.78	2.1	2.57	1.97	1.92	2.21	1.27	1.9	1.93
1jx6A	11.3	11.7	9.25		1.02	2.07	1.91	2.06	1.78	1.87	2.66	2.69	2.37	2.84	2.38	1.32	1.48	1.25	1.16	1.2
2vk2A	21.2	16.6	15.4	13.2		1.97	1.95	1.94	1.54	1.67	2.4	2.42	1.81	2.57	2.8	1.2	1.39	1.77	1.29	1.51
1o7tA	8.3	6.32	8.12	6.79	11.8		0.772	1.23	1.25	1.11	2.54	2.76	2.33	2.79	2.95	1.43	1.92	1.72	2.12	2.18
2pt1A	7.63	4.32	5.82	8.02	7	26.3		1.02	1.21	1.27	2.7	2.83	2.07	2.65	2.65	2.09	1.95	1.49	1.71	1.94
2v84A	9.39	7.58	6.46	6.67	7.48	9.09	13.5		1.49	1.35	2.91	2.64	2	2.37	2.61	2.01	1.92	1.73	1.72	1.94
1sbpA	9.3	10.3	9.45	7.31	9.52	10.4	12.6	10.6		1.42	2.48	2.67	2.64	2.17	3.07	1.48	1.92	1.21	1.89	1.63
2qryA	7.84	7.52	6.82	6.82	6.43	20	18.2	12.4	11.4		2.77	2.84	2.14	2.69	2.68	1.84	1.65	1.62	1.76	1.86
3bp5A	8.85	8.77	6.14	8.77	7.89	3.51	2.63	5.45	5.26	1.75		1.13	1.41	1.53	1.28	3.62	3.38	3.85	3.48	3.46
2ok0L	7.76	6.42	7.34	7.37	4.19	5.48	4.67	6.45	7.32	4.29	21.6		1.49	1.39	2.07	3.36	3.44	3.21	3.38	3.31
2aw2A	5.45	5.45	3.64	7.27	7.27	4.59	5.45	3.64	6.36	6.48	17.2	16		1.76	1.34	3.84	3.43	3.56	3.58	3.34
1xedA	7.21	6.31	6.31	5.41	4.5	2.7	6.31	5.41	6.36	10.8	19.3	18.3	15.8		1.21	3.36	3.41	3.44	3.23	3.22
1q8mA	10.3	10.3	4.67	2.48	6.09	6.61	8.26	10.9	4.31	5.31	17.7	3.36	11.7	18.4		3.47	3.62	3.36	3.58	3.21
3dtoA	7.14	7.18	6.18	5.95	7.98	6.92	7.22	5.38	8.47	7.39	4.39	6.67	4.55	7.21	5		0.526	0.563	0.83	0.561
3djbA	6.45	5.98	5.85	6.38	7.74	13.3	3.8	5.35	5.56	5.49	8.77	8.28	5.61	2.75	8.26	51.7		0.563	1.09	0.802
3b57A	4.65	10.5	5.23	9.09	7.65	2.53	5.59	5.08	4.32	2.98	4.39	4.43	3.67	8.74	7.63	42.3	41.4		0.802	0.522
2pq7A	8.67	4.6	5.17	8.05	8.05	4.65	5.88	7.88	6.94	6.59	6.25	8.33	6.36	4.5	2.5	24.3	13.2	18.3		0.893
2pjqA	7.77	7.18	6.57	3.96	8.11	6.86	6.15	4.59	8.38	9.69	6.14	6.32	3.64	1.89	6.96	43.3	37.2	32.2	19.5	

Figure 67: Screenshot of the output HTML-format average central score matrix resulting from the all-on-all comparison of 20 chains. Average Procrustes score is shown in colour on the upper-right triangle (blue text represents similarity, gradually changing to red indicating relative dissimilarity), and sequence identity of aligned residues is shown in grey on the lower-left triangle. Compared chains consist of five chains from each of four *SCOP* (Murzin et al., 1995; Andreeva et al., 2008) families, termed: A 'L-arabinose binding protein-like' (α/β class), B 'Phosphate binding protein-like' (α/β class), C 'V set domain' (β class) and D 'HD domain' (α class). Family A comprises chains 1gub(A), 1gcg(A), 1tjy(A), 1jx6(A) and 2vk2(A); B comprises 1o7t(A), 2pt1(A), 2v84(A), 1sbp(A) and 2qry(A); C comprises 3bp5(A), 2ok0(L), 2aw2(A), 1xed(A) and 1q8m(A); and D comprises 3dto(A), 3djb(A), 3b57(A), 2pq7(A) and 2pjq(A). Representative chains from each family are illustrated above the matrix, specifically 1gub(A) (a, green), 1o7t(A) (b, blue), 3bp5(A) (c, red), and 3dto(A) (d, yellow).

This example leads us to conclude that the average of Procrustes scores of aligned fragments should not be generally considered sufficient to unambiguously infer degrees of similarity, without the complementary consideration of external information (e.g. class). The global scoring problem is considered in more detail in Chapter 4.

3.3.5 Fragment Type Identification – ProSMART Library

Further to the ability to identify an alignment between two protein chains, *ProSMART* provides the additional functionality of being able to identify the presence of particular types of fragments in a target protein chain, given appropriate prior description. Specifically, residues may be discretely classified according to their local structural environments. In general, this functionality aims to be able to identify whether small regions of the protein chain (fragments) are structurally similar to some target(s).

The implemented solution in *ProSMART* aims to categorise each residue according to a predefined list of target fragment classes. The list of classes is specified in a fragment library, which is extensible and customisable. By providing class criteria, we are able to identify all substructures within a protein chain that satisfy the given criteria. Some fragments/residues may be unclassified, since observed fragments must be sufficiently similar to the class representative in order to be classified (although note that complete partitioning of space could be achieved by relaxing criteria).

ProSMART Library

Fragment types and classification criteria are specified in an external fragment library, which is established upon *ProSMART* installation but may be easily subsequently modified. We shall refer to entries of the library as fragment classes, although without any implication, or intention, of the set of chosen representative fragments constituting a reasonable classification system. This allows generalisation to a much wider range of problems.

Each entry in the library comprises a PDB file and a list of class parameters. The specified PDB file contains the atomic coordinates of the representative fragment of that particular class. Multiple classes may be specified in a single PDB file by having different chain IDs. The parameters serve as classification criteria, and must be specified separately for each class. Specifically, these are:

- PDB filename – specifies location of fragment coordinates;
- Chain ID – desired chain within the PDB format file;
- Class identifier – unique code for each class;
- Procrustes dissimilarity score threshold – determines whether observed fragments are sufficiently similar to the class representative, thus specifying the allowed intraclass isotropic variability;
- Fragment length – indicates only the first n residues in the chain to be used;

- RGB colour code – used to represent this class in the output *PyMOL* colour scripts.

The class identifier is a code used to distinguish between classes; having multiple entries with the same class identifier would effectively result in multiple representative fragments for the class, thus allowing for more complex (e.g. anisotropic) class criteria if desired. The fragment length is specified separately for each class so that different classes may operate at different levels of structural resolution. The Procrustes dissimilarity score cutoff specifies the allowed within-class structural variability. Note that this threshold will be highly dependent on the choice of fragment length n; greater score variability would be expected for larger fragments. For example, an entry in the fragment library configuration file may be:

```
helix  A  H  0.3  5  0.0 0.5 1.0
```

which indicates that the class with ID H is represented by a 5-residue fragment in chain A of a file called 'helix.pdb' present in the library. Fragments are considered to belong to class H if the Procrustes score between the fragment and this representative is no greater than 0.3Å. Any resultant colour script will indicate for this class to be displayed in a blue hue $(0.0, 0.5, 1.0)$.

Setting the Procrustes dissimilarity score cutoff too low could result in false negative results; setting it too high may result in false positives (although note that the notion of 'false' is subjective). Therefore, careful consideration should be given to the choice of parameters for each class. For repetitive structure, a reasonable strategy may be to choose a relatively small fragment length, which allows a tight score cutoff to be used. Otherwise, the higher score cutoff required for longer fragment lengths would result in an larger margin for error.

Fragment Class Variability

It is important to acknowledge that there are various factors that may influence the observed within-class variability. For example, poor positional reliability (e.g. due to crystallographic mosaicity, thermodynamic properties, poor structural refinement etc.) would lead to worse dissimilarity scores, which would in turn increase the value of the score cutoff required to optimally recognise the fragments. Another important factor is the influence that a protein's global conformation has on local structure. Even sequence-identical protein structures can exhibit fragment conformational flexibility when in different bound states or subject to different conditions, or even between different stages of a protein during its ordinary dynamic behaviour. In extreme cases, local structure may adopt different conformations, as often occurs in disordered regions. As an aside, this highlights one of the potential difficulties that may be encountered during structural alignment, and identifies some objective differences between identifying similar substructures, and attempting to identify correspondences between structures.

A fragment's surroundings apply pressure that distorts the fragment away from the local attractor (or, more practically, the closest representative fragment) in conformation space, thus increasing the required score cutoff. This effect occurs along the main chain backbone, which causes curva-

ture and torsion of the fragment, but also occurs across spatially-related substructures. In the latter case, residues' side chains and their interactions play an important role in determining the exact conformation of the fragment. As well as interprotein interactions, there are other factors that influence side chain conformation, thus affecting main chain fragment conformation, such as hydrophobicity and surface accessibility. The net effect of all of these factors causes seemingly similar fragments to exhibit conformational differences, to a greater or lesser degree. Since different fragment types likely have different levels of exposure to the different factors, and different levels of resistance to their influence, the amount of within-class conformational variability will likely be different for each fragment class. Therefore, the Procrustes dissimilarity score cutoff should be sensibly chosen for each class.

Logically, fragments that are more rigid, compact and tightly packed, with many strong inter-fragment bonds that reduce the effective conformational degrees of freedom (e.g. helices), are less vulnerable to influence than those which are inherently more conformationally flexible (e.g. strands). Consequently, we might expect that a fragment's intrinsic properties, such as shape, could provide some information about the conformational variability within a class. For example, it may be possible to use shape descriptors, such as eigenvalues (see Chapter 4), in order to make inferences regarding class flexibility. Other suitable descriptors may include non-structural external information that can not be derived directly from the atomic coordinates, such as refinement statistics, and thermodynamic parameters (such as B-factors). However, the varying levels of quality of deposited structures in the PDB, combined with the improvement in crystallographic software over the years, amongst other factors, means that the accuracy and comparability of such parameters may be very variable. Consequently, the use of such external information may prove unreliable.

Quasi-Secondary Structure Identification

Whilst this formalism is general enough to identify the presence of any structural motif, thus having potential for a wide range of applications, it is apparent that this functionality has potential for a loose form of secondary structure identification. There are multiple existing methods of identifying or predicting secondary structure. Secondary structure elements (SSEs) are often defined by hydrogen bonding patterns, according to the criteria specified by the *DSSP* dictionary (Kabsch and Sander, 1983). Structure based approaches towards identification include those based on dihedral angles (Wood and Hirst, 2005), since secondary structure has a strong tendency to occupy deterministic regions of the Ramachandran plot (Hovmoller et al., 2002).

It should be noted that, rather than identifying SSEs according to their formal definition based on hydrogen bonding patterns, *ProSMART* categorises residues' local structural environments based on their raw similarity with that of the target fragments (e.g. representative SSEs). This criteria means that we only consider the similarity of structures connected in sequence; in contrast with traditional methods, information regarding the relative spatial location of fragments is purpose-

fully not utilised. Indeed, it is important to clarify that it is not intended for *ProSMART* to be used for secondary structure identification in the traditional sense, even though the employed approach might be used for quasi-SSE alignment by assuming criteria informally defining such types. Usefulness of the employed approach will depend on particular application.

Choice of Class Representatives

At present, the *ProSMART* fragment library has been kept as simple as possible. The library consists of two entries, namely an ideal α-helix and a representative β-strand, representing two commonly occurring repetitive SSEs. Coordinates of these fragments were generated using *COOT* (Emsley et al., 2010). The current implementation of this functionality may thus be considered as quasi-secondary structure alignment, whereby fragments are categorised as either α-helix, β-strand, or other. One natural extension to such a general library would be to include other SSEs defined by *DSSP*; this would require the identification of representative fragments for each class.

Note that β-strands have a tendency to bend, twist, stretch and compress away from their ideal conformations due to surrounding local and global structural influences. The same applies to helices, although to a lesser degree. Consequently, whilst not unreasonable, the coordinates of ideal conformations of these structural elements may not prove to be the best class representatives. In future, it may be appropriate to use well-determined naturally occurring, non-ideal fragments as class representatives. If combined with sensible library parameters, this may help to reduce the number of classification false-positives and false-negatives, depending on the desired application.

The employed formalism allows multiple fragments to be used to define a class. Using multiple representatives with smaller dissimilarity score cutoffs may help to improve class criteria. For example, straight, bent and twisted β-strands should all be classified as β-strands, according to their *DSSP* definition. However, having one class representative to encompass all of these types of strands would require a relatively large dissimilarity score cutoff, which would most likely induce some false-positive results. This is a direct consequence of the (presumed) anisotropy of substructural classes in n-residue fragment conformation space. The univariate nature of our primary criteria (namely the Procrustes score) means that all observations within a hypersphere of given radius are identified as positive hits. Therefore, we can never achieve parameter values such that we can guarantee that there will be no false-positives and no false-negatives for non-spherically distributed classes. The ability to vary the fragment length n helps, to some degree, by allowing us to sensibly choose the value of n that maximises cluster separability, thereby reducing the number of undesirable results. Consequently, the use of multiple representatives may prove useful for reducing the number of undesirable results for particularly non-spherical classes.

More generally, it may be appropriate to separately consider a representative from each region of high density in n-residue fragment conformation space of naturally occurring substructures. In effect, doing so would automatically classify all possible commonly occurring structural elements.

Naturally, different values of n would yield different results. By considering different values of n, it may be possible to form a multi-resolution view of local conformation space, potentially allowing the hierarchical classification of commonly occurring rigid substructures. The resultant list of representative structures would likely include, but not be limited to, the existing known SSEs; any other identified classes may lead to the identification of quasi-SSEs that correspond to favourable conformations, without the necessity to be described by hydrogen bonding patterns. However, since we intend to keep the default fragment library as simple as possible, such a study exceeds the scope of this work.

It should be acknowledged that previous studies have performed small fragment clustering in conformation space, and used the consequent representation of protein chains as structural sequences in various applications. For example, Friedberg et al. (2007) used structural fragments of length 5 residues to construct an alphabet of 20 letters, each of which represent a different region of conformation space. After representing protein chains using this alphabet, traditional sequence alignments were performed on the resultant structural sequences. The authors concluded that this method, which utilises structural information, yielded better results than a (select) sequence-based, but worse than a structure-based, alignment algorithm. However, this method was found to be dramatically less computationally expensive than the structural alignment algorithm.

Implemented Residue Categorisation

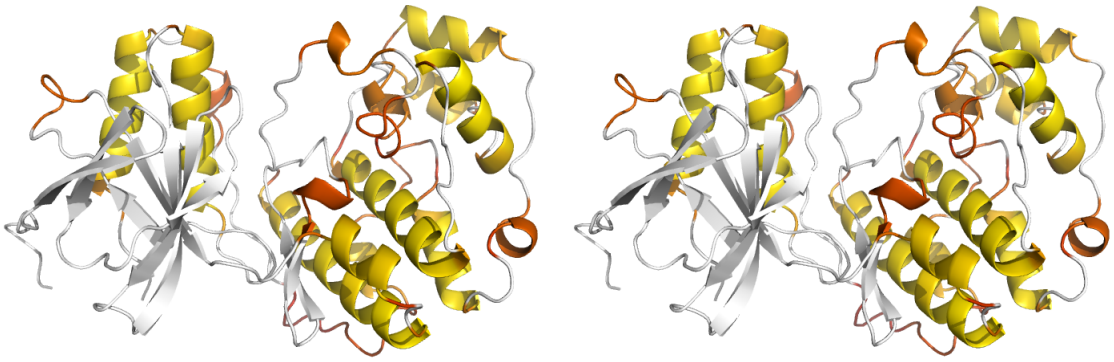
The implemented fragment type identification scheme, with reference to a fragment library, grants the ability to categorise individual residues according to their local structural environments. Specifically, if a residue belongs to a fragment that is identified as belonging to one of the classes specified in the fragment library, then the residue may also be considered to belong to that class. This means that, given an appropriate library, residues might be classified as sufficiently similar to either an α-helix (H), β-strand (S), or neither (N). This would allow the amino acid chain to be rewritten as a structural sequence, whose alphabet comprises the letters H, S, and N, in this case.

Remembering that the fragment library can be extended or customised, this functionality may have the potential to be utilised in various applications and fields, including structural analysis, alignment validation, structure identification, structure prediction, and crystallographic refinement. In particular, *ProSMART* currently utilises this functionality during the generation of atomic distance restraints for use in crystallographic refinement by *REFMAC* (see §3.4.2). Application in this field results from the ability to realise correspondences between a protein structure and a set of target fragments. For example, sometimes the positions of atoms in SSEs are not reliable, particularly in poorly determined low-resolution structures. If SSEs can be aligned using target fragments, restraints may be generated and used to potentially help their formation and stability during crystallographic refinement.

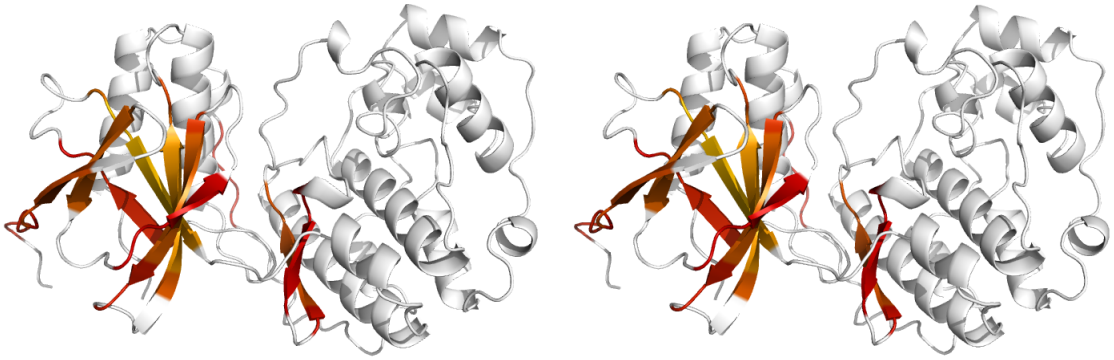
Unlike with the other functionalities of *ProSMART*, fragment type identification does not nec-

essarily require a secondary (external reference) structure in order to operate; results are unique to each target, subject to the entries and parameters specified in the fragment library. The procedure is the same whether one or multiple target structures are provided; if multiple structures are provided then fragment type identification is performed independently for each of the targets.

In implementation, for each entry k in the fragment library, a list of fragments is reallocated to the target protein structure according to the corresponding fragment length n_k. Each of the fragments i in the structure are compared with the coordinate matrix corresponding to the k^{th}



(a) Stereo view of residues sufficiently similar to an α-helix, coloured by Procrustes score.



(b) Stereo view of residues sufficiently similar to a β-strand, coloured by Procrustes score.

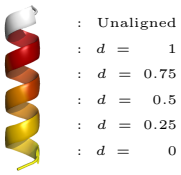


Figure 68: Fragment type identification results using the MAP kinase 3dt1 (Herberich et al., 2008), using a simple library comprising two fragments corresponding to a helix and a strand (see text for details). In this example, the fragment library is configured so that both fragment classes are represented by fragments of length 5 residues, with a Procrustes score threshold of 1Å. Residues are coloured according to scores between aligned fragments and the target class representatives; red indicates relative dissimilarity, fading to yellow for well-conserved. White residues are unaligned, according to class criteria. Results corresponding to the helix (a) and strand (b) fragments are shown.

entry in the library. The Procrustes dissimilarity score between these two compatible coordinate matrices is used to represent the distance between fragment i and class k. If this score is less than the score cutoff s_k for the library entry, it is accepted that fragment i belongs to class k (note that it is possible for a fragment to belong to multiple classes). Utilising the known many-to-one correspondence between residues and fragments, each residue that belongs to at least one fragment belonging to class k is also considered to belong to class k, relative to the class representative.

For each residue belonging to class k, the 'minimum' score is retained and reported in the same manner as for residue alignments. This score is also used in the generation of colour scripts for each fragment class. This provides information complementary to the categorisation alone, allowing assessment by visualisation of the degree of structural conservation/flexibility of the categorised fragments. An example of this is illustrated in Figure 68, where classified residues are coloured according to similarity between local environment and class representative. Firstly, note that not all, nor only, residues belonging to SSEs (according to *PyMOL*'s cartoon representation) are aligned. This is exactly as intended, since only structure-based class criteria must be satisfied in order for fragments to be aligned. Also, note that there are many more identified α-helical fragments that are extremely similar to the class representative, in comparison with the β-strand fragments, as is immediately apparent by the residue colouring. This agrees with the previous assertion that strands, having a tendency to bend and twist away from the representative fragment, generally have greater conformational flexibility than helices.

Note that any particular residue may belong to multiple classes, which may be undesirable in some situations. Consequently, any given residue is then considered to belong to the class with the lowest (best) Procrustes dissimilarity score, out of all classified fragments that it belongs to. This results in all residues either belonging to one unique class from the library, or being unclassified. This leads directly to the representation of the amino acid chain as a structural sequence, based on the alphabet defined in the fragment library.

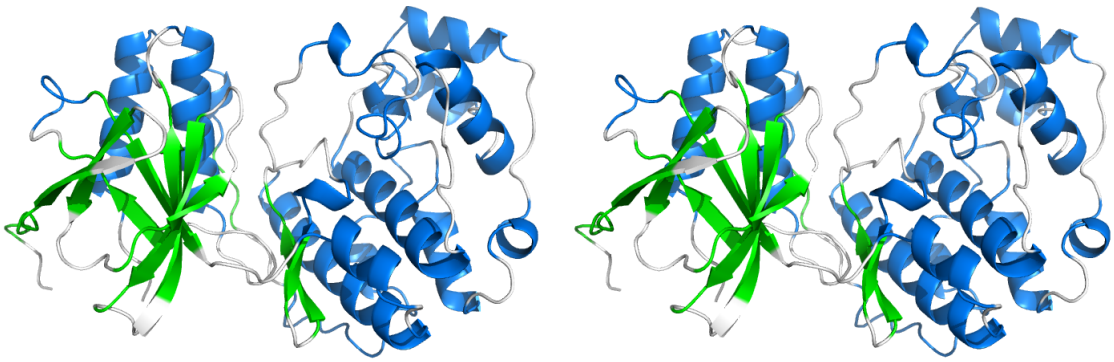


Figure 69: Fragment type identification results from the self-alignment of the MAP kinase 3dt1 (Herberich et al., 2008), using the same library configuration as in Figure 68. In the stereo illustration, residues are coloured according to the colours specified in the fragment library configuration file, where the helix class is coloured blue, and the strand class green. Unclassified residues are coloured white.

In cases where ordinary *ProSMART* alignment is performed on two protein chains, the fragment library is used to subsequently perform fragment type identification on the resultant fragment alignment, as described above. This allows the achieved residue alignment to be rewritten in terms of the implied structural sequence. This information may be used for structural analysis and alignment validation, as appropriate. Colour scripts are also generated using this categorisation, with residues coloured according to specification in the fragment library configuration file. If two different chains are compared, then only the aligned residues are categorised/coloured. Alternatively, all residues can be categorised/coloured by performing a self-alignment, as shown in Figure 69.

3.4 Examples Demonstrating Functionality of ProSMART RESTRAIN

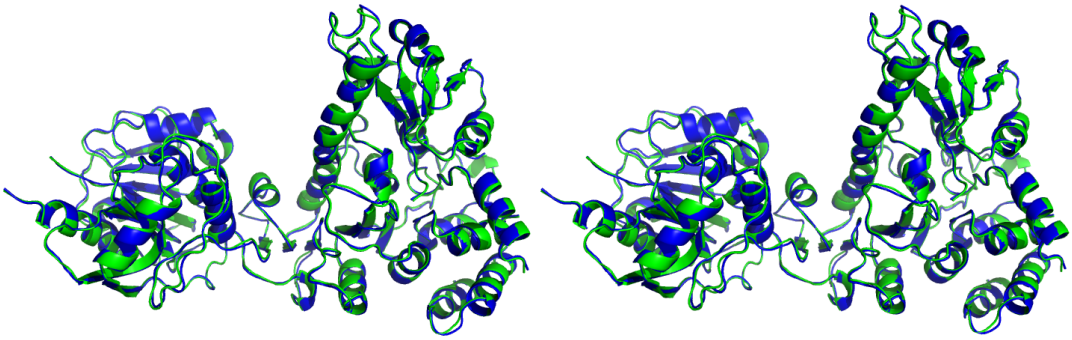
3.4.1 Use of Generated External Restraints in Refinement

In protein crystallography, it is sometimes not known whether observed dissimilarities between similar structures are biologically relevant, artefacts of the crystallisation process, or a consequence of poor quality data or refinement. In such cases, it may be of interest to reduce the uncertainty due to poor refinement. Any improvement in refinement may aid structural analysis, since increasing the reliability of atomic positions would perhaps result in observed dissimilarities being actual rather than erroneous.

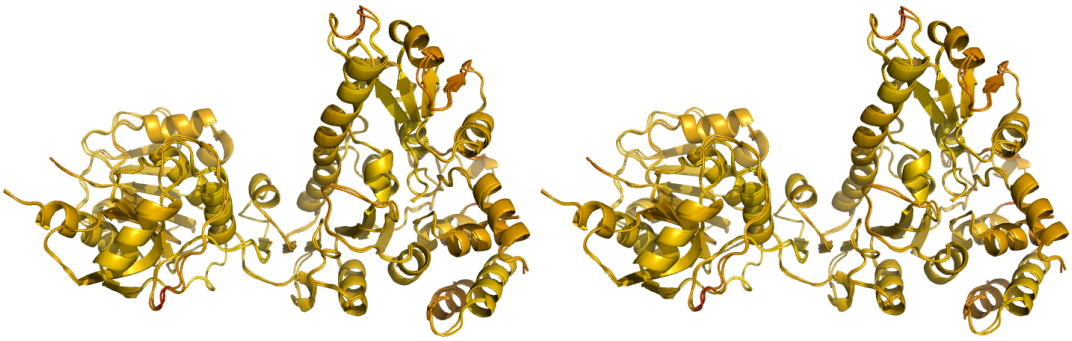
Potential use of external restraints generated by *ProSMART* is demonstrated using a simple example of a bluetongue virus VP4 enzyme (Sutton et al., 2007), with refinement performed by *REFMAC5* (Murshudov et al., 2011, 1997). The re-refinement of the 3.4Å structure with PDB code 2jha is attempted using external restraints from the sequence-identical 2.5Å structure 2jhp. The consideration of altering some major parameters illustrates behaviour that might be expected in such simple cases. However, no implication is made regarding the practical suitability of the achieved results or refinement methodology; refinement is automated, largely using default settings, and no attempt is made to achieve a 'good' final model. Refinement quality of local regions is not considered, given the purpose of this example, being interested only in the qualitative effect of external restraints on global statistics.

Both target (2jha) and external reference (2jhp) structures comprise one chain, and were crystallised in the same space group. As can be seen in Figure 70, they share very similar global conformations, with no major conformational change. However, the backbone trace is not identical. At a local level, differences in backbone conformation are detected in a few regions, and many residues are observed to have side chains in different conformations. For the purpose of this example, it is unknown/unassumed whether these observed differences are actual, due to unoptimal refinement, or due to other causes such as conformational flexibility.

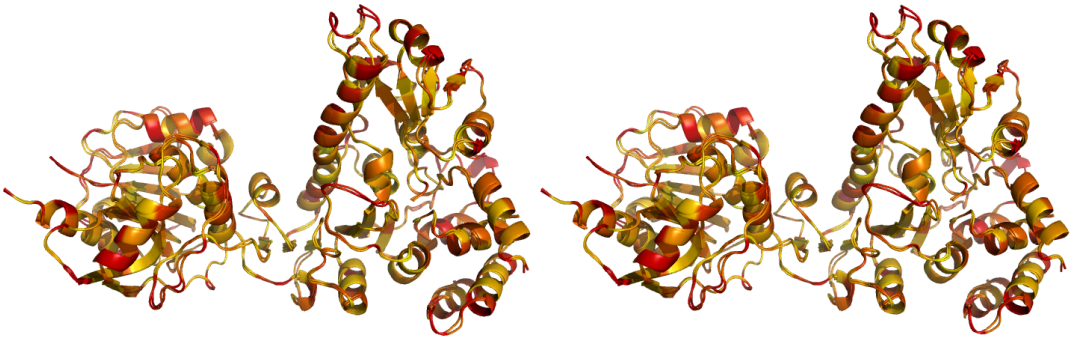
Figure 71 shows statistics resulting from the refinement of 2jha, starting with the deposited



(a) Stereo superposition.



(b) Stereo superposition, coloured by 'minimum' main chain score.



(c) Stereo superposition, coloured by local side chain RMSD.

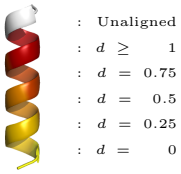
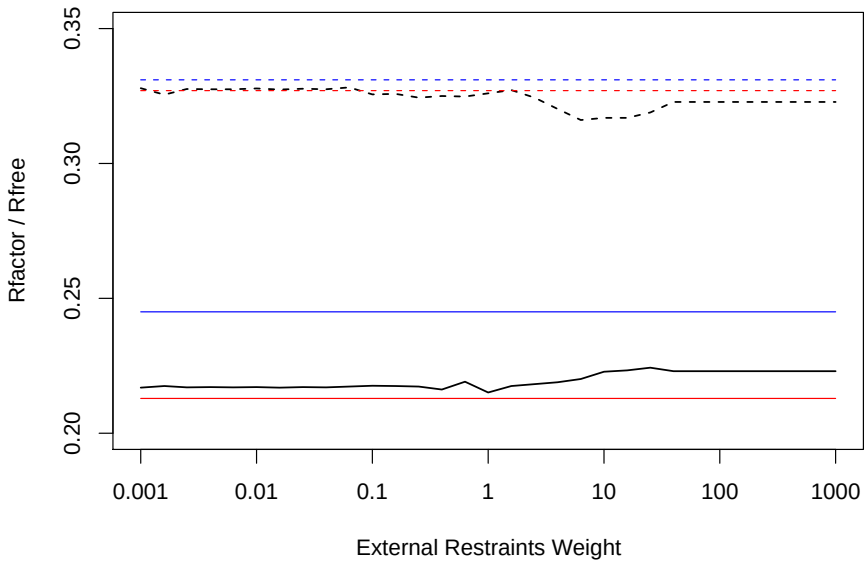
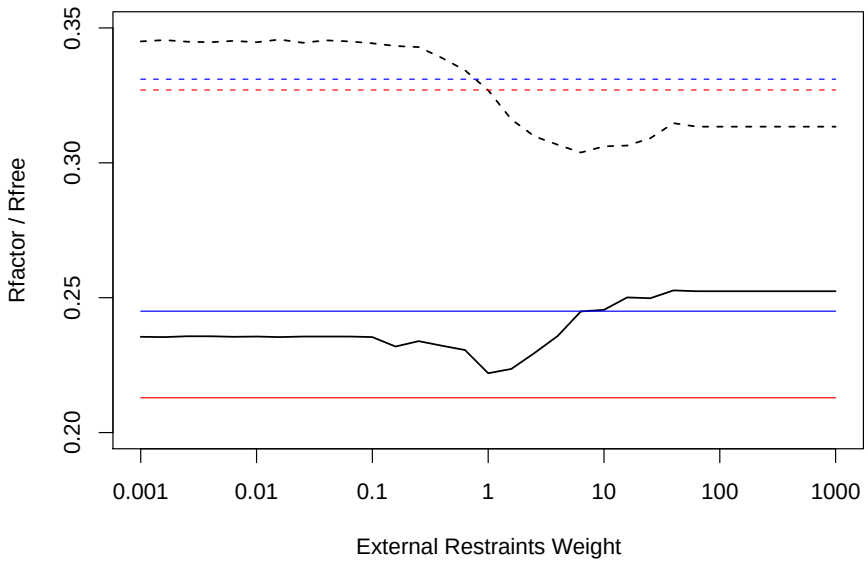


Figure 70: Illustration of the superposition and structural conservation of sequence-identical 2jha and 2jhp. Superposition is shown (a) with 2jha and 2jhp coloured blue and green, respectively. In (b) and (c), residues are coloured according to scores; yellow for conserved, red for relatively less conserved. Main chain dissimilarity is depicted using the 'minimum' Procrustes score (b). Side chain RMSD is also displayed (c), relative to the local fragment-based coordinate frame. Superposition and colour code scripts were generated by *ProSMART ALIGN* (using default parameters).



(a) Main chain external restraints only



(b) Main and side chain external restraints

Figure 71: Statistics from the re-refinement of the structure with PDB code 2jha. In each graph, solid lines represent the R-factor, and dotted lines R_{free}. Blue lines represent the original statistics quoted at deposition. Statistics achieved after ten *REFMAC* refinement iterations are shown with (black) and without (red) using external restraints generated using PDB code 2jhp. Where external restraints are used, statistics are shown for a range of refinement weights, on a logarithmic scale for clarity. The upper graph (a) shows results when using external restraints only between main chain atoms; in the lower graph (b), side chain atoms are also used for external restraint generation. Results were calculated for weights $\log(w) = 0.2x \log(10)$ for $x = -15, \ldots, 15$.

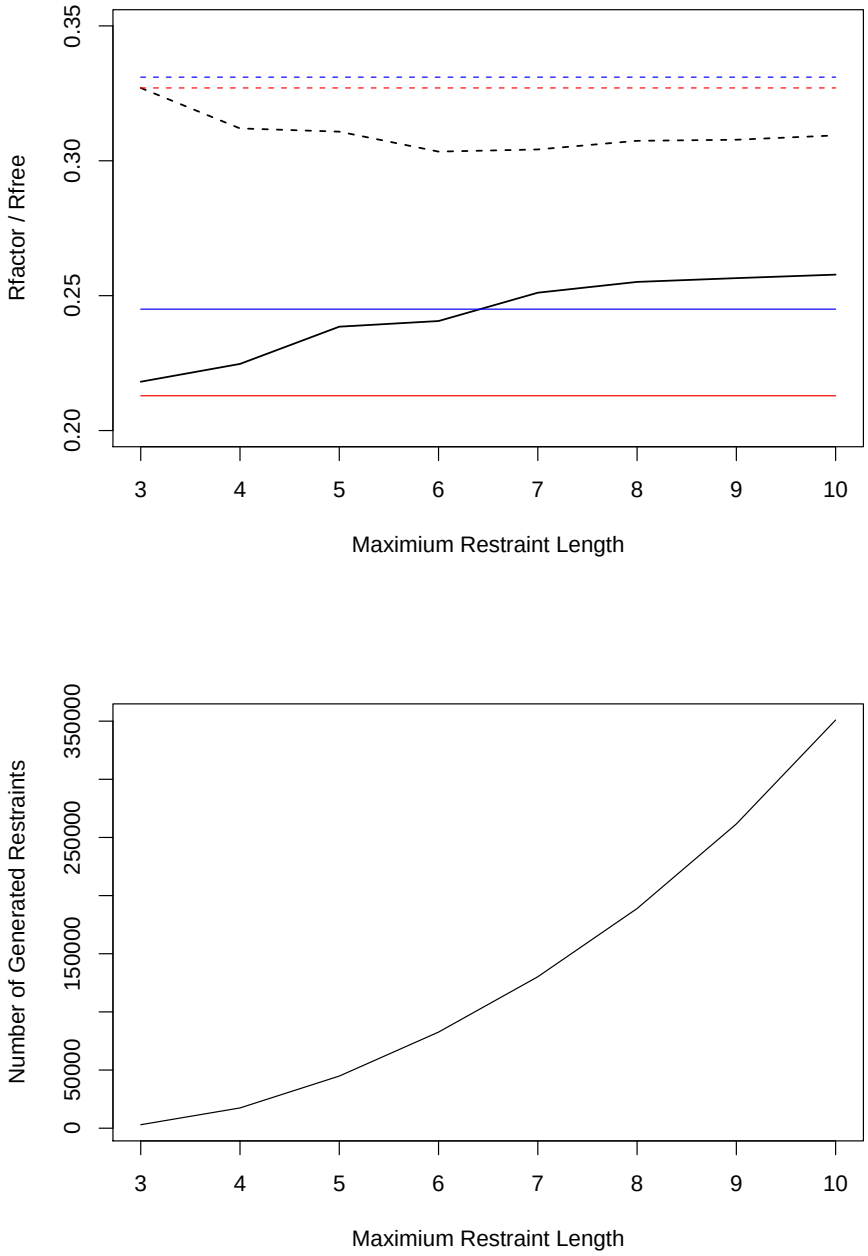


Figure 72: Statistics from the re-refinement of the structure with PDB code 2jha. In the upper graph, solid lines represent the R-factor, and dotted lines R_{free}. As in Figure 71, blue lines represent the original statistics quoted at deposition, and statistics achieved after ten *REFMAC* refinement iterations are shown with (black) and without (red) using external restraints generated using PDB code 2jhp. Where external restraints are used, a refinement weight of 7 was enforced, and restraints were generated using main and side chain atoms. Statistics are shown resulting from the use of different restraint ranges, by varying the maximum restraint length (denoted by the parameter $r_{\max}$ in §2.4), shown at integer intervals. The lower graph shows the corresponding frequencies of generated external restraints.

model. The deposited model was refined using *REFMAC* 5.2.0019. All re-refinement was performed using *REFMAC* 5.6.0081, using default parameters unless otherwise specified. Refinement of the model without using external restraints yielded decrease in both R and R_{free}, although the difference between them increased.

Distance-dependent sigmas were generated when using external restraints. When only main chain external restraints were used, the difference between R and R_{free} was reduced, particularly for higher weights. A noticeable reduction in R_{free} was observed for a small range of intermediate weights. This behaviour was more pronounced when external side chain restraints were also used, resulting in a substantial decrease in R_{free} and $R_{\text{free}} - R$. This example illustrates the importance of choice of weight to control the influence of external restraints relative to other components in the likelihood function.

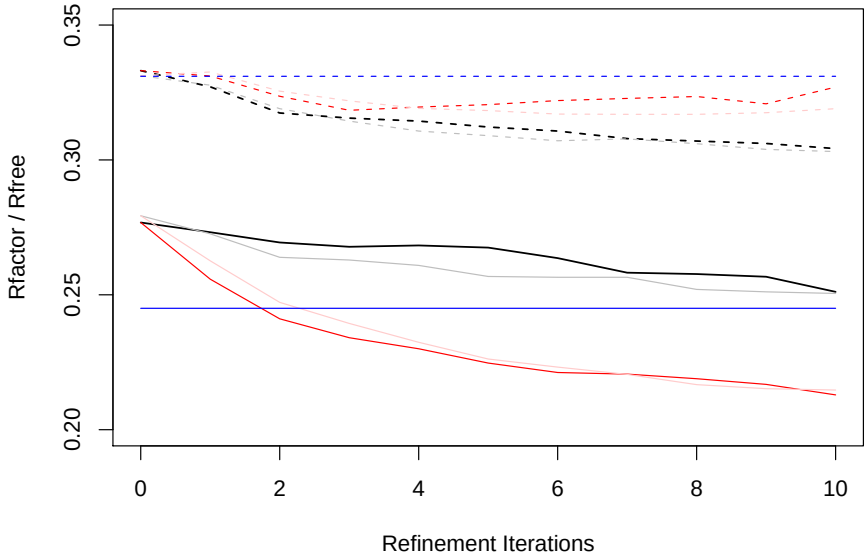
Such situations, where refinement statistics appear improved, might suggest local relative atomic positions to be largely conserved between the two structures, and that information contained in the higher-resolution structure might be used to improve positions of both main chain and side chains in the lower-resolution structure. In other cases, side chain restraints may not be appropriate if the structure of the external reference is less similar to the target.

The integer weight that minimised R_{free} was $w_{\text{ext}} = 7$, when using both main and side chain restraints. This parameter was fixed for further analysis, although note that it is not implied that this would be a good general strategy for parameter selection.

Results arising from consideration of a variety of maximum restraint lengths $r_{\max}$ are shown in Figure 72. Using a low $r_{\max}$ results in relatively few restraints being generated, thus having little effect on refinement. Note that external restraints between atom-pairs separated by less than three bonds are not generated. As $r_{\max}$ increases, more restraints are generated, and the external restraints have a greater impact on refinement. In this case, the difference between R and R_{free} decreases as $r_{\max}$ increases, up to approximately $r_{\max} = 7$, after which point both R and R_{free} steadily increase. More generally, longer restraints are less tolerant to conformational change or flexibility, influencing tighter rigid structural agreement. The effect of this is not strong in this case, since the structures are reasonably rigidly conserved at the global level already. However, this may have stronger negative effects in cases where conformational changes are present. The use of distance-dependent sigmas may help to reduce this effect.

We now further consider refinement when using $w_{\text{ext}} = 7$ and $r_{\max} = 7$. Note that more appropriate parameter values may exist, and other parameters can be adjusted. For example, such other parameters/approaches that may be considered with *ProSMART RESTRAIN* include: minimum restraint length; method of sigma estimation; removal of outliers; exclusion of restraints between residues with poor main chain or side chain alignment scores; sigma modification using B-factors; and consideration of short-range external restraints. However, this simple parameter selection is deemed sufficiently suitable for the purposes of this example.

Statistics resulting from the first ten refinement cycles are shown in Figure 73. Without using external restraints, R_{free} is reduced below the original value, although the difference between R and R_{free} also increases substantially, suggesting unstable refinement (note that different behaviour may be observed if non-default parameter values were selected). When using external restraints, R_{free} is lower than both the original value, and that achieved from re-refinement without external restraints. Furthermore, both R and R_{free} appear to steadily decrease as the refinement iterates, with their differential reasonably stable. This would suggest improved reliability of the achieved model when using external restraints, in this case. Unsuccessful application of external restraints, e.g. due to unsuitable selection of reference structure(s), atom-pairs, estimated sigmas, or chosen weight parameters, would generally result in both R and R_{free} being increased.



Model	R-factor	R-free
Original Deposited 2jha	0.245	0.331
External Reference Structure 2jhp	0.229	0.289
Refined Without External Restraints	0.2139	0.3270
Refined With External Restraints	0.2511	0.3042
Using Hydrogens, Without External Restraints	0.2147	0.3190
Using Hydrogens, With External Restraints	0.2505	0.3031

Figure 73: Statistics from the re-refinement of the structure with PDB code 2jha, calculated for each of the first ten *REFMAC* refinement iterations. In the graph, solid lines represent the R-factor, and dotted lines R_{free}. Blue lines represent the original statistics quoted at deposition. Black and red lines correspond to statistics resulting from refinement with and without using external restraints, respectively. Where external restraints are used, a refinement weight of 7 was enforced, and restraints up to 7Å were generated using both main and side chain atoms, using 2jhp as the external reference structure. Faint lines (grey and pale red, for with and without external restraints) represent the corresponding results when hydrogens were generated and used in refinement. Statistics achieved after ten iterations are displayed in the table.

Whilst being good indicators, the considered refinement statistics alone are not sufficient to unambiguously deduce model improvement, since it is necessary for the resultant model to be chemically feasible, further to agreeing with the experimental data. Consequently, model validation includes verification that the geometry is reasonable. For this, we use *MolProbity* (Chen et al., 2010; Davis et al., 2007) to assess whether there is any improvement in model geometry; results are shown in Table 2.

	Target Structure (2jha)	External Reference Structure (2jhp)
Clashscore (percentile)	49.05 (56^{th})	23.06 (60^{th})
Poor rotamers	17.34%	9.23%
Ramachandran outliers	4.63%	2.48%
Ramachandran favoured	79.67%	92.23%
C_β deviations > 0.25Å	11	9
MolProbity score (percentile)	3.85 (37^{th})	3.06 (33^{rd})
Residues with bad bonds	0.49%	0.0%
Residues with bad angles	1.63%	2.12%

	2jha Re-Refined	2jha with External Restraints
Clashscore (percentile)	45.5 (59^{th})	31.39 (78^{th})
Poor rotamers	12.92%	11.07%
Ramachandran outliers	4.13%	2.64%
Ramachandran favoured	80.83%	91.57%
C_β deviations > 0.25Å	10	9
MolProbity score (percentile)	3.7 (51^{st})	3.27 (77^{th})
Residues with bad bonds	0%	0%
Residues with bad angles	1.14%	0.82%

	Re-Refined w/Hydrogens	External Restraints w/Hydrogens
Clashscore (percentile)	41.26 (64^{th})	28.33 (86^{th})
Poor rotamers	11.99%	9.59%
Ramachandran outliers	4.63%	2.98%
Ramachandran favoured	81.98%	91.57%
C_β deviations > 0.25Å	5	7
MolProbity score (percentile)	3.62 (55^{th})	3.19 (80^{th})
Residues with bad bonds	0%	0%
Residues with bad angles	0.49%	0.49%

Table 2: *MolProbity* summary statistics corresponding to various models: original structures of 2jha and 2jhp (top); 2jha after ten *REFMAC* refinement cycles with and without external restraints, with (bottom) and without (middle) generating and using hydrogen atoms during refinement. Colour coding represents quality relative to resolution, as appropriate; all results were achieved using the *MolProbity* online server (Chen et al., 2010).

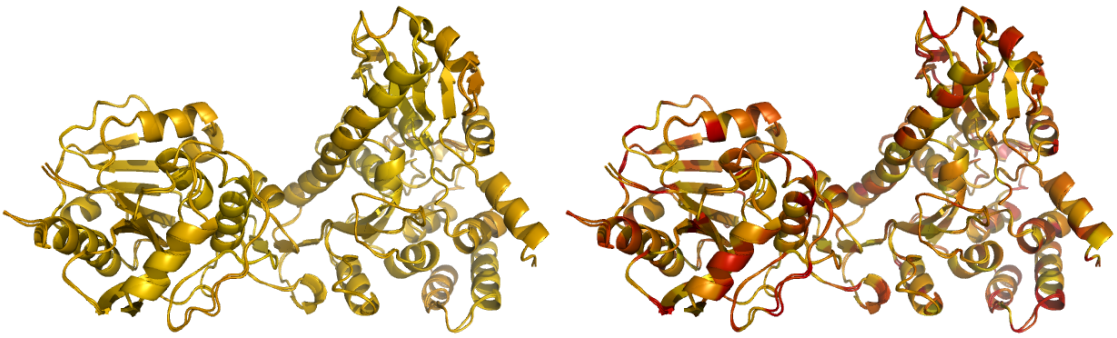
Without using external restraints, running ten cycles of default *REFMAC* refinement, all statistics show improvement over that of the original model. This may be indicative of improvements in refinement methods in recent years. Of course, the relative increase in $R_{\text{free}} - R$ means that the model is not unambiguously improved; a more reliable model may be achieved by adjusting refinement parameters.

The use of external restraints results in substantial further improvement over the model refined without external restraints, for all statistics. Since *MolProbity* uses assumed knowledge of the position of hydrogen atoms in validation, refinement with and without external restraints was repeated with hydrogen atoms generated and used in refinement by *REFMAC* (using the '`MAKE HYDROGEN ALL`' keyword; results shown in Figure 73 and Table 2). Using hydrogen atoms, *MolProbity* summary statistics were further improved in all tests, except for an increase in Ramachandran outliers for both models. The model refined with external restraints was found to have more favourable statistics, for all tests apart from the number of C_β deviation outliers.

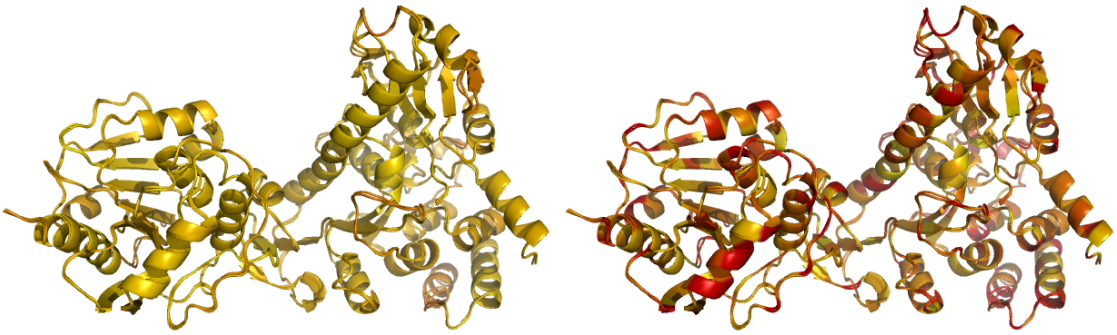
Combined with knowledge of the reduction in R_{free} and $R_{\text{free}} - R$, these results would suggest that the use of external restraints can result in an improved model, in this case. However, local regions should be manually inspected to ensure local suitability of the use of external restraints, in order to identify any serious artefacts that may arise due to bias towards the reference structure (note that particular residues can be excluded from external restraint generation). Generally, further iterations of manual and automatic refinement (with optimised parameters), aided by external restraints, might lead to a greatly improved model. Of course, the degree of any improvement due to external restraints will be limited by the quality of the reference structure.

Figure 74 illustrates a simple structural comparison of the 'final' model achieved by re-refinement with external restraints (with $w_{\text{ext}} = 7$ and $r_{\max} = 7$) and the original 2jha model; 2jha re-refined without external restraints; and 2jhp. It is apparent that the local backbone trace diverges from the original 2jha model, and is pulled tightly towards the conformation of 2jhp. However, two regions in particular are relatively dissimilar to 2jhp (coloured red in Figure 74c, left), adopting conformations more similar to those in both the original 2jha model and the model re-refined without external restraints. The differences in these regions may be due to actual differences in the crystal, or may be due to inaccuracies in the models (we shall not investigate which). The matter of import is that these regions appear to have not been over-restrained towards 2jhp, allowing the external restraints to highly influence conformation where appropriate, whilst at the same time allowing conformational differences in less conserved regions (compare with Figure 70). In practice, this would be further investigated/validated by manual inspection of the model's agreement with the electron density map.

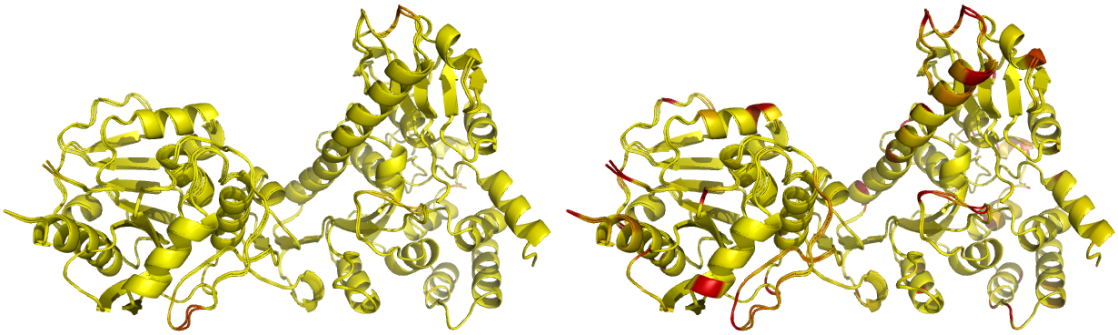
Even though the final model is extremely structurally similar to the reference 2jhp in many regions, it is important to note that the superposition of these structures visually appears to suggest more similarity to 2jha than to 2jhp, in many regions (in particular, consider the upper-left



(a) Comparison with original model



(b) Comparison with re-refined model



(c) Comparison with 2jhp

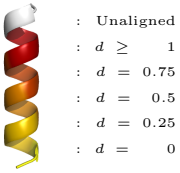


Figure 74: Superposed structural comparisons between the 'final' model of 2jha re-refined with external restraints from 2jhp and (a) the original deposited model of 2jha; (b) the model re-refined without external restraints; and (c) the higher-resolution reference model 2jhp. Structures are coloured according to *ProSMART* local alignment scores: the 'minimum' score for backbone conservation (left images), and the side chain RMSD in the local coordinate frame (right images). Red hues indicate degrees of structural dissimilarity, fading to yellow hues indicating strong structural agreement.

quadrants of images in Figure 74). This is because local structure has been restrained towards 2jhp, but global conformation has not been overly affected. Even in this case, where any global conformational change/flexibility is very subtle, this observation is encouraging, agreeing with our intentions of providing a conformation-invariant solution.

It should be acknowledged that exact results will vary depending on the particular version of *REFMAC*, and parameters used. All results shown here involved the use of robust estimation using the Geman-McClure function. Otherwise, if the ordinary least-squares function was used, then the influence of outliers may need to be down-weighted by other means in order to achieve satisfactory results. This might be achieved by some functional sigma modification, or by the removal of outliers.

Note also that examples of the re-refinement of pre-solved deposited structures, such as that considered here, may not necessarily reflect the use of external restraints in practice. It is supposed that external restraints might be useful at earlier stages of model refinement, where structural stability cannot be adequately maintained by an optimal weighting of x-ray and geometric components in cases where electron density maps are of poor quality. It is hoped that, in some such cases, external restraints might be able to stabilise local structure without introducing more bias than can be justified by the increase in reliability.

	2w72(A)	2w72(B)	2w72(C)	2w72(D)
1ydz(A)	0.245	0.428	0.264	0.414
1ydz(B)	0.422	0.247	0.424	0.258
1ydz(C)	0.236	0.415	0.234	0.403
1ydz(D)	0.429	0.263	0.433	0.264

(a) Average of aligned fragment Procrustes dissimilarity scores

	2w72(A)	2w72(B)	2w72(C)	2w72(D)
1ydz(A)	0.444	1.38	0.615	1.4
1ydz(B)	1.37	0.508	1.54	0.489
1ydz(C)	0.428	1.39	0.553	1.4
1ydz(D)	1.41	0.517	1.59	0.545

(b) Global RMSD after superposition

	2w72(A)	2w72(B)	2w72(C)	2w72(D)
1ydz(A)	97.2	41	97.9	41
1ydz(B)	41.7	97.9	41.7	97.9
1ydz(C)	97.2	41	97.9	41
1ydz(D)	41.7	97.9	41.7	97.9

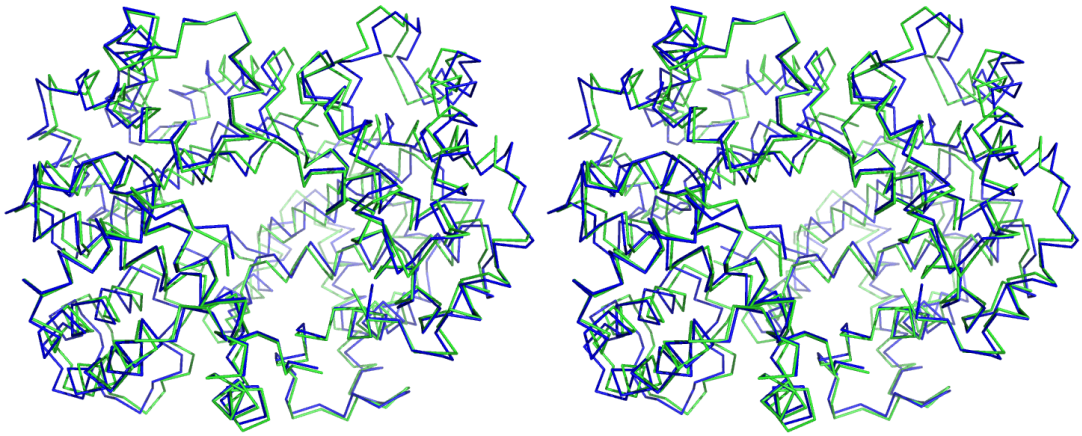
(c) Sequence identity of aligned residues

Table 3: Global statistics resulting from the comparison of all chains in 1ydz with those in 2w72.

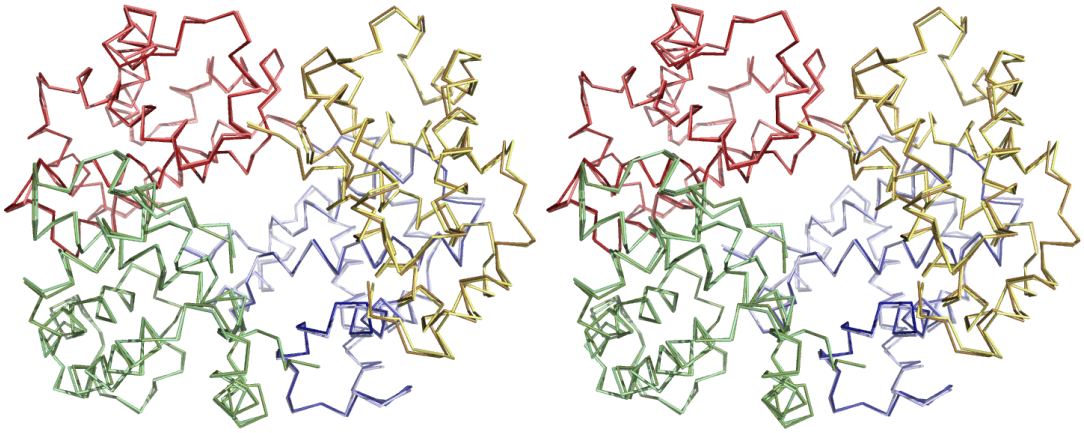
3.4.2 Use of External and Helix Restraints for Multiple Chains

We now consider the re-refinement of the 3.3Å model of human hemoglobin with PDB code 1ydz (Kavanaugh et al., 2005). The use of external restraints is demonstrated using fragment restraints from an ideal helix, and also external restraints from a 1.07Å reference structure 2w72 (Savino et al., 2009) sharing approximately 98% sequence identity with the target.

Both target (1ydz) and external reference (2w72) structures comprise four subunits; two alpha chains and two beta chains. As can be seen in Table 3, the global statistics output by *ProSMART* indicate that in both structures the alpha chains (A and C) are locally and globally very similar, and similarly for the beta chains (B and D). Note that it is also possible to infer the alpha chains to be reasonably similar to the beta chains, although to a lesser degree, given their scores. This

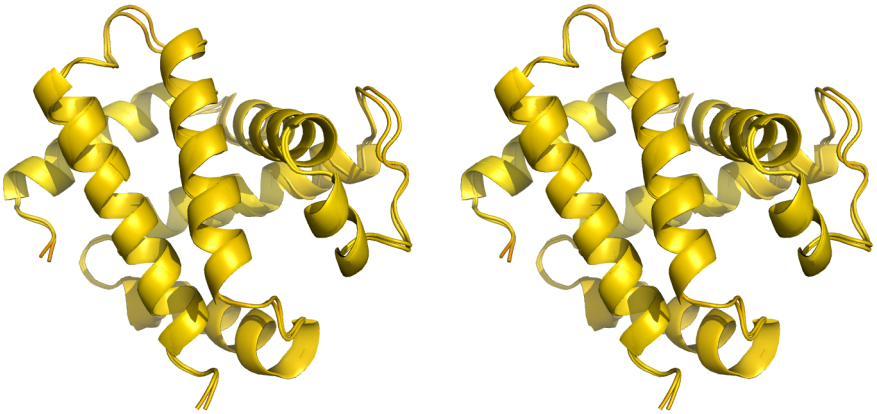


(a) Stereo superposition of four chains.

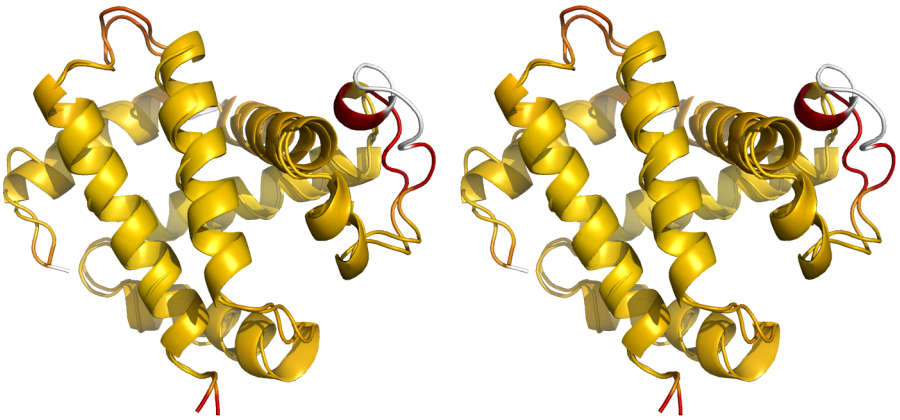


(b) Stereo superposition of each individual chain.

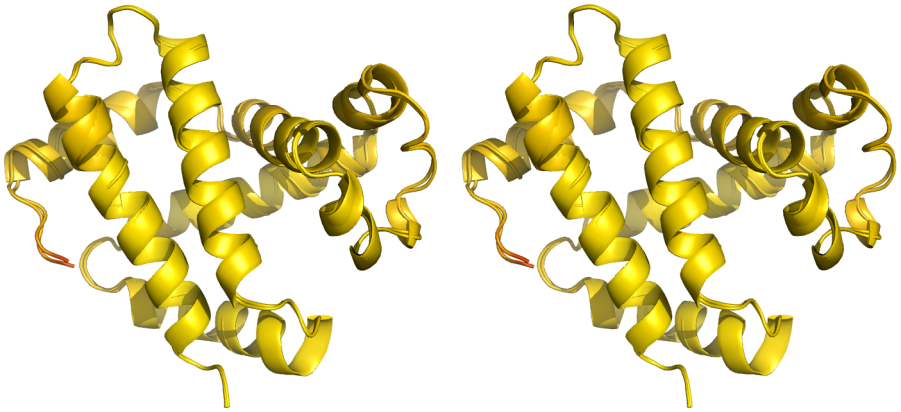
Figure 75: Superposed structures of the four subunits in 1ydz and 2w72. Structures are superposed (a) maintaining relative positions of the four subunits for both 1ydz (blue) and 2w72 (green) (using *PyMOL*; Schrödinger, LLC, 2010; DeLano, 2007), and (b) by superposing each chain-pair separately, in the coordinate frame of 1ydz (using *ProSMART*). Chains A, B, C and D are coloured red, green, blue and yellow, respectively, with chains from 1ydz shown in a lighter, and those from 2w72 in a darker hue.



(a) Stereo superposition of 1ydz(A) and 2w72(A).



(b) Stereo superposition of 1ydz(A) and 2w72(B).



(c) Stereo superposition of 1ydz(B) and 2w72(B).

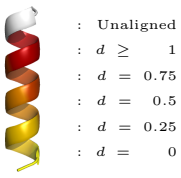
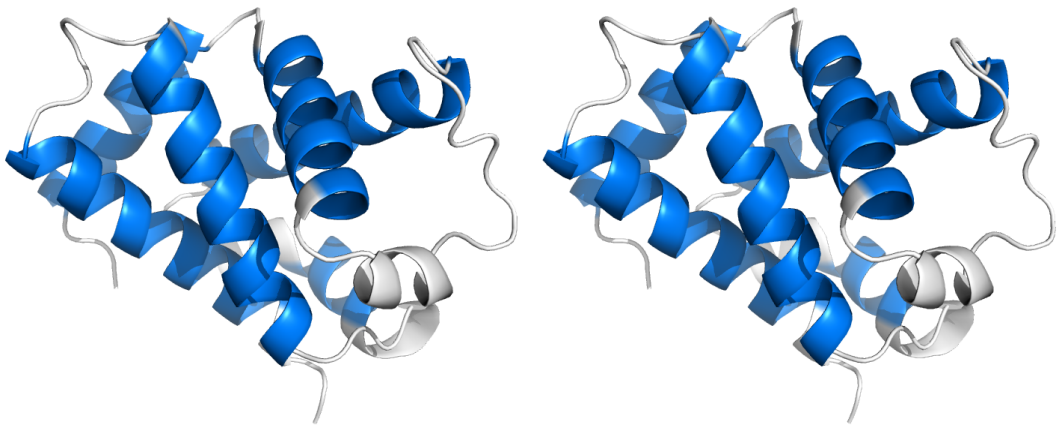


Figure 76: Superposed structures of chains A and A (left), A and B (middle), and B and B (right) from 1ydz and 2w72, respectively. Structures are coloured according to the 'minimum' score for backbone conservation. Red hues indicate degrees of structural dissimilarity, fading to yellow hues indicating strong structural agreement.

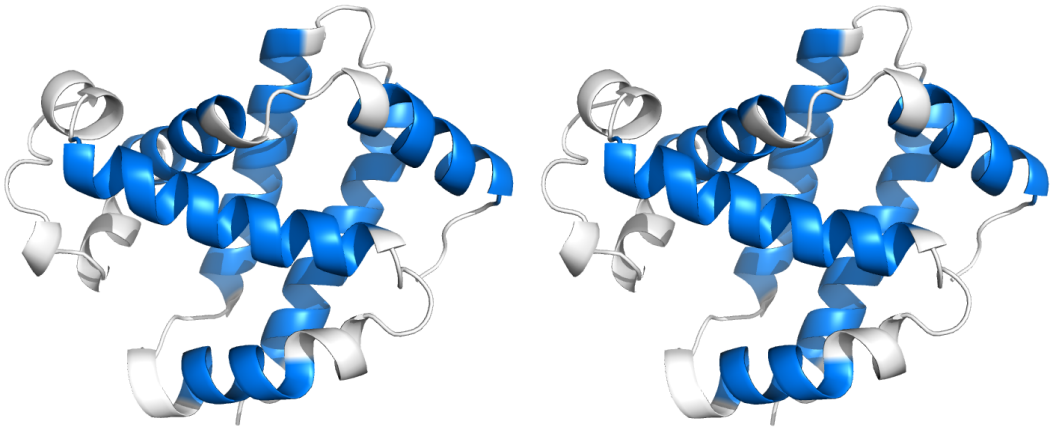
behaviour is confirmed by Figures 75 and 76. Also, Figure 75 demonstrates how the intra-chain rigid structural conservation is high for each corresponding subunit, whilst their relative positions are slightly different in the two crystals. Figure 76 shows that the global arrangement of the alpha and beta subunits is similar, whilst the backbone is less locally conserved in some loop regions.

In accordance with the default automated protocol, the chains with the most favourable local scores are selected for restraint generation. This results in restraints for chains A, B, C and D in 1ydz being generated using the reference chains A, B, C and B in 2w72, respectively. Only main chain restraints were generated, as side chain restraints did not have a positive influence on refinement in this case. Helix restraints were generated from the ideal helix fragment in the *ProSMART* library, using a fragment length of $n = 5$ residues and a Procrustes score threshold of 0.3Å, and were applied using a weight of $w_{\text{ext}} = 6$ (see Figure 77). Selecting different parameter values would result in the generation of different restraints, although the effect of this is not investigated here.

Figure 78 displays statistics for the first ten refinement cycles. Since the difference between R and R_{free} for the original model is large, any reductions in R_{free} and $R_{\text{free}} - R$ would be desirable

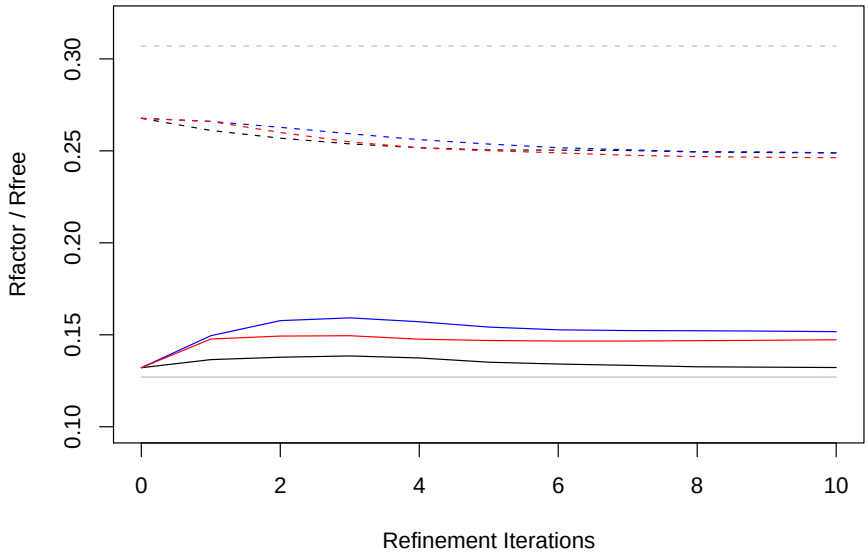


(a) Stereo view of alpha chain A from 1ydz.



(b) Stereo view of beta chain B from 1ydz.

Figure 77: Alpha chain A (a) and beta chain B (b) from 1ydz. Residues identified as sufficiently similar (using a Procrustes score threshold 0.3Å) to the ideal helix are coloured blue; other residues are coloured white. A fragment length of $n = 5$ was used.



Model	*R*-factor	*R*-free
Original Deposited 1ydz	0.127	0.307
External Reference Structure 2w72	0.129	0.153
Refined Without External Restraints	0.1322	0.2489
Refined With External Restraints	0.1473	0.2463
Refined With Helix Restraints	0.1517	0.2489
Refined With External and Helix Restraints	0.1529	0.2474

Figure 78: Statistics from the re-refinement of the structure with PDB code 1ydz, calculated for each of the first ten *REFMAC* refinement iterations. Solid lines represent the *R*-factor, and dotted lines R_{free}. Grey lines represent the original statistics quoted at deposition. Other lines correspond to statistics resulting from refinement: without external restraints (black); with helix restraints (red) and with external restraints from 2w72 (blue). Lines corresponding to refinement with both helix and external restraints are not shown, due to being very similar to those using external restraints only (blue). A refinement weight of 6 was used for external restraints, and restraints up to the default 6Å were generated using only main chain atoms. Statistics achieved after ten iterations are displayed in the table.

due to the implied increase in reliability. Re-refinement without external restraints results in a noticeable reduction in R_{free} and an increase in R, suggesting potential model improvement by using modern refinement methods. The use of helix restraints reduced R_{free} by a small amount, and increased R by a substantial amount, suggesting a further increase in reliability. Main chain external restraints from 2w72 further increased R, but maintained an increased R_{free} equal to that observed without using external restraints. The use of both external restraints from 2w72 and helix restraints did not result in substantially different results to without the helix restraints, although $R_{\text{free}} - R$ was slightly reduced.

The resultant models after refinement were then validated using *MolProbity*, as shown in Table 4. Results suggest the re-refined model without external restraints to be greatly improved over the original model. The use of helix restraints gave similar results. The use of external restraints appears to improve the model further, apart from having a greater number of poor rotamers and C_β outliers; such issues would be investigated by considering the electron density in practice.

Figures 79, 80, 81 and 82 show the intrafragment rotational dissimilarity score between the reference 2w72 and the other models (this score is more sensitive to backbone conformational changes than the minimum score). It can be seen that the model refined with external restraints is more similar to the reference structure than any other model, as would be expected. However, definite dissimilarities are observed between this model and 2w72, indicating that the external restraints are not too strong in these regions, as appropriate (as observed in the previous example using 2jha and 2jhp). These regions may represent actual differences between the structures corresponding to 1ydz and 2w72, or may be due to model incorrectness. Consequently, these regions would in practice be targets for manual inspection/refinement.

It is interesting that the backbones of some of the helices in the model refined using helix restraints (but not external restraints from 2w72) were more similar to 2w72 than to the original model 1ydz. This suggests that the bias introduced by the helix restraints, which pulled helical fragments towards the helix attractor, also independently pulled the low-resolution structure towards the conformation of the higher-resolution structure.

Differences between the side chains in the original model 1ydz and the other models are illustrated in Figures 83, 84, 85 and 86. It is clear that there are many differences between the side chains of the target and external reference models. However, all re-refined models reasonably maintain side chain conformation of the original model. This might be expected, since no side chain restraints were imposed. However, such situations may also be due to the influence of local minima of the energy function, that cannot be escaped by *REFMAC* refinement. As an aside, most non-terminal side chains exhibiting greatest differences were lysine surface residues.

In summary, the use of helix restraints, and external restraints from a reference structure, appears to have a positive impact on refinement in this case. Both types of restraints result in improved refinement statistics, and improved geometry as indicated by *MolProbity* scores. The use

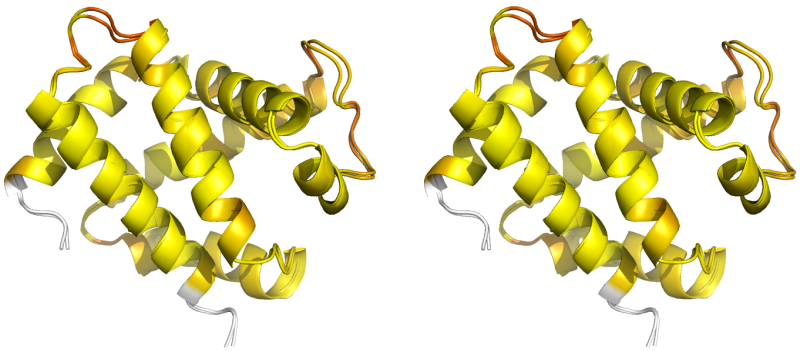
of external restraints was found to result in better *MolProbity* scores than the use of helix restraints (although the resultant models should be verified by manual local inspection). This makes sense, since helix restraints generically pull local structure towards the helix attractor, whilst external restraints from 2w72 contain information specific to the particular protein. We conclude that helix restraints may be useful, particularly when appropriate high-resolution reference structures are not available.

	Target Structure (1ydz)	External Reference Structure (2w72)
Clashscore (percentile)	39.02 (67^{th})	11.54 (19^{th})
Poor rotamers	15.22%	1.95%
Ramachandran outliers	0.0%	0%
Ramachandran favoured	96.47%	98.76%
C_β deviations > 0.25Å	25	4
MolProbity score (percentile)	3.2 (77^{th})	1.8 (35^{th})
Residues with bad bonds	0.0%	0.0%
Residues with bad angles	1.74%	0.17%

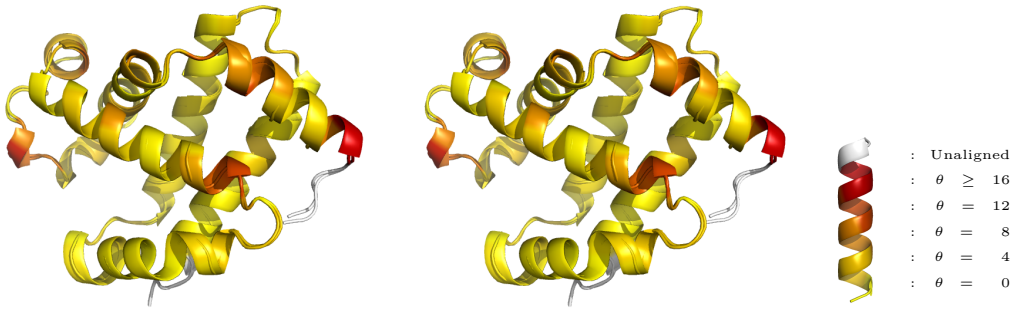
	1ydz Re-Refined	1ydz with External Restraints
Clashscore (percentile)	22 (91^{st})	19.79 (97^{th})
Poor rotamers	8.7%	9.57%
Ramachandran outliers	0.18%	0%
Ramachandran favoured	96.11%	98.76%
C_β deviations > 0.25Å	1	3
MolProbity score (percentile)	2.81 (92^{nd})	2.54 (97^{th})
Residues with bad bonds	0.0%	0.0%
Residues with bad angles	0.35%	0.0%

	Helix Restraints	External and Helix Restraints
Clashscore (percentile)	21.22 (91^{st})	21.45 (91^{st})
Poor rotamers	9.35%	9.78%
Ramachandran outliers	0.18%	0.0%
Ramachandran favoured	97.35%	98.58%
C_β deviations > 0.25Å	4	3
MolProbity score (percentile)	2.68 (95^{th})	2.58 (97^{th})
Residues with bad bonds	0.0%	0.0%
Residues with bad angles	0.35%	0.0%

Table 4: *MolProbity* summary statistics corresponding to various models: original structures of 1ydz and 2w72 (top); 1ydz after ten *REFMAC* refinement cycles with and without external restraints, with (bottom) and without (middle) also using generated helix restraints. Colour coding represents quality relative to resolution, as appropriate; all results were achieved using the *MolProbity* online server (Chen et al., 2010).

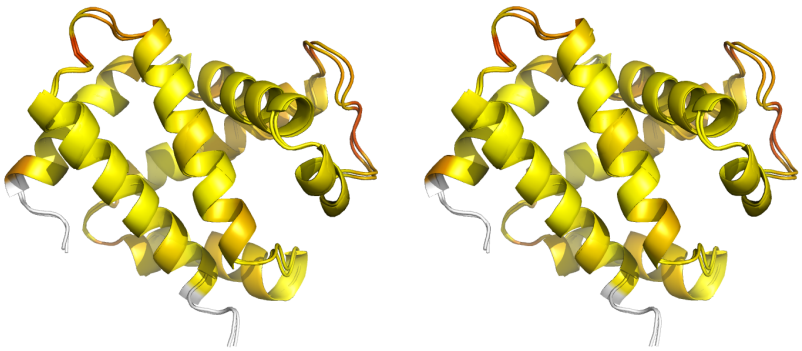


(a) Alpha chain A

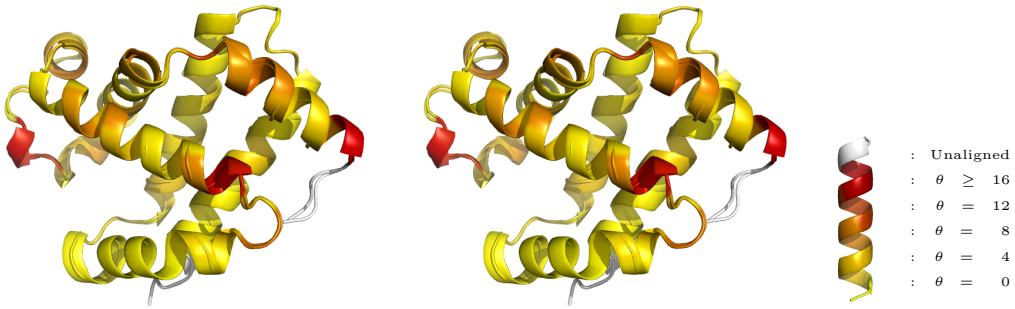


(b) Beta chain B

Figure 79: Stereo superpositions of the reference model 2w72 and the target model 1ydz for (a) alpha chain A, and (b) beta chain B. Structures are coloured by intrafragment rotational dissimilarity score. Red hues indicate degrees of backbone conformational dissimilarity, fading to yellow hues indicating rigid agreement.

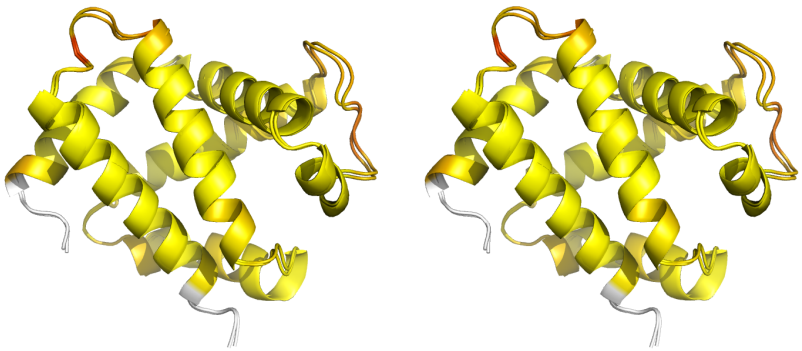


(a) Alpha chain A

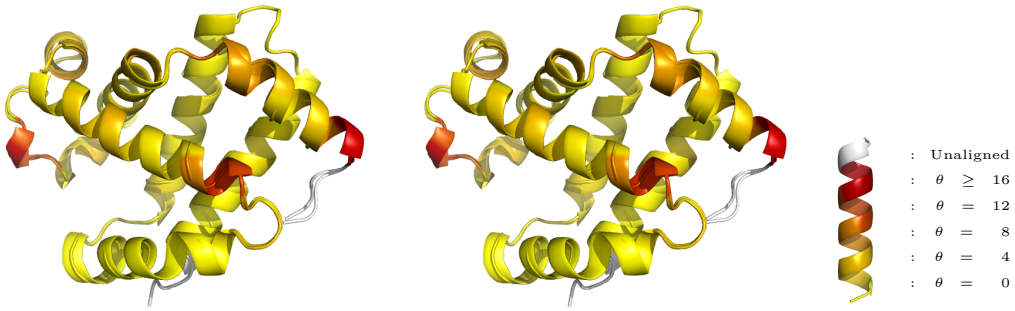


(b) Beta chain B

Figure 80: Stereo superpositions of the reference model 2w72 and the model of 1ydz re-refined without external restraints for (a) alpha chain A, and (b) beta chain B. Structures are coloured by intrafragment rotational dissimilarity score, as in Figure 79.

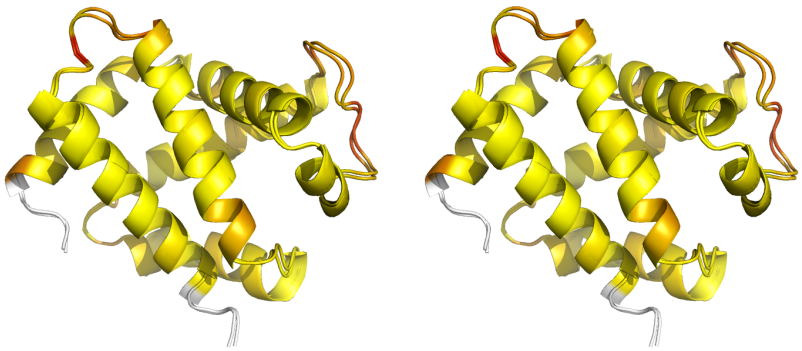


(a) Alpha chain A

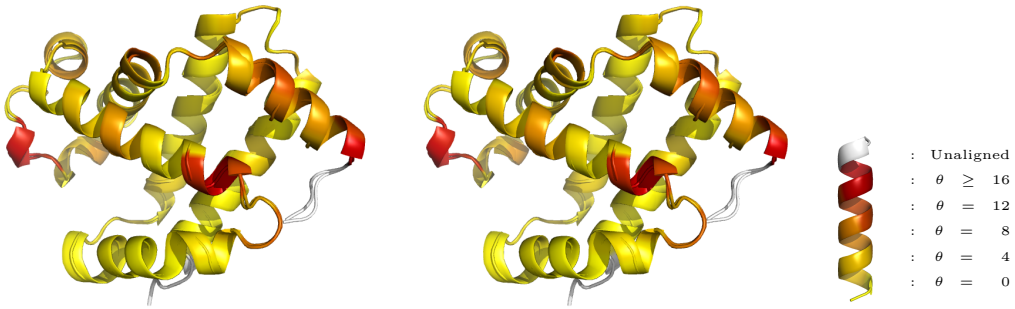


(b) Beta chain B

Figure 81: Stereo superpositions of the reference model 2w72 and the model of 1ydz re-refined with helix restraints for (a) alpha chain A, and (b) beta chain B. Structures are coloured by intrafragment rotational dissimilarity score, as in Figure 79.

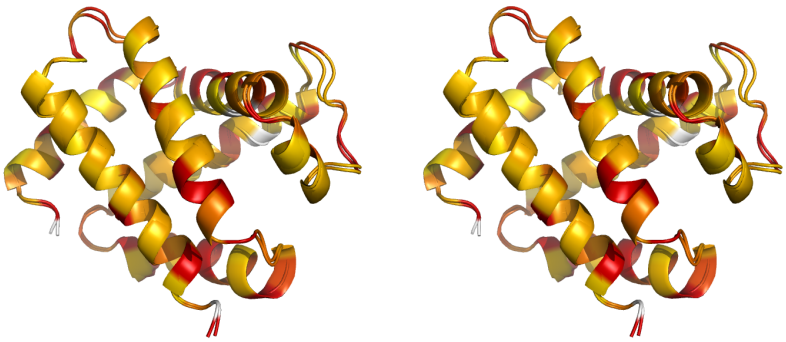


(a) Alpha chain A

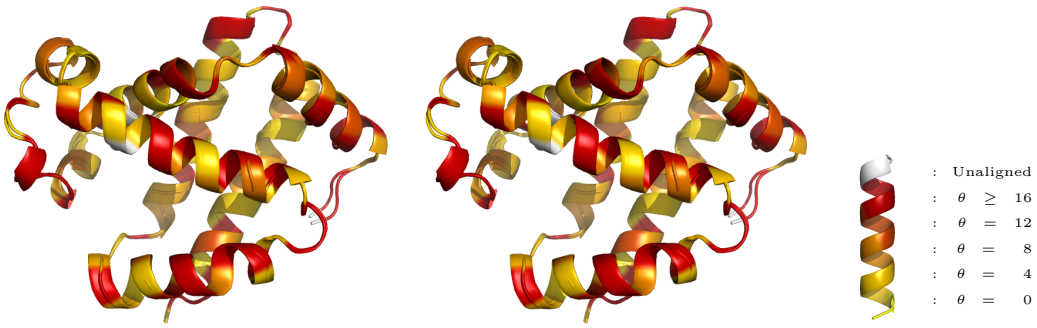


(b) Beta chain B

Figure 82: Stereo superpositions of the reference model 2w72 and the model of 1ydz re-refined with external restraints from 2w72 for (a) alpha chain A, and (b) beta chain B. Structures are coloured by intrafragment rotational dissimilarity score, as in Figure 79.

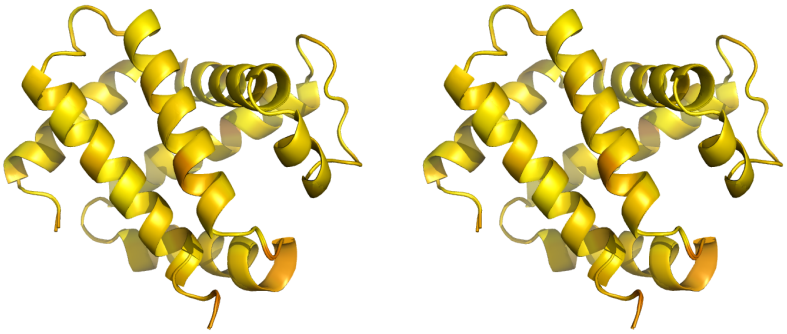


(a) Alpha chain A

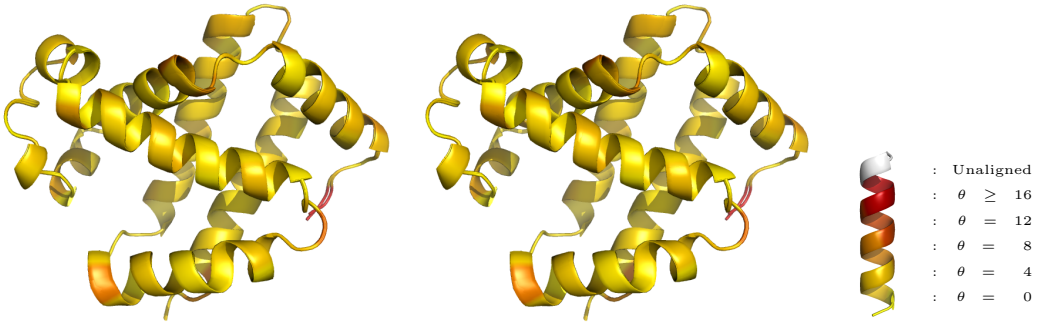


(b) Beta chain B

Figure 83: Stereo superpositions of the target model 1ydz and the reference model 2w72 for (a) alpha chain A, and (b) beta chain B. Structures are coloured according to the side chain RMSD score. Red hues indicate side chain conformational dissimilarity, fading to yellow hues indicating rigid agreement.

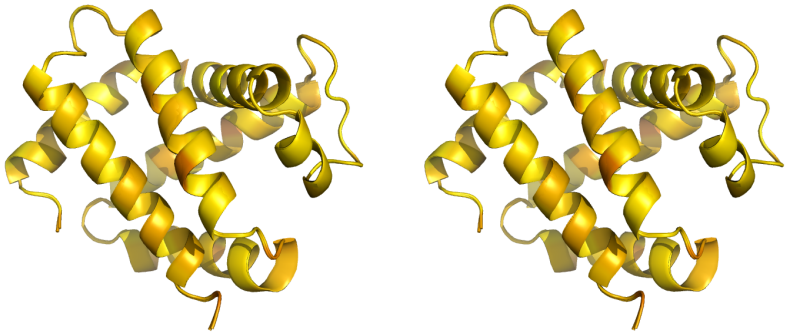


(a) Alpha chain A

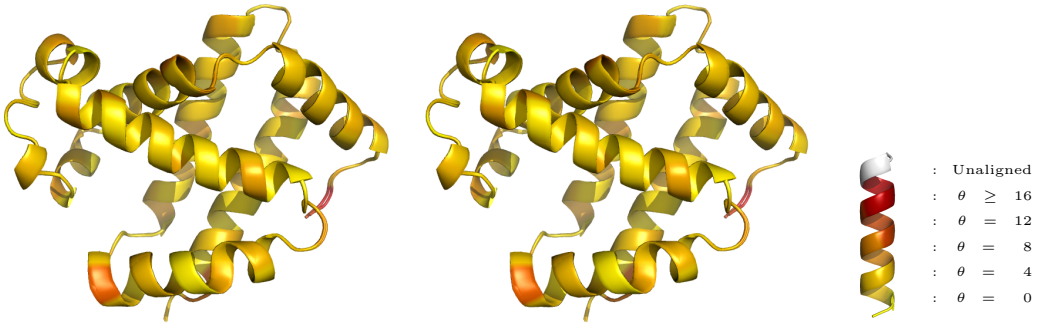


(b) Beta chain B

Figure 84: Stereo superpositions of the target model 1ydz and the model of 1ydz re-refined without external restraints for (a) alpha chain A, and (b) beta chain B. Structures are coloured by side chain RMSD score, as in Figure 83.

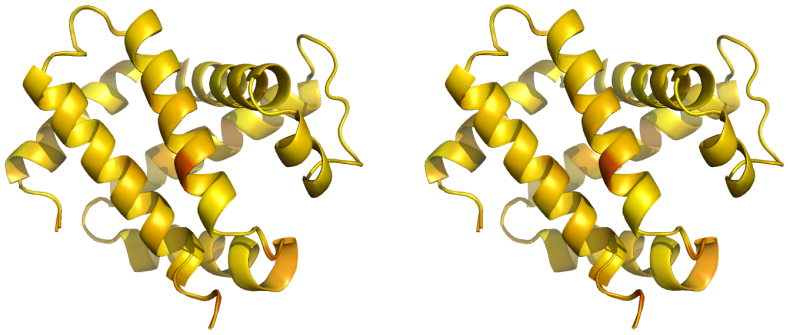


(a) Alpha chain A

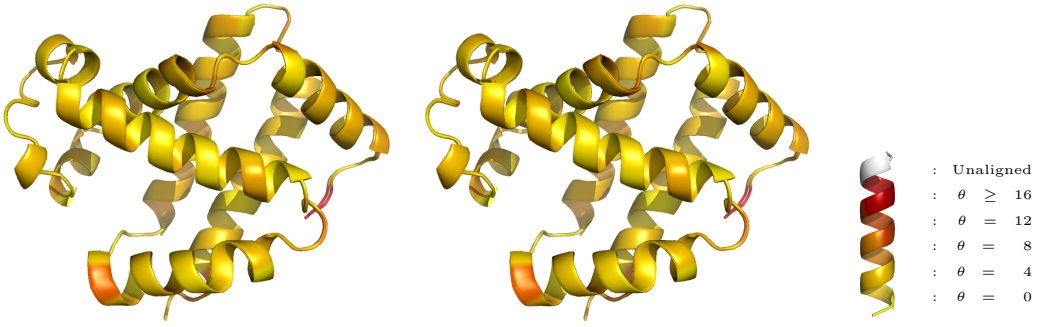


(b) Beta chain B

Figure 85: Stereo superpositions of the target model 1ydz and the model of 1ydz re-refined with helix restraints for (a) alpha chain A, and (b) beta chain B. Structures are coloured by side chain RMSD score, as in Figure 83.



(a) Alpha chain A



(b) Beta chain B

Figure 86: Stereo superpositions of the target model 1ydz and the model of 1ydz re-refined with external restraints from 2w72 for (a) alpha chain A, and (b) beta chain B. Structures are coloured by side chain RMSD score, as in Figure 83.

Chapter 4

The Scoring of Structures

4.1 Meditation on Pairwise Protein Chain Scoring

One of the major problems with the pairwise scoring of protein chains is the difficulty in assessing significance of observed scores. As discussed in §1.3.3, many existing methods use statistical significance-based similarity/dissimilarity scores, most commonly z-scores. This usually involves the transformation of a score into a standardised form. However, careful consideration must be given to the interpretation and use of such measures. Indeed, any given measure does not imply the probability of two structures being similar. One reason for this is that there is no unique definition of 'similarity'. Consequently, any such measures are very specific to both the philosophical approach and the specific heuristics employed in implementation.

Any particular significance-based measure (e.g. z-score or associated p-value) generally reports how 'good' an observed score is in relation to that expected from random chain pairs. However, the notion of 'random' in this context is usually empirically derived, and based on experimental design that may be subjective. For example, without referring to any particular studies, nor making sweeping generalisations, such data sets used to quantitatively define the distribution of scores for random chain pairs might include: all structures in the Protein Data Bank (PDB); or a subset of the PDB deemed to be non-redundant (e.g. in sequence). Various such methods have been employed (see §1.3.3). At this point, it is important to note that the PDB does not contain the structures of every possible naturally occurring protein chain (of course, this would be impossible). Therefore, it should be acknowledged that there is a degree of ambiguity to such non-theoretically derived standardisations, due to the fact that the whole population is not observed. However, this issue may in some cases be trivialised by the use of arbitrarily and/or empirically chosen parameters in defining the similarity/dissimilarity score prior to such standardisations, which may have a much greater impact on results.

In this context, care should be taken when interpreting probabilities inferred from such statistics. In general, such values indicate how the observed score compares with the presumed random

structures in the dataset. The intention is that all scores are standardised in a way that makes them comparable. They are transformed to a similar scale so that the ordering of observed scores, and the relative magnitude of scores, contains meaningful and useful information. They may in some cases imply the significance of the two structures being more similar than a random pair. However, they do not imply the probability of the two structures being similar.

One very interesting and important issue to contemplate is the effect of bias inherent in the PDB. There are multiple forms of such bias. For example, certain 'classes' are disproportionately well-represented – this bias may be reduced by the consideration of a non-redundant subset of the PDB. In contrast, certain classes that exist in nature may not be represented in the PDB – this may be due to practical issues making them difficult to study, a lack of interest in such proteins, or due to other considerations. Further complexity is introduced by convergent evolution, where certain conformations (local or global) are more favourable than others. In turn, this results in structural similarities existing, to some degree, between unrelated structures, thus introducing more bias to the problem of structural comparison. A low-level example of this is the presence of the favourable repetitive secondary structure conformations across a wide range of unrelated classes.

The effect of any bias in a dataset used to determine parameters for the generation of standardised statistics, or for the inference of statistical significance, will vary greatly from method to method. In conclusion, the usefulness and appropriateness of such standardised statistics will vary for different approaches and implementations. Consequently, it is important to understand and give proper consideration to the exact methodologies involved in all aspects of the derivation of such results. It is important for such results to be interpreted sensibly, and the limitations of such results to be acknowledged.

4.1.1 The Nature of Protein Chain Conformation Space

The composition of the polypeptide chain, combined with the nature of chemical interactions and the surrounding environmental conditions, determines the observed conformation of a protein structure. At a local level, a residue's amino acid type determines the atoms present in its side chain, which in turn determines local conformation, given the residue's surroundings (and its effect on the surrounding structure). Importantly, residues' side chains are conformationally restricted, meaning that they may only adopt certain conformational states; departure from such states is energetically unfavourable. Actual restrictions, which may be considered as restraints, are different for each amino acid. This means that conformation is highly dependent on the amino acid sequence, in turn implying conformation space to be highly heterogeneous.

Note that it is not always clear whether observed regions of high density (attractors) in fold space (Holm and Sander, 1996) are due to bias in the PDB, or due to the phenomenon of the existence of specific energetically favourable configurations. Note also that there is no unique definition of distances in fold space; each method that gives a dissimilarity between structures provides a

different representation of fold space.

Some runs of consecutive amino acids tend to adopt relatively deterministic conformations, although of course this is dependent on external factors. Perhaps the most striking example of this is the formation of helices, and their presence in unrelated structures from a wide range of classes. Similar principles also apply at a more global level; certain spatial relationships have noticeable favourability. For example, the organisation of β-strands to form parallel or anti-parallel β-sheets is observed in a wide range of unrelated classes.

Here, it is argued that helices (and similarly for strands, etc.) should ideally not be considered a discrete group, since all conformations are unified under a common fundamental framework. For example, whilst helices are stabilised by hydrogen bonds, the underlying rules governing chemical interactions are no different for helices than for any other fragment of structure. Since secondary structural elements are traditionally classified, this suggested approach of considering conformation space to be continuous may be contrary to intuition. Whilst the hydrogen bonding patterns can be classified discretely (as reflected in secondary structure classification), the information contained in such patterns is implicitly transferred to structural conformation. Consequently, conformation space need not be considered discrete nor treated hierarchically, since its density already contains all relevant structural information. Rather, the prior discretisation of such a space would incur a loss of information.

The range of protein chain lengths, combined with the number of potential amino acid sequences (given the alphabet size), further complicated by external/environmental conditions, means that there are a huge number of potential conformations for protein chains to adopt. Due to favourability of certain conformations (local and/or global), conformation space is highly heterogeneous. As a consequence, the consideration of such a space is highly problematic.

Problems with Classification

The nature of protein chain conformation space has been previously considered, as is evident by the presence of structural classification schemes. Such schemes tend to describe conformation space discretely by identifying classes. Such classes are defined according to specific prior beliefs regarding the definition of similarity; such beliefs are either in accordance with the beliefs of a particular alignment/scoring tool (e.g. FSSP Holm et al., 1992), or subject to manual inspection (e.g. Murzin et al., 1995; Andreeva et al., 2008). The identification and recognition of such classes may be of great use in various practical applications. However, the concept of the dissimilarity between classes is often less meaningful; dissimilarity measures may be provided, but the derivation of such measures may mean that they are of limited use. Consequently, interpretation of such results should be carefully considered. It should be acknowledged that there is no unique solution to the problem, since there is no unique definitive distance between arbitrary structures. The consideration of new ways of perceiving conformation space, in terms of the density and in terms

of the distance between conformations, would be highly desirable (Kolodny et al., 2006). Indeed, the need for suitable metrics for describing protein fold space has been acknowledged (Sippl, 2008).

The Smoothness of Conformation Space

When considering the nature of protein conformation space, it is important to address the conceptual difficulties regarding the continuity of such a space. This problem is not trivial, and any general solution will be an abstraction. For example, consider the trivial case of the comparison of different conformations of the same structure – the number of atoms is equal, and a correspondence between atoms is known. In this case, a smooth transition in space between the two conformations exists, such that any conformation along this path could be represented by a set of points in Euclidean 3-space. However, if the sequences are different, then the fact that not all atoms correspond means that abstract conformations along such a path cannot be represented in Euclidean 3-space, meaning that conformation space is not smooth. This issue could be addressed by considering only the main chain atoms, which inherently always correspond. However, this would only solve the problem if the compared protein chains comprise the same number of residues. Therefore, due to the fact that protein chains exist with a wide variety of chain lengths, conformation space cannot be considered truly smooth.

The common solution to this problem is to compare substructures of the chains (i.e. a residue-residue alignment). This corresponds to the mapping of the two objects in $3N$ and $3M$-dimensional space, respectively, both into a space with at most $3\min(N, M)$ dimensions, where N and M are the number of considered atoms in the two chains. Such substructures may then be represented as directly comparable landmark configuration matrices. The corresponding conformational space of the substructures is indeed smooth, although information regarding unaligned atoms is discarded. Nevertheless, this form of solution is currently deemed the most suitable. Consequently, if this abstraction is accepted, then the considered conformational space can indeed be considered smooth.

Here, it is proposed that the dimensionality of the compared substructures should ideally be as high as possible, in order to produce a result that most closely conceptually represents the conformational change between the two chains. Specifically, the dimensionality of compared substructures should be equal to, and not less than, $3\min(N, M)$. It is for this reason that one of the primary criteria governing the alignment of chains in the developed software is that the alignment length is maximised (see §2.2). This ensures that the substructures used to generate any global score for the dissimilarity between protein chains have dimensionality as close as possible to the dimensionality of the shorter of the two chains.

Note that there may be other approaches. For example, the use of dimension reduction techniques could be employed in order to reduce dimensionality without losing the same information – this approach may result in interpretative difficulties, yet still achieve meaningful quantitative results in some situations. Such an approach is not explored in the present work, being mentioned

only to demonstrate that other possibilities may exist.

4.1.2 The Nature of Fragment Conformation Space

The developed software represents local structure using structural fragments (see §1.2.2 and §2.1). The use of these constructs allows a view of local conformation at a specific level of structural resolution. Consequently, the consideration of multiple fragment lengths could in principle allow a multi-resolution view of local conformation.

Unlike the protein chains we attempt to compare, structural fragments are specially designed so that any two n-residue fragments can be represented as directly comparable landmark configuration matrices. Consequently, any two n-residue fragments have equal dimensionality. Therefore, unlike protein conformation space, fragment conformation space can be considered to be smooth. Note however that not all possible conformations will exist in nature, due to chemical restraints. Nevertheless, any theoretical conformation (whether chemically plausible or not) can be represented, and thus a smooth path exists between the conformations of any two structural fragments. The existence of energetically favourable/unfavourable conformations is reflected by the extremely heterogenous nature of fragment fold space; this space comprises regions where density is very high (e.g. the α-helix attractor) and very low (chemically impossible configurations).

The Effect of Fragment Length on Scoring

Using the employed alignment method, chains are compared by considering the net agreement of optimally aligned fragment-pairs. The chosen fragment length n reflects the structural resolution of the comparison; using a small n considers the net optimal agreement of structural details, whilst using a large n is more concerned with the agreement of larger portions of the chain. Consequently, the qualitative information contained in any reported statistic (e.g. average fragment score, although this is not the only possible statistic) will depend on n. However, whilst such information is different for different n, the transition is smooth and interpretable.

For example, consider the case of domain movement, where intra-domain structural conservation is high. If n is low, then the average fragment score represents net local dissimilarity, thus will score relatively well. If n is extremely large (equal to chain length), then the score will represent global rigidity, thus will score relatively poorly. The behaviour of the score for intermediate fragment lengths will depend largely on the sizes of the domains.

A contrasting example would be the case where there is a high degree of topological and global conservation, but lower local conservation, such as structures that are poorly determined (e.g. low-resolution, high B-factors) or structures with the same fold but low sequence identity. In such cases, scores corresponding to low fragment lengths would score relatively worse, whilst scoring better for higher values of n.

In a given session, a specific fragment length may be chosen, depending on the desired level of

structural resolution for comparative discrimination. In such cases, appropriate interpretation is vital. However, in more detailed studies, the consideration of a variety of fragment lengths could provide useful and complementary information regarding conformational differences between the chains, given the optimal alignment (as an aside, note that the optimal alignment may vary with fragment length). Given a choice of statistic to quantify the global agreement, this approach would result in a profile vector of scores (covering some range of n), allowing the assessment of similarity at a range of structural resolutions.

An alternative strategy might be to combine the results achieved using various fragment lengths on a per residue basis, rather than considering differences in global statistics. This might involve selecting the best of all residue-based scores realised using various fragment lengths, resulting in a multi-resolution residue-based scoring system, producing a scalar score. However, the scores achieved using different fragment lengths would have to be comparable, i.e. either on the same scale, or selected using a probabilistic approach. Such multi-resolution residue-based scores might be subsequently combined into global statistics, which would provide complementary information to that contained in the profile vector suggested above.

The Effect of Fragment Length on Conformation Space

All fragment spaces are related. Let fragment conformation space of order n refer to the space of n-residue fragments. The structure of any element in the space of order n is a substructure of a theoretically infinite[1]number of elements in all higher-order spaces. Note that the density of a particular element in space order n is shared between those associated (infinite number of) conformations in space order $n+1$ of which it is a substructure. However, the density is not distributed equally among those infinite number of conformations. Rather, it is the range of possible conformations of naturally occurring protein chains that determines exactly how this density is distributed. Furthermore, the structures of $k+1$ fragments in the space of order $n-k$ are substructures of any element in the space of order n. Consequently, this apparent hierarchical nature of fragment spaces implies that there are relationships between these spaces, even if such relationships may be highly complex. For example, exactly two structures in the space of order $n-1$ are substructures of a single element in the space of order n. Although these two elements may be separated in space, they are related since they share one common substructure, which is an element of the space of order $n-2$.

Structural fragments have fewer degrees of freedom when n is smaller. Therefore, the consideration or characterisation of the conformation space of shorter fragments should be a simpler

[1]In nature, chemical restraints cause correlations in atomic positions. Consequently, an infinite number of fragments could not share a common substructure in practice. However, since we are interested only in the density of conformation space, we may relax this condition, requiring only that the substructures be sufficiently close in conformation space in order to consider them equal. Therefore, the theoretical case may be considered without loss of generality or practical applicability.

problem than that of longer fragments, or of whole protein chains. Tentatively, it may be reasonable to suspect that there are some features of such spaces that change in a reasonably smooth way as n increases (of course, it cannot be truly smooth, since n is discrete); it seems to make sense that there could be some qualitative similarities between fragment spaces, providing the order of these spaces (n) is consecutive, or at least similar. For example, a region of extremely high density corresponding to α-helices would be expected to exist in fragment conformation space, providing the order of that space is sufficiently low. However, qualitative changes will be expected, particularly when the density of a region of space order n is dispersed relatively more widely amongst elements of space order $n+1$ than for lower order spaces. For example, the α-helix attractor will become relatively less dense as n surpasses the length of naturally occurring α-helices. For higher values of n, the structural resolution of the fragments surpasses that of secondary structure, and begins to approach that of tertiary structure.

4.1.3 Application to Scoring

It is desirable to achieve a meaningful representation of fragment conformation space. Specifically, it would be useful to be able to view these high dimensional n-residue fold spaces in just a few dimensions, using just a few descriptors. Ideally, such a representation would allow the comparison of all these spaces in a common reference frame. In order to be meaningful, any such descriptors must be invariant to rigid body transformations.

Various views of protein fold space have been previously achieved. The general approach is to use a measure of dissimilarity (e.g. a global dissimilarity score resulting from an alignment program) and perform multidimensional scaling, producing a representation of fold space in few dimensions. Such works have produced meaningful results; space has been separated into regions that generally correspond to low-level classes in protein classification schemes (e.g. according to α-helix and β-sheet composition). Of course, any such results would depend highly on the dissimilarity measure used, and more generally on the philosophical approach towards the definition of similarity. Consequently, such representations are not unique, and many other complementary ways to describe fold space may yet exist. Importantly, any such representation should be interpreted appropriately given the source methodology.

Due to the comparative simplicity of fragment conformation space (in relation to protein fold space), there may be other ways to achieve meaningful information regarding the nature of such spaces. Specifically, if there are meaningful representations that do not explicitly require the use of information regarding the dissimilarity of elements (in this case, fragments), then it may be possible to infer knowledge that might be subsequently used in structural analyses.

In protein fold space, it is unclear whether it would be possible to infer useful information using just a few descriptors without assuming prior knowledge regarding some notion of the distance between structures. This is due to the inherent relative sparsity of the higher-dimensional spaces

that protein chains occupy. However, it is suggested that it may be possible to extract useful information about general trends regarding density in fragment conformation space using just a few descriptors.

Proposed Approach

Density is a very interesting property of fragment conformation space. Any heterogeneities in density will be a direct consequence of the nature of chemical bonds, and of protein folding. Fragments in a region of high density will commonly occur, and thus even non-homologous fragment-pairs would be expected to score relatively well (e.g. have a low Procrustes score) when compared with fragment-pairs in less dense regions. This behaviour may be readily observed by considering differences between helix and loop fragments. Two helix fragments from non-homologous structures would generally have a very low (well-scoring) Procrustes score. However, two random loops would be expected to score poorly. This behaviour causes severe artefacts in the alignment and scoring of structures; particularly in the method adopted by *ProSMART*. In general, an alignment is dominated by the favourability of helices, whilst the relative agreement of loops is often less influential. This is also reflected heavily in global scoring, whereby non-homologous structures that may be categorised as predominantly α-helical may score much better than homologous structures comprising mainly β-sheets. Whilst it might be argued that this behaviour provides information in itself, it would be very useful to achieve an alternative (complementary) strategy that reduces this inherent bias.

Here, it is proposed that the use of features that capture information regarding the density of fragment space could be used to standardise fragment dissimilarity scores. Such a standardisation would reduce bias to some degree, achieving a score that better reflects the significance of observed conformational dissimilarities. It is hoped that using such a scheme will result in less-commonly observed features (e.g. specific loops) dominating an alignment where appropriate, and commonly observed features (e.g. helices) being identified as random unless in near-identical conformations. Furthermore, it is hoped that the magnitude of achieved global scores might be more readily interpretable, and that, ambitiously, such scores might in future be used in the consideration of the (non-unique) distance between protein structures in conformation space. Due to being conformation-invariant, and achievable at multiple levels of structural resolution, such scores would be complementary to existing approaches.

In principle, there may be many descriptors appropriate for the standardisation of scores in this way. In accordance with the continuous and purely structure-based approach of the present work, here we consider some simple shape descriptors. One of the key features of this approach is that it is continuous. We do not discretise/classify fragment conformation space into specific regions (e.g. helices, strands, other commonly occurring fragments, etc.) nor explicitly identify any regions of high density. Rather, we are able to smoothly and implicitly describe continuous trends through

all of fragment conformation space using just a few descriptors. Since these descriptors are engineered, and not observed (as with multidimensional scaling), sensible and succinct interpretation of dimensions is possible. Furthermore, the achieved representation/view of fragment fold space is independent of any protein chain alignment, thus is not subject to bias from external methodology.

4.2 Trends in Fragment Conformation Space

Whilst there are many shape descriptors that could be used to obtain a useful representation of fold space, here we focus solely on the eigenvalues of the fragment covariance matrices. Such other shape descriptors might include the average centroid size and higher spherically-invariant moments, or those in Fourier space.

The general idea is to infer information regarding the Procrustes score distribution, given properties of the fragments' shapes. Note that fragment-pairs with different overall shapes (e.g. helices versus strands) will inherently have a poor Procrustes score. However, predicting the score between fragment-pairs with similar overall shapes, without having knowledge of their actual structures, is less obvious. We might infer that the approximate 'best case scenarios', i.e. the lowest Procrustes scores, can only occur if the shapes of the fragments are near-identical. Consequently, given that we are interested in identifying the significance of observed similarity, and are not interested in the significance of dissimilarity, we focus on the case where the shapes of the compared fragments are (randomly) near-identical. This means that, when considering the density of fragment space, we aim to only look at fragment-pairs that are very close in space, being not interested in the comparison of those that are greatly separated. However, importantly, this is achieved without any explicit reference to the (Procrustes) distance between fragments.

4.2.1 Choice of Descriptors

Remembering that we ultimately aim to predict the Procrustes score distribution, given the values of the descriptors, consider the functional form of the Procrustes score:

$$d(\mathbf{F_1}, \mathbf{F_2}) = \sqrt{\frac{\mathrm{tr}\left(\hat{\mathbf{F}}_\mathbf{1}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{1}\right) + \mathrm{tr}\left(\hat{\mathbf{F}}_\mathbf{2}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{2}\right) - 2\mathrm{tr}\left(\mathbf{S}\right)}{4n}} \tag{4.1}$$

Now consider the case where the compared fragments are precisely equivalent, i.e. $\mathbf{F} = \mathbf{F_1} = \mathbf{F_2}$. In this case, the Procrustes score reduces to:

$$d(\mathbf{F}, \mathbf{F}) = \sqrt{\frac{1}{2n}\mathrm{tr}\left(\hat{\mathbf{F}}^\mathbf{T}\hat{\mathbf{F}}\right) - 2\sum_{i=1}^{3}\lambda_i} = 0 \tag{4.2}$$

where λ_i are the eigenvalues of the covariance matrices $\frac{1}{4n}\hat{\mathbf{F}}^\mathbf{T}\hat{\mathbf{F}}$. Note that the singular values and eigenvalues of $\frac{1}{4n}\hat{\mathbf{F}}_\mathbf{2}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{1}$ are equivalent when $\mathbf{F_1} = \mathbf{F_2}$, due to $\frac{1}{4n}\hat{\mathbf{F}}_\mathbf{2}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{1}$ being symmetric. Given their equivalence, it is more convenient to use eigenvalues rather than singular values, for purposes of computational speed (at least in the present implementation; see Figure 87). One major practical

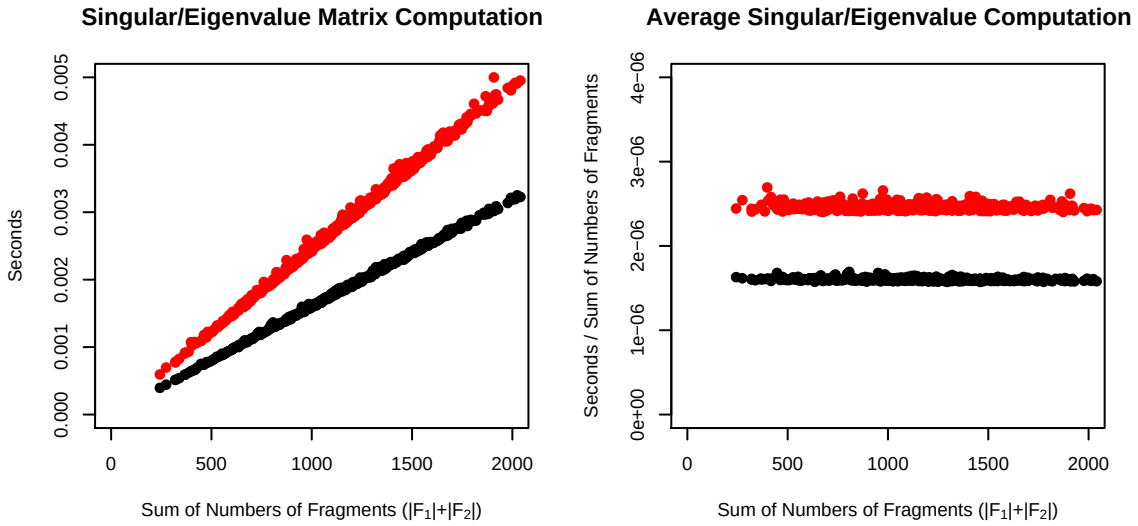


Figure 87: Computation time of calculation of the singular values (or eigenvalues) of the covariance matrices for each fragment in the two chains (a total of $|F_1| + |F_2|$ calculations per observation), according to the employed implementation. Specifically, we use the default implementation of the C++ JAMA package (Pozo, 2003; Hicklin et al., 2000) for calculation of eigenvalues and the singular value decomposition; further optimisation of these algorithms may produce varying results. A description of the 30-chain dataset used is provided in Chapter 3. Black points correspond to calculation of the eigenvalues; red points to the singular values.

difference between the diagonal elements of $\frac{1}{4n}\hat{\mathbf{F}}^{\mathbf{T}}\hat{\mathbf{F}}$ and the eigenvalues is that the latter are invariant to rotation of the original coordinate frame; one of our requirements for being a suitable descriptor.

It is possible to rewrite the Procrustes score in terms of the eigenvalues:

$$d(\mathbf{F_1}, \mathbf{F_2}) = \sqrt{2\sum_{i=1}^{3}\bar{\lambda}_i - \frac{1}{2n}\mathrm{tr}\,(\mathbf{S})} \tag{4.3}$$

where $\bar{\lambda}_i = \frac{1}{2}(\lambda_{1i} + \lambda_{2i})$ is the average of the i^{th} eigenvalues. Consequently, we have identified a completely deterministic component of the Procrustes score ($\sum\bar{\lambda}$), and a component which is dependent on the details regarding the dissimilarity of structures ($\frac{1}{4n}\mathrm{tr}(\mathbf{S})$). Since theoretical conformation changes are smooth, we also know that $\frac{1}{4n}\mathrm{tr}(\mathbf{S})$ tends to $\sum\bar{\lambda}$ as the structures become identical.

We must acknowledge some very important implications:

$$\sum_{i=1}^{3}\bar{\lambda}_i = \frac{1}{4n}\mathrm{tr}\,(\mathbf{S}) \iff d(\mathbf{F_1}, \mathbf{F_2}) = 0 \tag{4.4}$$

$$\sum_{i=1}^{3}\bar{\lambda}_i = \frac{1}{4n}\mathrm{tr}\,(\mathbf{S}) \implies \lambda_{11} = \lambda_{21}, \lambda_{12} = \lambda_{22}, \lambda_{13} = \lambda_{23} \tag{4.5}$$

$$\sum_{i=1}^{3}\bar{\lambda}_i = \frac{1}{4n}\mathrm{tr}\,(\mathbf{S}) \not\Longleftarrow \lambda_{11} = \lambda_{21}, \lambda_{12} = \lambda_{22}, \lambda_{13} = \lambda_{23} \tag{4.6}$$

The third of these is particularly vital to the proposed approach. In summary, a zero Procrustes score implies fragments to be identical and for the eigenvalues of their covariance matrices to be equal. However, simply having identical eigenvalues does not imply structures to be identical, and thus does not imply the Procrustes score to be zero.

These properties suggest the average eigenvalues $\bar{\lambda}_i$ to be ideal candidates for use as descriptors of fragment conformation space (for the purpose of investigating Procrustes score trends). This is due to them being rotation-invariant, due to not requiring any explicit alignment/comparison of the fragment-pair in order to achieve these descriptors, and due to their intrinsic relatedness to calculation of the Procrustes score.

Note that it would not be possible to use the average eigenvalues to infer useful information regarding the traditional scale-invariant Procrustes distance; the fact that our implementation of the Procrustes score is only invariant to a rigid body transformation is vital for the proposed approach. It is interesting that invariance to such scaling is desired in other applications, whilst in our application it is the preservation of such scale information that is exploited in order to obtain a view of conformation space.

For brevity, from here onwards we will refer to the eigenvalues λ_{ki} of a normalised fragment covariance matrix $\frac{1}{4n}\hat{\mathbf{F}}_\mathbf{k}^\mathbf{T}\hat{\mathbf{F}}_\mathbf{k}$ as the 'fragment eigenvalues', and similarly for the average fragment eigenvalues $\bar{\lambda}_i$. Also, we assume implicit ordering of the eigenvalues ($\lambda_{k1} \geq \lambda_{k2} \geq \lambda_{k3}$), and refer to the largest of these (λ_{k1}) as the 'principle eigenvalue'.

4.2.2 Eigenvalue-Based Fragment-Pair Filtering

We wish to know the distribution of Procrustes scores that may arise by random, given that they have the same overall shape properties. Furthermore, we hypothesise that, due to the heterogeneous nature of the density of fragment conformation space, such a distribution will depend on the particular values of these overall shape properties (average eigenvalues of covariance matrices). Consequently, we aim to estimate the distribution of Procrustes scores of fragment-pairs that appear to be arbitrarily close in space, given knowledge only of their average eigenvalues.

We propose that it is appropriate to consider only fragment-pairs that have arbitrarily close eigenvalues, since those which have different eigenvalues will naturally have a poor Procrustes score; the aim is that any achieved standardisation should still result in such fragment-pairs having a relatively poor score. This is an important point – if we were to include fragment-pairs with very different eigenvalues in the analysis, then the resultant standardisation would be inclined towards scale-invariance, which is highly undesirable. For example, the alignment of helices with helices should score as favourably as the alignment of strands with strands, although the alignment of helices with strands should still score very unfavourably due to their general shape differences.

Figure 88 shows the typical relationship between Procrustes score and average (and difference between) fragment eigenvalues, for 9-residue fragments. Results are shown for the intra-chain

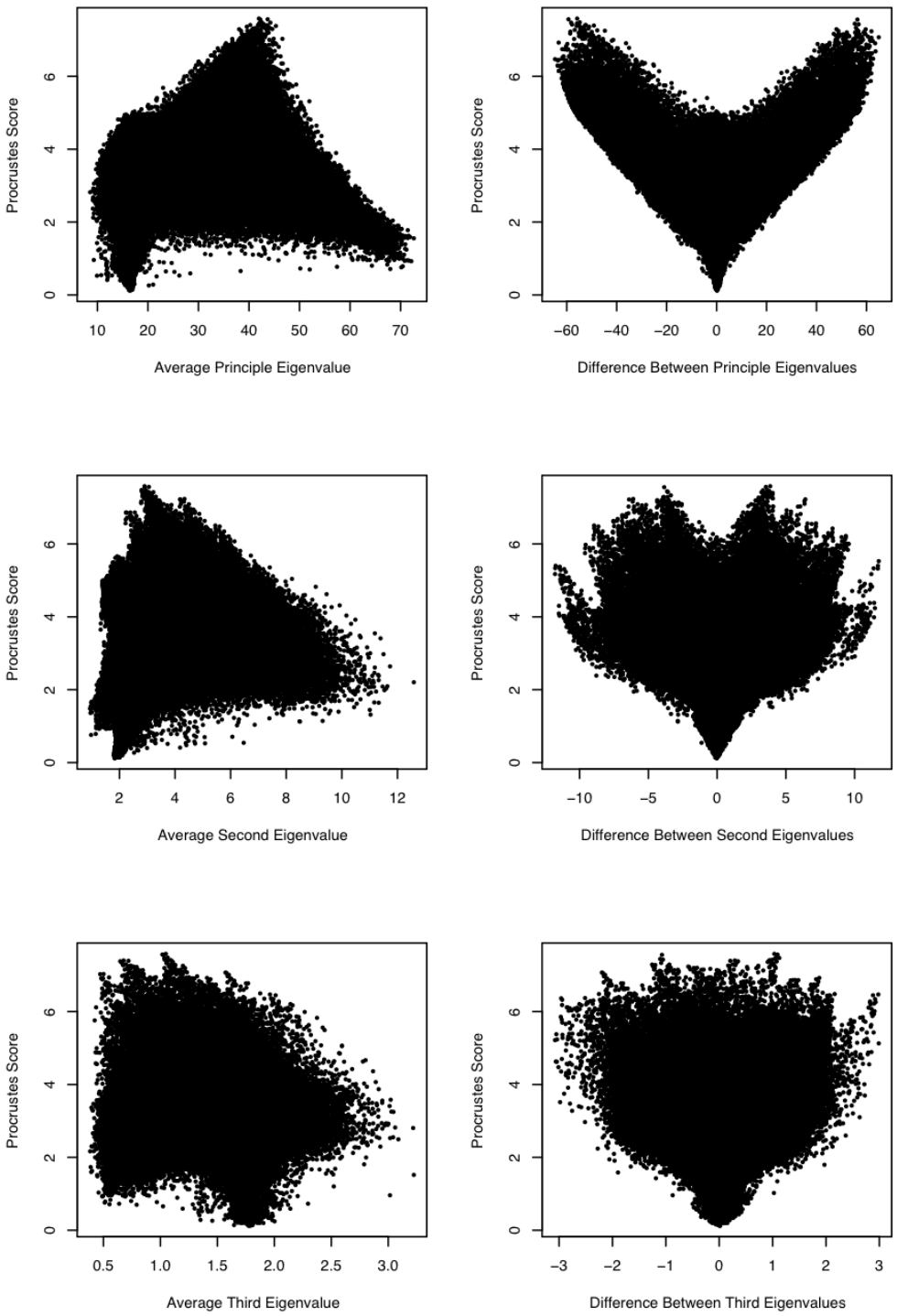


Figure 88: Procrustes score versus average eigenvalue (left) and difference between eigenvalues (right). The top two graphs correspond to the principle, the middle two to the second, and the bottom two to the third eigenvalues. All graphs relate to the triangular all-on-all comparison of 9-residue fragment-pairs from a 0.95Å structure (PDB code: 1n1p) comprising 498 residues; 29% helices and 23% strands.

comparison of fragments from a high-resolution structure. Similar results may be observed for other structures, although with different emphases on different regions, according to their fragment composition (the structure used in this example had a reasonable amount of helices, strands and loops, so as to show representative trends). The relationship for average principle eigenvalue (top left graph) is of great interest, being the descriptor that accounts for the largest amount of variability in fragment shape. Helices occupy the region on the left with low Procrustes scores, whilst strands occupy the far right. There is a curious region in the middle of the graph in which the maximum Procrustes score extends to over 7Å. This behaviour can be explained by considering the difference between principle eigenvalues (top right graph), which exhibits a 'V' shaped trend. The 'wings' in this graph correspond to fragment-pairs that have very different shapes, e.g. the comparison of a strand and a helix. There appears to be a lower bound to the graph, indicating that there is a lower threshold on the Procrustes score, given the difference between principle eigenvalues. This supports the assertion that it is reasonable to consider only fragment-pairs that have very close eigenvalues – those whose eigenvalues are too different will score poorly, and thus there is little danger of them being identified as significantly similar (after some standardisation). Note also that the Procrustes score upper bound is much lower in the centre of the graph than on either wing – this indicates that the wings must be removed in order to achieve a meaningful estimate of the distribution of the Procrustes score. Otherwise, the estimate would be biased towards higher Procrustes scores due to the influence of fragment-pairs with dissimilar eigenvalues, and thus the achieved estimated distribution would not be optimally useful and applicable for similar structures, which would have similar eigenvalues. Similar, although somewhat convoluted, trends are observed for the second and third eigenvalues.

Ideally, the Procrustes score distributions for random fragment-pairs should be realised when the eigenvalues of the compared fragments are exactly equal. However, since we take an empirical approach, this is not possible. Instead, we consider only fragment-pairs that have arbitrarily close eigenvalues. Consequently, it is now necessary to filter the fragment-pairs so that only those which have sufficiently similar first, second and third eigenvalues are considered. Specifically, we require the absolute difference between the eigenvalues to be small relative to the average value:

$$|\lambda_{1i} - \lambda_{2i}| < \alpha\bar{\lambda}_i \qquad \text{for } i = 1, 2, 3. \tag{4.7}$$

The parameter α determines the allowed relative dissimilarity between eigenvalues in order for the fragment-pair to be used in subsequent analysis. Standardising with respect to the average eigenvalue effectively places the three eigenvalues on the same scale, thus allowing α to be fixed for all eigenvalues, whilst allowing extra flexibility for fragments with larger eigenvalues. Note that this test may be performed without requiring knowledge of the sample, which is deemed powerful. Other methods of standardisation may have also been reasonable, although sample-based methods would have led to some ambiguity, and been more computationally expensive. The ratio between $|\lambda_{1i} - \lambda_{2i}|$ and $\bar{\lambda}_i$ is shown in Figure 89; for illustration, lines are shown representing the filtering

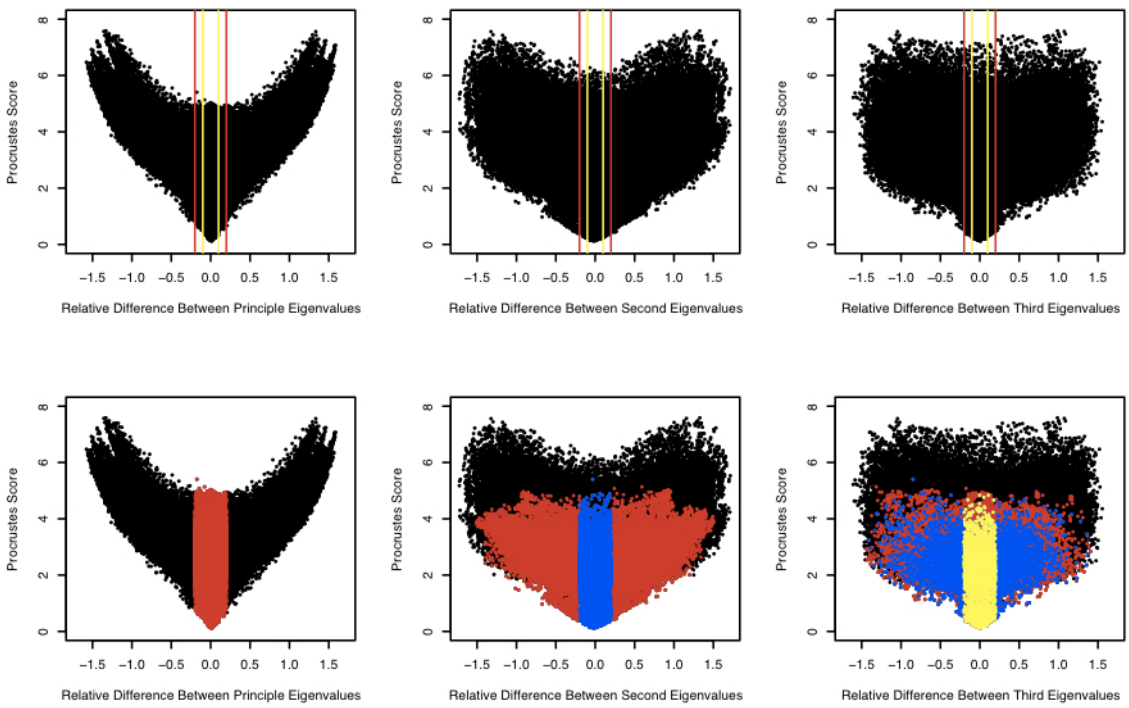


Figure 89: Procrustes score versus ratio between difference and average eigenvalues, for the same data presented in Figure 88. The left graphs corresponds to the principle, the centre to the second, and the right to the third eigenvalues. Top: lines are shown corresponding to $\bar{\lambda}_i^{-1}|\lambda_{1i} - \lambda_{2i}| = 0.1$ (yellow) and $\bar{\lambda}_i^{-1}|\lambda_{1i} - \lambda_{2i}| = 0.2$ (red). Bottom: data are shown before filtering (black), after principle eigenvalue filtering (red), after principle and second eigenvalue filtering (blue), and after all eigenvalue filtering (yellow), using a threshold of $\alpha = 0.2$.

threshold for $\alpha = 0.1$ and $\alpha = 0.2$.

In practice, the value of α should be chosen to be as small as possible so that results approximate fragment-pairs with equal eigenvalues, whilst being large enough to be left with a workable amount of data after filtering. Due to only considering fragments from one structure, in Figure 89 a threshold of $\alpha = 0.2$ is used to illustrate the filtering process. Filtering by the principle eigenvalue removes the 'wings', thus removing the highest Procrustes scores (e.g. removes helix-strand fragment-pairs). Principle eigenvalue filtering still leaves a wide range of differences between second eigenvalues, suggesting the necessity for the further filtering. Similarly, after second eigenvalue filtering there is still a wide range of third eigenvalue differences, suggesting that filtering should be performed on all three eigenvalues.

At this point, it is appropriate to acknowledge that, after eigenvalue filtering, we are able to discern some information regarding the regions that certain fragment types occupy in eigenvalue space. This is illustrated in the Procrustes score versus eigenvalue graphs displayed in Figure 90. Specifically, we are able to see that there are dense regions which correspond to particularly favourable conformations, or at least favourable eigenvalues, for the particular protein structure

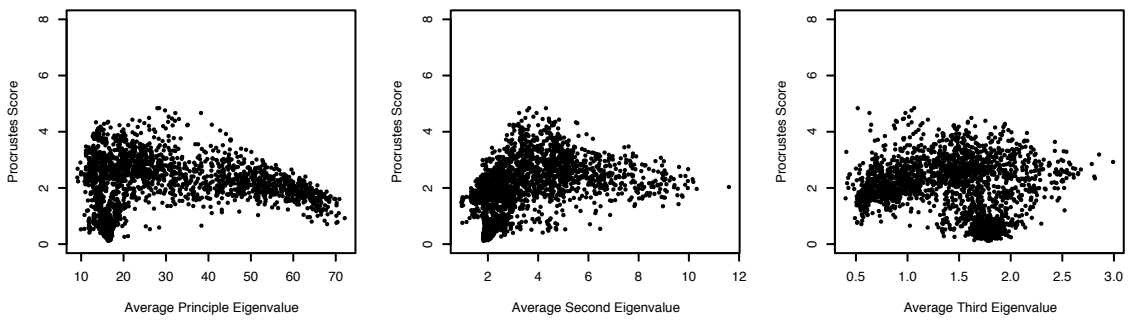


Figure 90: Procrustes score versus average eigenvalues, for the data presented in Figures 88 and 89, after eigenvalue filtering with $\alpha = 0.2$. The left graphs corresponds to the principle, the centre to the second, and the right to the third eigenvalues.

used in the example. Observed trends agree with intuition. Broadly speaking, α-helix fragments have low Procrustes scores, low principle and second eigenvalues, and middling third eigenvalues. In contrast, β-strand fragments have middling Procrustes scores, low second eigenvalues, and low third eigenvalues. Other fragments (loops) vary greatly in eigenvalue, but have a tendency towards higher Procrustes scores. Note that more globular loops will exhibit higher second eigenvalues. Importantly, some undesirable features, such as the 'wings' previously observed in the graph for the principle eigenvalue, have been removed by the filtering process, allowing trends to be more readily exploited.

By applying eigenvalue filtering to fragment-pairs, we have obtained a useful and meaningful view of fragment conformation space. Any region of high density in this representation may correspond to a region of high density in fragment conformation space. However, it could correspond to an artefact of the eigenvalue-based approach – it is not possible to distinguish between different regions of space that contain fragments having equal eigenvalues. Consequently, any observed density corresponding to eigenvalues $(\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3)$ in this representation is the net density over all regions of fragment conformation space that correspond to these specific eigenvalues. For example, by looking at the graphs of eigenvalue-filtered fragment-pairs shown in Figure 90 we can see that there are some fragment-pairs with similar eigenvalues that have poor Procrustes dissimilarity scores in the region of 4–5Å (these correspond to loops). This phenomenon is vital to the proposed approach. Specifically, this artefact allows us to use this information to infer knowledge regarding the significance of observed Procrustes scores, given fragments' average eigenvalues.

4.2.3 Non-Redundant Dataset

In order to achieve an appropriate view of fragment conformation space, it is desirable for the fragments used in the analysis to adequately represent the wide range of potential fragment conformations that may occur in nature. However, it is also important that any given fragment is not

disproportionately over-represented; it is intended that fragments should not align particularly well, since they will be used to represent the distribution of scores for random fragments. Consequently, it is desirable to aim to achieve a database of fragments that are unrelated by homology, so that any observed dense regions in conformation space (low Procrustes scores) should be due to natural favourability and not due to significant bias in the database.

The list of protein chains used to generate the database of random fragments was compiled using *PDBselect* (Griep and Hobohm, 2010; Hobohm and Sander, 1994; Hobohm et al., 1992), a tool for the creation of customised representative (non-redundant) datasets. This tool allows the identification of a list of protein chains such that no two chains have greater than a specified level of sequence similarity. The identified structures are such that their quality is deemed better than other structures sharing sequence homology. Specifically, quality is assessed using a function of crystallographic resolution and R-factor. This approach is ideal for our purpose, since the resulting list of structures should in principle be non-redundant according to sequence homology. Furthermore, we should have a high degree of confidence that the obtained structural fragments should be realistic/consistent with their natural conformations, given that they are assumedly well-refined, at least in relation to other homologous structures.

The list of protein chains inspected by *PDBselect* when creating the non-redundant dataset was specified by the *PDBFINDER* database (Hooft et al., 1996) ('PDBFIND.TXT' was obtained from http://swift.cmbi.ru.nl/gv/pdbfinder/ on 12/4/2011). The criteria for inclusion was as follows: experimental method must be X-ray crystallography; chain must contain at least 25 residues; resolution must be no lower than 3.5Å; R-factor must be no worse than 0.35.

The sequence similarity threshold for determining maximum sequence homology was set to 10%. This contrasts with the default non-redundant database, which uses a 25% threshold. However, it was acknowledged that some structurally homologous chains can have surprisingly low sequence homology. The final non-redundant list of structures with no greater than 10% sequence homology

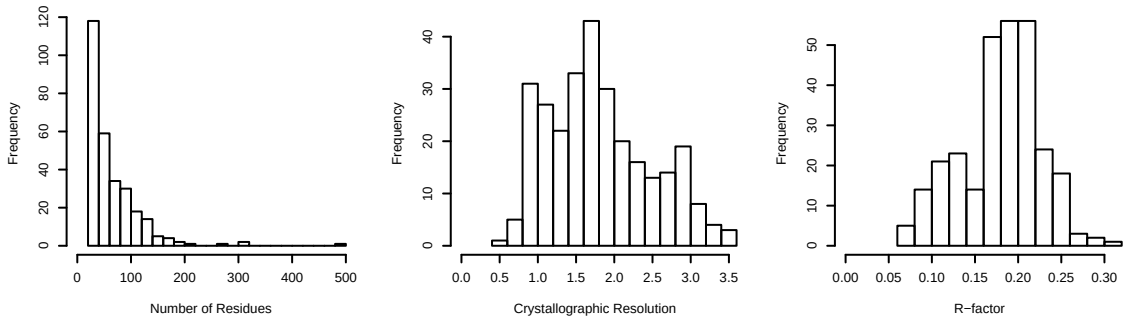


Figure 91: Histograms of the number of residues per chain (left), crystal resolution (centre), and R-factor (right), for the 289 chains selected for inclusion in the non-redundant dataset, with 10% sequence homology threshold, by *PDBselect* (see text).

returned by *PDBselect* comprised 289 chains, a total of 18,923 residues. Some properties of this dataset are displayed as histograms in Figure 91. A noteworthy observation is that the chains in the database have a distinct tendency to comprise relatively few residues (average 65 residues); this is expected due to the selection criteria employed by *PDBselect* (i.e. selecting structures with optimal resolution and refinement statistics). Consequently, it must be acknowledged that any results obtained in our subsequent analysis are based on this underlying potential bias; we cannot at this stage claim fragment conformation space to be similar for chains of all lengths. This issue should be investigated in future by determining whether the results of this analysis are dependent on the range of chain lengths in the employed non-redundant dataset.

4.2.4 A View of Fragment Conformation Space

Now that we have a non-redundant dataset, chosen candidate descriptors (average eigenvalues), and identified a method of data processing (eigenvalue-based filtering), it is possible to obtain a meaningful view of fragment conformation space, with the intention of searching for a useful representation.

The chains in the non-redundant dataset comprise 16,732 fragments (with fragment length $n = 9$). These fragments are pooled, and any information regarding their source is discarded. Of the 139,963,180 potential fragment-pairs, 3,959,271 were left after fragment-pair filtering (with $\alpha = 0.1$). Corresponding relationships between the Procrustes scores and the average eigenvalues are shown in Figure 92. There is a clear correlation between principle eigenvalue and Procrustes score for larger values of $\bar{\lambda}_1$. Such information could be exploited in order to gain knowledge about Procrustes score distribution. However, this is ruined by the behaviour for lower values of $\bar{\lambda}_1$. Specifically, the principle eigenvalue is not able to distinguish between helices, which have low $\bar{\lambda}_1$ and low Procrustes score, and loops that have low $\bar{\lambda}_1$ but high Procrustes score. Similarly deficient behaviour is observed for the second and third eigenvalues. We may conclude that it is not possible

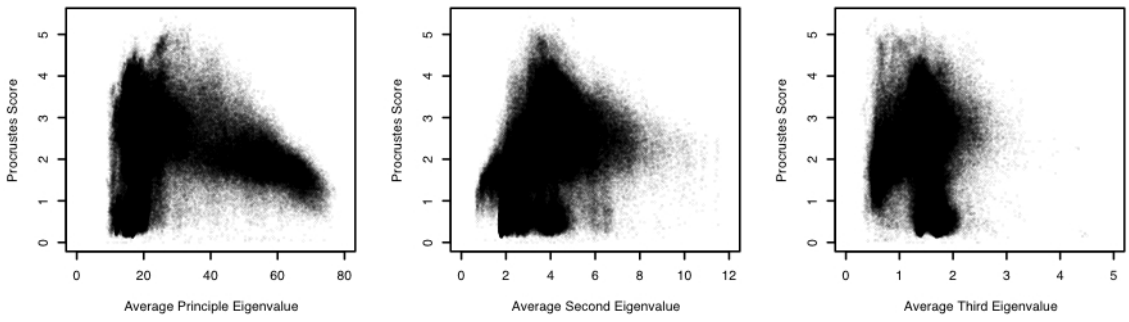


Figure 92: Procrustes score versus average eigenvalues for fragments in the non-redundant dataset, after eigenvalue filtering with $\alpha = 0.1$. The left graphs corresponds to the principle, the centre to the second, and the right to the third eigenvalues.

to obtain practically useful information by considering any one of the three descriptors alone; one such descriptor is not enough to adequately describe the correlations in conformation space.

We now consider whether it is possible to adequately model Procrustes score using two eigenvalues, in particular the principle and second eigenvalues. In order to do this, we consider the average Procrustes score as a function of these two descriptors. This required discretisation of $(\bar{\lambda}_1, \bar{\lambda}_2)$ space. A 100×100 grid was used, and mean and standard deviation of Procrustes scores were calculated for each cell. In order to ensure that estimates of mean and standard deviation were reasonably robust, cells were discounted if they contained less than 10 observations (these parameters were chosen arbitrarily).

Figure 93 illustrates average and standard deviation of the Procrustes score and the number of fragment-pairs in a cell, as a function of principle and second eigenvalues, using the 100×100 grid to partition space. The regions corresponding to helices and strands are identifiable as regions of relatively high density. Both of these regions have relatively low average and standard deviation of Procrustes scores. It is interesting that there appears to be a certain degree of correlation between average and standard deviation of Procrustes scores; this behaviour suggests potential for these two eigenvalues being suitable for predicting the distribution of Procrustes scores.

Figure 94 displays the relationship between average Procrustes score and the first two eigenvalues, uncovering some incredible behaviour. Strong trends in average Procrustes score are observed, suggesting that relevant information regarding the Procrustes score distribution can indeed be inferred directly from these two eigenvalues. The observed relationship is surprisingly smooth, given the heterogeneous nature of fragment conformation space. Average Procrustes scores lie approximately on a surface. It would seem logical that the nature of this surface is specific to the nature of

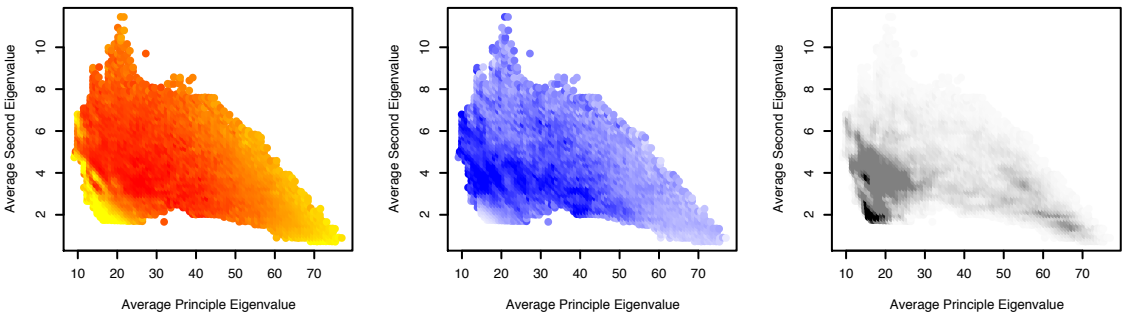


Figure 93: Properties of the fragment-pairs from the non-redundant database, after eigenvalue filtering with $\alpha = 0.1$. Specifically, average Procrustes score (left), standard deviation (centre) and density (right) are shown against the principle and second eigenvalues. Left: yellow corresponds to low Procrustes scores, red to high scores. Centre: light blue corresponds to low standard deviation, darker blue to higher standard deviation. Right: pale grey corresponds to low density of fragment-pairs, solid grey to high density, and black to extremely high density.

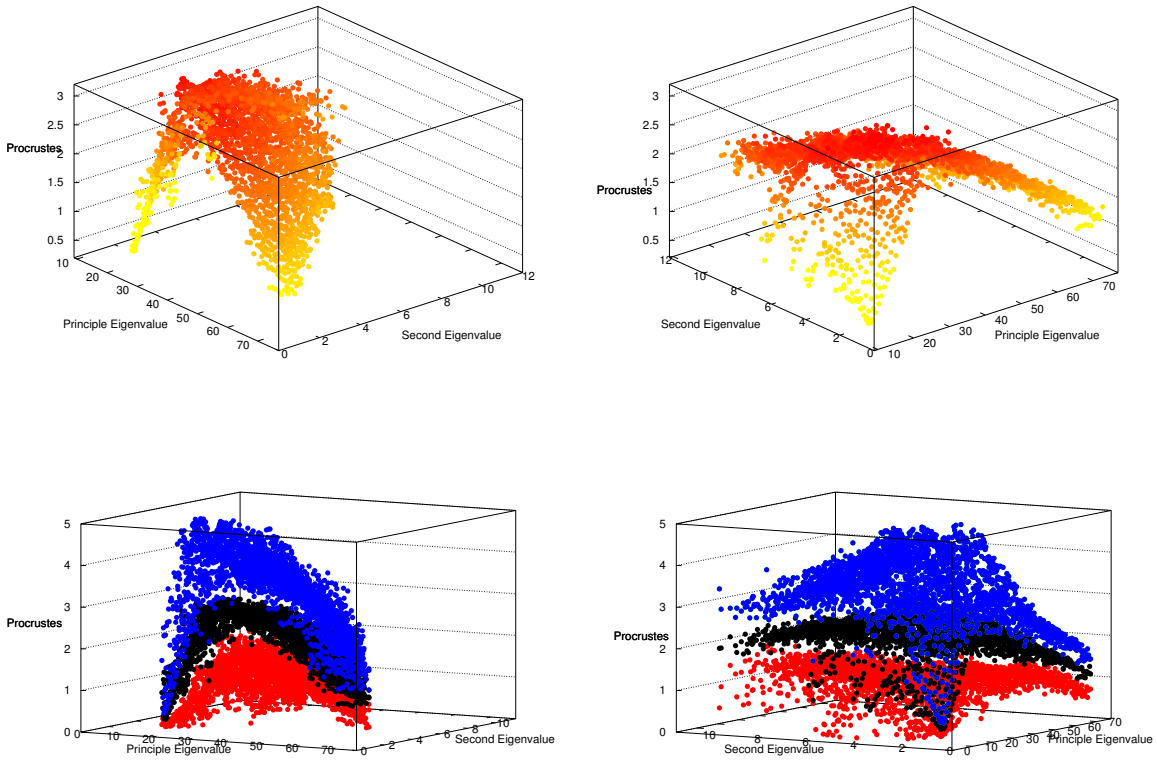


Figure 94: Various views of the relationship between average Procrustes score and principle and second eigenvalues for fragments in the non-redundant dataset, after eigenvalue filtering with $\alpha = 0.1$. In the upper graphs, the red-yellow colour gradient corresponds to the magnitude of the average Procrustes score, yellow indicating a low average Procrustes score. In the lower graphs, black points represent the average Procrustes score, and blue/red points represent two standard deviations from the mean of the observed Procrustes scores.

n-residue fragment conformation space, being fundamentally restricted by the correlation of atomic positions; fragments must adopt chemically sensible conformations. Importantly, note that the intuitively distinct regions of conformation space now appear suitably separated. For example, when considering the principle eigenvalue alone (as in Figure 92), it was impossible to distinguish between helices and loops with low principle eigenvalues. In contrast, when utilising the second eigenvalue, we see that these two types of fragments are separated. Specifically, fragments with low principle and second eigenvalues accumulate near an α-helix cusp point where both average and standard deviation of Procrustes scores are very low; the distribution of fragments with low principle but high second eigenvalues exhibits noticeably higher average value and variability.

Interpretation of the Achieved Representation

The observed surface is intrinsic to our view of conformation space. Specifically, if the average Procrustes score corresponding to $(\bar{\lambda}_1, \bar{\lambda}_2)$ is high, then that tells us that random fragments with eigenvalues $(\bar{\lambda}_1, \bar{\lambda}_2)$ have a tendency to be located very far apart in conformation space (providing distance between fragments is defined as Procrustes score). In contrast, if the average Procrustes score is low, then that tells us that fragments with eigenvalues $(\bar{\lambda}_1, \bar{\lambda}_2)$ have a distinct tendency to accumulate around a specific region of conformation space. This is exactly why the surface predicts lower Procrustes scores for helices, and to a lesser degree strands, than for loops. This is because helices and strands are extreme conformations – helices are extremely compact, and strands are extremely elongated. It intuitively makes sense that there cannot exist many fragments well-separated in space with such extreme principle eigenvalues. Consequently, in these two cases, the observed Procrustes score distributions are mainly due to natural conformational flexibility of fragments, rather than to the comparison of fragments in completely different conformations. For example, this is why random strands tend to score better than random loops, given equal eigenvalues of compared fragments.

4.2.5 Interpretation of Results and Limitations of the Approach

Correct interpretation of the proposed view of fragment conformation space is vital. Specifically, it is important to acknowledge that the distribution of the Procrustes score, given $(\bar{\lambda}_1, \bar{\lambda}_2)$, consists of two main components:

1. The first component consists of fragment-pairs that (randomly) adopt a similar conformation due to energetic favourability. If, in a given $(\bar{\lambda}_1, \bar{\lambda}_2)$ cell, there exists such a favourable conformation, then a distinct cluster of low Procrustes scores would be expected. Magnitude and variability of Procrustes scores would be a direct consequence of the density of conformation space. If multiple favourable conformations exist in the cell, then the distributions would be convoluted.

2. The second component comprises fragment-pairs that adopt completely different conformations, but happen to have approximately equal values of $(\bar{\lambda}_1, \bar{\lambda}_2)$. These fragment-pairs will have comparatively high Procrustes scores. If one of the fragments belongs to a cluster (favourable conformation) then there will in turn be a cluster of high Procrustes scores. If both fragments belong to clusters, then the cluster of high Procrustes scores will be very dense.

The important consequence of this is that the distribution of Procrustes scores in a given $(\bar{\lambda}_1, \bar{\lambda}_2)$ cell cannot be considered to be Normally distributed. Rather, the observed distributions may be multimodal, and exhibit significant skew and (positive or negative) excess kurtosis. Consequently,

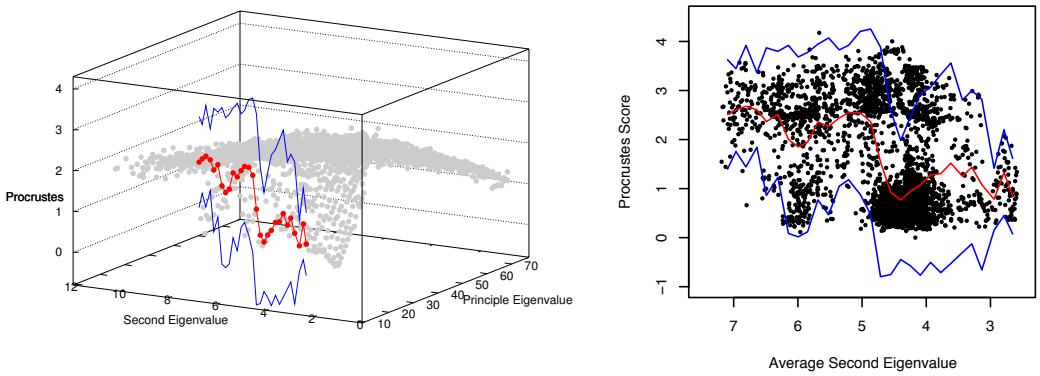


Figure 95: Right: distribution of Procrustes scores against average second eigenvalue, for fragment-pairs with average principle eigenvalues in the range $[12, 12.5]$. The red line represents average Procrustes score in the cell, and the blue lines represent two standard deviations from the mean. For visualisation, these lines are also shown in the graph on the left, which displays average Procrustes score (grey points) against principle and second eigenvalues.

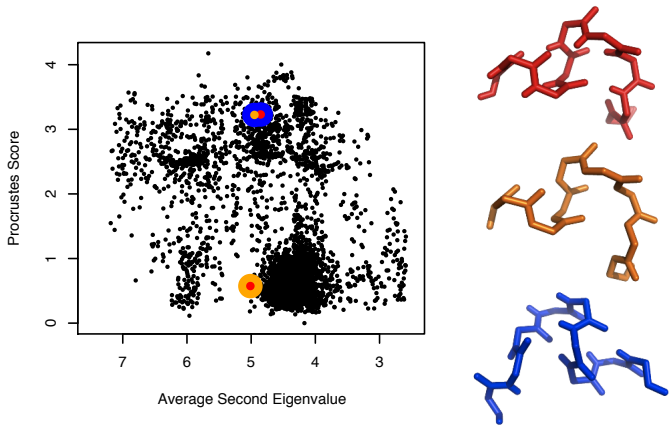


Figure 96: Graph showing the same distribution of Procrustes scores as in Figure 95. Three fragments are identified: residues 10–18 from the structure with PDB code 2vpl chain A; residues 288–296 from 1n1p chain A; and residues 604–612 from 1oai chain A. These fragments are represented by colours red, orange and blue, respectively, and are depicted on the right. The first two are located at helix C-termini, the third at a helix N-terminus. The Procrustes scores corresponding each of the three possible fragment pairings are identified by large coloured points in the graph on the left. Specifically, the score corresponding to the comparison of the fragments from 2vpl and 1n1p is shown as a red dot in an orange dot; that from 2vpl and 1oai as a red dot in a blue dot, and that from 1n1p and 1oai as a orange dot in a blue dot.

standardised scores based on these distributions cannot be reliably converted directly into a probability.

For example, Figure 95 illustrates a noisy principle eigenvalue cross-section, displaying Procrustes score against average second eigenvalue. Whilst the distribution in some cells is well-behaved, a large amount of non-Normal multimodal behaviour is evident. One large cluster can be seen, corresponding to $\bar{\lambda}_2 \approx 4.5$, for which the loosely interpreted 'confidence interval' represented by blue lines does not agree well with observed data; this is an artefact of the highly non-Normal behaviour. This cluster mainly comprises fragment-pairs located at helix termini. The frequent occurrence of such conformations results in clusters, with a particularly dense cluster having a low Procrustes score. Figure 96 shows the Procrustes scores arising from the comparison of three fragments; two found at helix C-termini, the third at a helix N-terminus. Consequently, they all have very similar eigenvalues, being found in the same cell corresponding to $(\bar{\lambda}_1, \bar{\lambda}_2) \approx (12, 5)$. In this case, whilst the two fragments from helix C-termini have a low Procrustes score, the cross-comparisons of the two different conformations result in (very similar) high scores.

We conclude that standardised scores based on the distributions of scores in $(\bar{\lambda}_1, \bar{\lambda}_2)$ cells cannot reliably directly imply a probability. This is an artefact of the method. The consideration of all Procrustes scores from any potential random fragment-pair, given approximately equal eigenvalues, is required. We are not trying to estimate the density of conformation space, nor the significance of a fragment-pair's score relative to its conformational neighbours. Rather, we intend to use the observed general trends in order to reduce the overall weight of helices (and to a lesser degree strands) relative to loops, in a reasonably smooth way. This is achieved by standardising with respect to the overall observed score magnitude and variability, in a way that is consistent with trends in the non-redundant database, rather than being based on arbitrary weight parameters or functional forms.

Potential Modifications to the Approach

If it becomes desirable to estimate the distribution of Procrustes scores arising from conformational variability alone (i.e. not including scores between different conformations), then a different approach must be taken. A heuristic approach could be used to filter some of the fragment-pairs that adopt completely different conformations. For example, for each cell, multimodality could be handled by considering the distribution of the cluster with the lowest Procrustes scores, discarding all observations with Procrustes scores significantly higher than the cluster. Another approach might be to compare the Procrustes scores arising from fragment-pairs with and without sequence direction of one of the fragments being reversed. For example, comparing two fragments from helix C-termini with a fragment from a helix N-terminus (specifically, the example in Figure 96) yielded scores 3.23 and 3.22; inverting the latter fragment's sequence yielded scores 1.69 and 1.59. Such cases indicate that the fragments likely do not adopt the same conformation. A similar fil-

tering technique could result from consideration of the Procrustes scores arising from comparing fragments after a simple reflection; a rotation should score higher than a rotoreflection for similar conformations.

4.3 Standardisation of the Procrustes Score

4.3.1 Procrustes Score Distribution Smoothing

Given that the mean and standard deviation of the Procrustes score distributions have been calculated, as described in §4.2.4, a protocol should be established for determining the score distribution statistics for a newly observed fragment-pair. Since the discretisation of $(\bar{\lambda}_1, \bar{\lambda}_2)$ space into a 100×100 grid is arbitrary, and a reasonably smooth transition is desired, the statistics are smoothed by considering neighbouring cells. This should help to make the trends more continuous, reducing any noise on the surface (for both mean and standard deviation).

Since it is theoretically possible for observed fragment-pairs to have average eigenvalues outside the range of the non-redundant dataset, the approach should be able to sensibly assign values in such cases. Given the Procrustes score mean and standard deviation corresponding to each cell (i, j), and given a smoothing parameter K, the process may be described as follows:

1. Assign the observed fragment-pair to a cell (i, j), according to the average principle and second eigenvalues. Standardise the grid so that the cell size is unity, by dividing by maximum observed eigenvalues in the non-redundant dataset (for principle and second eigenvalues, accordingly).

2. Consider all adjacent cells. Identify all cells such that the distance between their average position and the observed eigenvalues is no greater than 1Å. If at least K such cells are identified then continue, otherwise expand the search grid, and repeat.

3. Identify the K cells whose average eigenvalues are closest to the target fragment-pairs' eigenvalues.

4. Calculate the mean and standard deviation of the Procrustes scores, pooling all data from the identified K closest cells. Since some cells contain an extremely large number of observations, the maximum weighting of any particular cell should be limited.

By default, $K = 9$ is chosen, so that adjacent cells from each direction will be selected, in general. Due to the vast amount of fragment-pairs in the non-redundant dataset, it is impractical to actually pool all fragment-pairs in the K selected cells. In fact, it is impractical even to simply read all data from the non-redundant dataset during normal runtime. In order to solve this problem, the means and standard deviations corresponding to all cells are calculated and stored in a library; this compiled information may then be subsequently utilised. This greatly reduces the amount of data

that has to be processed during ordinary runtime. Note that one library is required for each choice of fragment length. The smoothed statistics may be calculated given knowledge only of the means μ_k, standard deviations σ_k, and number of observations N_k in the K closest cells. Since there are some cells that have a disproportionate number of observations (e.g. some helical cells have 3×10^6 observations, which is many orders of magnitude more than most other cells), the number of observations N_k is forced to have maximum value 100. This effectively limits the weight of dense cells. The weighted smoothed mean and standard deviation of the Procrustes score distributions are then given by:

$$\begin{aligned} \mu &= \frac{1}{\sum_{k=1}^{K} N_k} \sum_{k=1}^{K} \sum_{i=1}^{N_k} x_{ki} \\ &= \frac{1}{\sum_{k=1}^{K} N_k} \sum_{k=1}^{K} N_k \mu_k \end{aligned} \tag{4.8}$$

$$\begin{aligned} \sigma &= \sqrt{\frac{1}{\sum_{k=1}^{K} N_k - 1} \sum_{k=1}^{K} \sum_{i=1}^{N_k} (x_{ki} - \mu)^2} \\ &= \sqrt{\frac{1}{\sum_{k=1}^{K} N_k - 1} \left(\sum_{k=1}^{K} \sum_{i=1}^{N_k} x_{ki}^2 - \frac{1}{\sum_{k=1}^{K} N_k} \left(\sum_{k=1}^{K} \sum_{i=1}^{N_k} x_{ki} \right)^2 \right)} \\ &= \sqrt{\frac{1}{\sum_{k=1}^{K} N_k - 1} \left(\sum_{k=1}^{K} ((N_k - 1)\sigma_k^2 + N_k \mu_k^2) - \frac{1}{\sum_{k=1}^{K} N_k} \left(\sum_{k=1}^{K} N_k \mu_k \right)^2 \right)} \end{aligned} \tag{4.9}$$

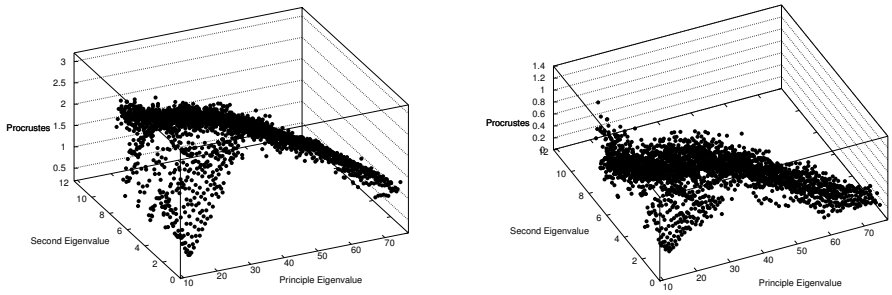
where x_{ki} is the Procrustes score of the i^{th} fragment-pair in the k^{th} selected cell. In principle, the different cells k could be further weighted, e.g. according to eigenvalue distance.

Figure 97 illustrates the effect of smoothing, which is applied to all cells that were originally occupied by the non-redundant dataset. There is a noticeable difference before and after applying smoothing. The smoothing appears to successfully reduce noise on the surfaces corresponding to both mean and standard deviation. Relatively little difference is observed between results when using smoothing parameters $K = 9$ and $K = 25$, thus $K = 9$ is used as the default in implementation. Note that if the cell weight N_k was not limited to 100 then regions of particularly high density, such as the α-helix attractor, would become dramatically over-processed, resulting in extremely skewed distributions (data not shown). The bottom two graphs display smoothing applied to the whole grid, demonstrating robustness to eigenvalue outliers. This may have particular application for some fragments in poorly refined low-resolution structures.

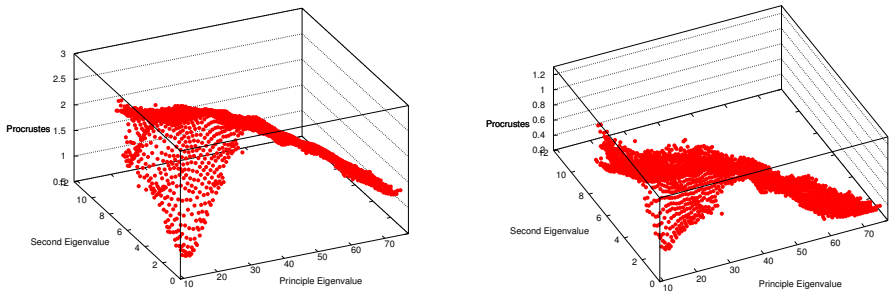
Smoothing with $K = 9$ is applied in all subsequent analysis.

4.3.2 Standardised Procrustes Score

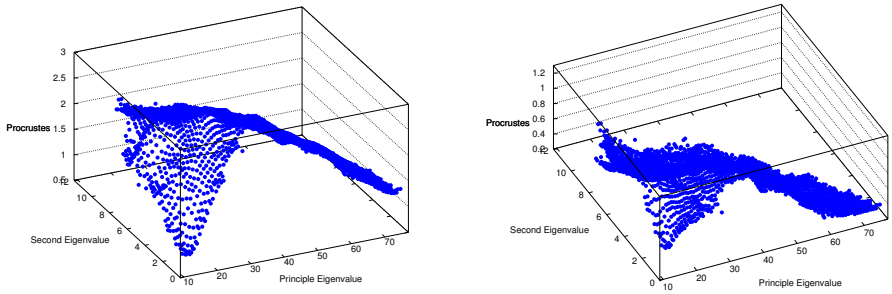
Now that we have an estimate of the mean and standard deviation of the Procrustes score distribution, given knowledge of $\bar{\lambda}_1$ and $\bar{\lambda}_2$, we are able to standardise any observed Procrustes scores. Specifically, given the fragment length n, we will standardise with respect to the first and second



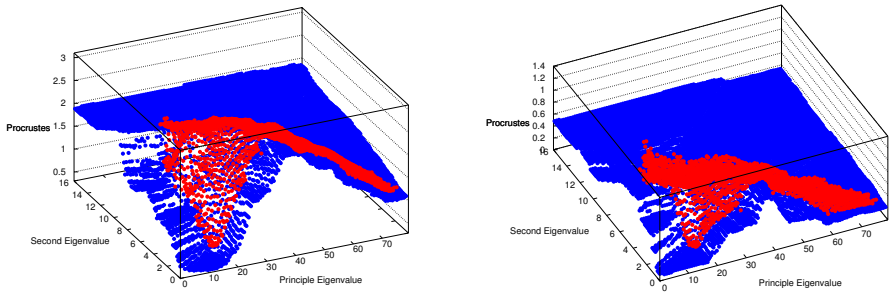
(a) Original data (left: mean, right: standard deviation).



(b) Smoothed with $K = 9$.



(c) Smoothed with $K = 25$.



(d) Smoothed with $K = 9$ (red), also showing the smoothed projection over the grid (blue).

Figure 97: Statistics of the Procrustes score distributions against principle and second eigenvalues. Graphs on the left show average Procrustes score; those on the right show standard deviation. Subfigure (a) shows the original data from the non-redundant dataset. Subfigure (b) shows the statistics after smoothing with $K = 9$, and (c) with $K = 25$. Subfigure (c) shows the statistics for $K = 9$, with data corresponding to the non-redundant dataset displayed in red, and the smoothed projection to all other grid locations shown in blue.

moments of the distribution, achieving the standardised Procrustes score:

$$\hat{d}(\bar{\lambda}_1, \bar{\lambda}_2) = \frac{d - \mu(\bar{\lambda}_1, \bar{\lambda}_2)}{\sigma(\bar{\lambda}_1, \bar{\lambda}_2)} \tag{4.10}$$

where d is the observed Procrustes score arising from the comparison of two fragments, with average principle and second eigenvalues $\bar{\lambda}_1$ and $\bar{\lambda}_2$. The mean $\mu(\bar{\lambda}_1, \bar{\lambda}_2)$ and standard deviation $\sigma(\bar{\lambda}_1, \bar{\lambda}_2)$ are calculated (using 'on-the-fly' smoothing) from the appropriate library of n-residue fragments created from the non-redundant dataset.

Figure 98 shows the distributions of the Procrustes score and the standardised Procrustes score, for fragments in the non-redundant dataset. The standardised Procrustes score seems reasonably well-behaved, given the suspected degree of non-Normality and multimodality. The wider region corresponding to $\bar{\lambda}_1 < 30$ is expected, being seemingly consistent with the very high frequency of observations in those cells (see Figure 93).

The standardised Procrustes score is amongst those known as z-scores (although it might be considered a t-statistic, being derived from a sample and not a population, although the distinction is unimportant in this case). However, due to the inherent non-Normality (discussed above), this score does not satisfy conditions required to be converted directly into a probability. Consequently, care should be taken when using this score to make strict statements regarding significance. However, the graphs in Figure 98 seem to suggest that an implication of significance would not be

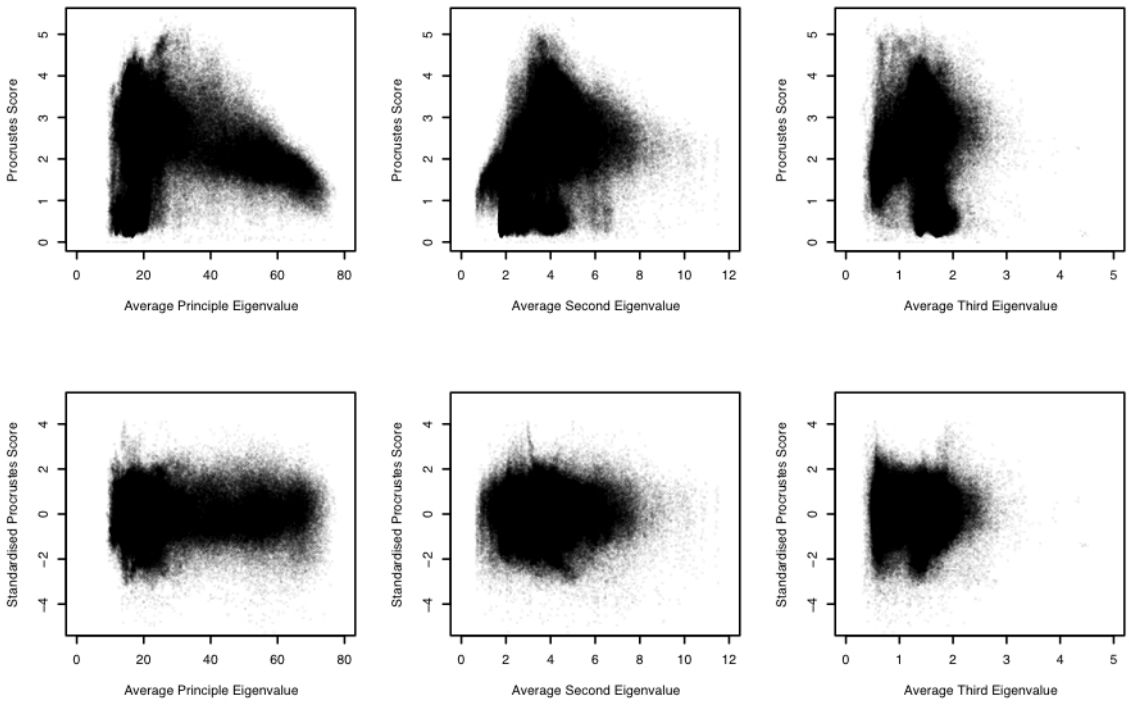


Figure 98: The three upper graphs show Procrustes score versus average eigenvalues for 9-residue fragments in the non-redundant dataset, for fragments with sufficiently similar eigenvalues ($\alpha = 0.1$), as shown in Figure 92. The three lower graphs display the corresponding standardised Procrustes score distributions.

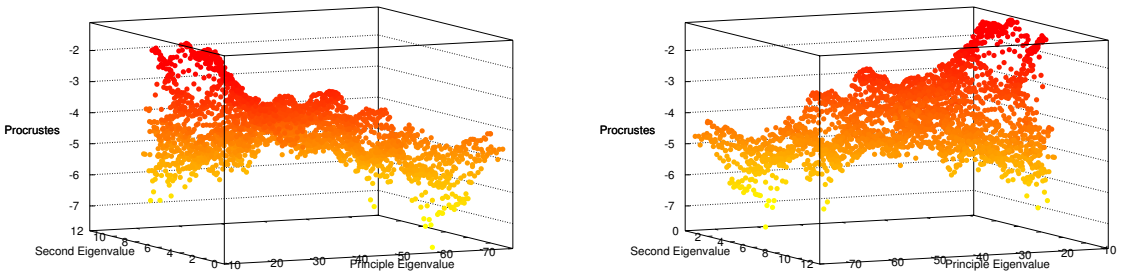


Figure 99: Two views of the standardised Procrustes score, corresponding to the Procrustes score $d = 0$ at all points. The red–yellow colour gradient illustrates magnitude of standardised Procrustes score.

unreasonable, providing a loose interpretation of such results is assumed.

Most importantly, at a glance, the standardised Procrustes score seems ideal for the purpose intended: to allow better comparison of scores resulting from the comparison of the major different types/classes of fragments. Specifically, any scores that would be observed from the comparison of random helical fragments will be on roughly the same scale as that of strands, and that of loops.

Since the Procrustes score is bounded below (by zero), the standardised score is also bounded. However, as illustrated in Figure 99, the lower bound varies with principle and second eigenvalue. This bound is high for particularly favourable conformations (e.g. helices) and low for less favourable conformations (e.g. particularly straight strands, and particularly globular loops). This means that if an observed fragment-pair has eigenvalues suggesting a conformation that randomly exhibits low Procrustes scores (e.g. helices), then observing them to have a low Procrustes score would not be particularly surprising. However, if the fragment-pair has eigenvalues observed to generally have high Procrustes scores by random, then the observation of low Procrustes scores would be surprising, and thus the significance of such information would be more extreme.

This behaviour is vital for the intended application in the alignment of structures. Specifically, the instance of well-scoring helix fragments should have little significance, and thus effect on the alignment. However, the instance of equally well-scoring loop fragments will be identified as more significant, and thus will have a greater influence on the alignment.

4.3.3 Summary

So far, we have only considered fragment-pairs whose eigenvalues are sufficiently similar. The assumption is that fragment-pairs with less similar eigenvalues will inherently have worse Procrustes scores. Evidence supporting this assumption is provided in Figure 100, which displays the scores against eigenvalues for all fragment comparisons within a single (arbitrarily chosen) structure.

Importantly, all possible fragment-pairs are shown, not only those that have similar eigenvalues. The major result here is that we can expect to observe no (or at least, not many) false-positive results arising from fragment-pairs with dissimilar eigenvalues. If the eigenvalues are different, then the resultant Procrustes score will inherently be reasonably poor, and, importantly, the transformation applied in order to achieve the standardised Procrustes score seems to preserve this relatively poor score. This is indicated by a lack of low standardised Procrustes scores when including fragment-pairs with dissimilar eigenvalues. This behaviour is ideal, and exactly as intended; since we are only interested in identifying fragment-pairs deemed sufficiently similar, we are not interested in how badly fragment-pairs may score, and consequently are only interested in scores better than that would be expected by random.

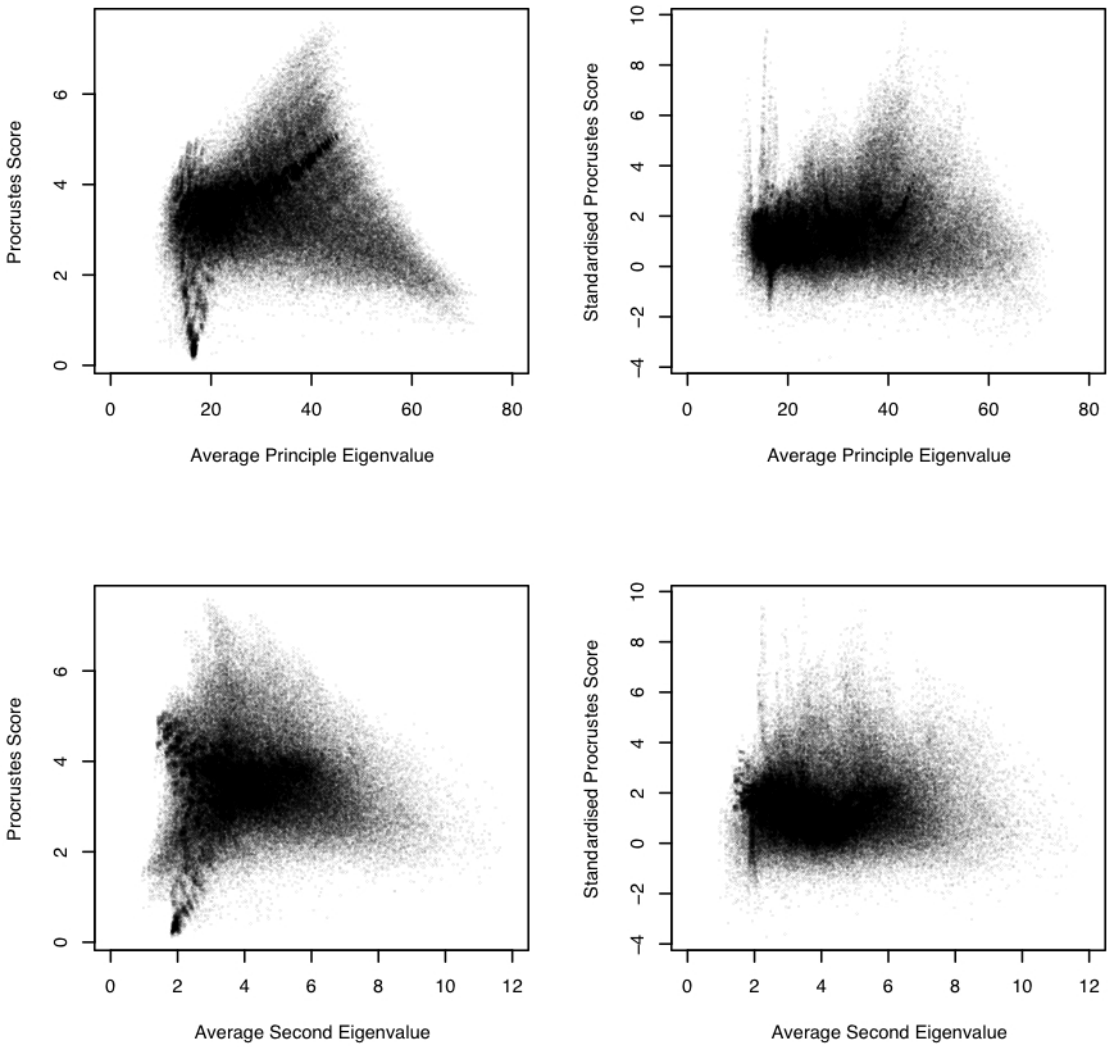


Figure 100: Procrustes score (left) and standardised Procrustes score (right) versus average principle (upper) and second (lower) eigenvalues. All graphs relate to the triangular all-on-all comparison of 9-residue fragment-pairs from a 0.95Å structure (PDB code: 1n1p) comprising 498 residues; 29% helices and 23% strands.

An interesting observation from Figure 100 is that, when including fragment-pairs with dissimilar eigenvalues, the observed distribution of standardised Procrustes scores has a definite tendency towards poorer (higher) scores. Note that the majority of scores are higher than zero, whilst hardly any are below -2 (two standard deviations better than the mean for fragment-pairs with similar eigenvalues).

To conclude, we have identified a new score, termed the standardised Procrustes dissimilarity score. This score is complementary to the raw Procrustes score, due to providing different sorts of information. The Procrustes score (RMSD) describes raw structural agreement, whilst the standardised Procrustes score reduces the weight of helices, and to a lesser degree strands, so that the scale of observed scores is more comparable for different fragment types.

The standardised Procrustes score is entirely reliant on the specific shape descriptors used to smoothly describe (and thus effectively partition) fragment conformation space. Consequently, the success of this score will be highly dependent on the suitability of these descriptors, namely the principle and second eigenvalues. It is possible that the use of more (or different) descriptors may improve the description of conformation space. As such, the use of more information would be beneficial, although it is important to note that the dimensional increase from using more descriptors may cause issues in terms of computational viability. Consequently, the score proposed here belongs to a larger class of scores, amongst which it is perhaps the simplest that could possibly capture sufficient information to be useful. Should other descriptors be identified as targets, it would be possible to construct new standardised scores accordingly, using the general approach suggested here.

Potential Modifications to the Approach

In essence, the standardised Procrustes score manages to smoothly distinguish between helices, strands and loops on the basis of helices and strands being extreme conformations, in terms of eigenvalues. Whilst this standardisation manages to place all scores on a comparable scale, it does not provide any succinct information regarding the more general chance of observing such scenarios. Specifically, information regarding the density of $(\bar{\lambda}_1, \bar{\lambda}_2)$-space could be utilised in order to achieve another measure providing different information. However, such an approach is not explored here.

Another potential improvement to the approach may arise from consideration of the distribution of the Procrustes score with a particular $(\bar{\lambda}_1, \bar{\lambda}_2)$ cell (this may only make sense for sufficiently unimodal regions, so would benefit from the use of more descriptors). Further to the observed distributions being non-Normal, many are asymmetric, and thus the median departs from the mean. Consequently, the corresponding standard score will not reliably imply significance of results, meaning that other standardisations may be desirable.

For example, the unimodal region around the helix attractor is highly skewed. The distribution corresponding to the helix attractor minimum is shown in Figure 101, representing the distance

between random helical fragments with sufficiently similar eigenvalues. As illustrated, this distribution may be very reasonably approximated using a skew-Normal distribution (O'Hagan and Leonard, 1976), which has density:

$$f(x) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\lambda\left(\frac{\lambda-\xi}{\omega}\right)\right) \tag{4.11}$$

where ξ, ω and λ are location, scale and shape parameters, respectively, and $\phi()$ and $\Phi()$ are the density and distribution functions of the Normal distribution. The parameters for the skew-Normal distribution may be estimated as (Azzalini, 1985):

$$\begin{aligned}
\xi &= \mu - \omega\delta\sqrt{\frac{2}{\pi}} \\
\omega &= \sqrt{\frac{\pi\sigma^2}{\pi - 2\delta^2}} \\
\lambda &= \frac{\delta}{\sqrt{1-\delta^2}} \qquad\qquad \text{with } \delta = \frac{\gamma_3}{|\gamma_3|}\sqrt{\frac{\pi}{2}\left(\frac{|\gamma_3|^{\frac{2}{3}}}{|\gamma_3|^{\frac{2}{3}} + \left(\frac{4-\pi}{2}\right)^{\frac{2}{3}}}\right)}
\end{aligned} \tag{4.12}$$

where μ, σ^2, and γ_3 are the (estimated) mean, variance and skew of the observed distribution of Procrustes scores, so that the first, second and third moments of the two distributions are equal.

Note that, in our application, this representation is only an approximation and is not derived analytically. The use of such an approach would be justified by the relaxation of the unrealistic symmetric Normality assumption. It may be desirable to utilise such asymmetric distributional assumptions when standardising the scores, where appropriate, in order to design a standardised score that better-reflects significance.

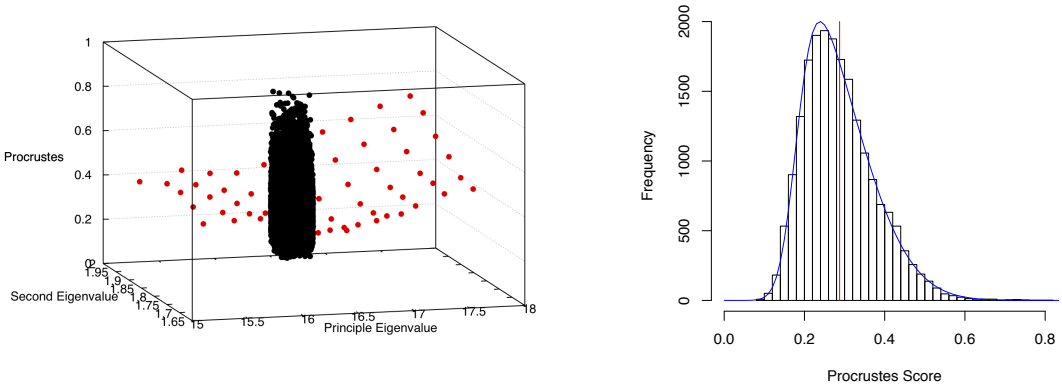


Figure 101: Distribution of Procrustes scores corresponding to the helix attractor minimum. Left: red points correspond to the mean of the distribution in a given $(\bar{\lambda}_1, \bar{\lambda}_2)$ cell, and black points correspond to a sample of 20000 of the fragment-pairs in the non-redundant dataset located in the cell of the helix attractor minimum. Right: histogram of the sampled Procrustes distances in the cell of the helix attractor minimum, with the mean indicated by a red line. The blue curve represents the skew-Normal distribution with location, scale and shape parameters given by $\xi = 0.1803$, $\omega = 0.1389$ and $\lambda = 3.933$, respectively.

4.4 Global Scoring

Now that we have ways in which to score fragment-pairs, we can consider scoring the alignment of protein chains. We must first determine what properties we would like such a score to exhibit. One of the most undesirable aspects of using the average Procrustes score $D = \frac{1}{N}\sum_{i=1}^{N} d_i$, where N is the number of aligned fragment-pairs, as a global measure of conformation-invariant protein chain dissimilarity, is the ambiguity in interpretation of observed scores. This issue is particularly prevalent in 'local' approaches; global methods that consider the rigid RMSD of large structures seem to suffer from this ambiguity to a much lesser degree, surmisably due to the relative sparsity of higher-dimensional conformation space. As previously discussed, a common approach is to consider standardised scores, often derived from a presumably non-redundant database, the intention being that such scores are then comparable. However, when using such an approach, it is vital that methods and results are interpreted carefully and succinctly.

The use of a non-redundant dataset for the purpose of assessing the significance of scores is also adopted here; such an approach has already been utilised in the construction of the standardised Procrustes fragment-based score, noting that in this context the non-redundant dataset is used for scoring individual features as opposed to whole chains. Similarly, we also consider a non-redundant dataset in the context of global scoring, in order to gain insight regarding the practical meaning of such measures. The use of the previously considered non-redundant dataset was deemed unsatisfactory for the consideration of chain alignments. Due to the previously acknowledged bias towards short chains, the resultant alignment length was generally in the region of 20–50 residues; this is unsatisfactory due to the dependency of the variability of $\hat{D} = \frac{1}{N}\sum_{i=1}^{N} \hat{d}_i$ on alignment length:

$$\mathrm{Var}(\hat{D} \mid \bar{\lambda}_1, \bar{\lambda}_2) = \frac{1}{N} + \frac{2}{N^2}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \mathrm{Cov}(\hat{d}_i, \hat{d}_j \mid \bar{\lambda}_{1i}, \bar{\lambda}_{1j}, \bar{\lambda}_{2i}, \bar{\lambda}_{2j}) \tag{4.13}$$

for the simple case where $\mathrm{Var}(\hat{d}(\bar{\lambda}_1, \bar{\lambda}_2)) = 1$ (i.e. approximately equal eigenvalues). In concert, shorter alignments were found to have particularly high score variability. Furthermore, the consideration of such short chains was not considered representative; the results should generalise, especially since most practical alignments would involve the comparison of longer chain-pairs. Note also that aligned fragments are likely to be correlated, since consecutive fragments overlap. Consequently, covariance terms may make substantial contributions, particularly in the event of consecutively aligned fragment-pairs.

A new representative dataset was generated with the added condition of a minimum chain length of 100 residues. This resulted in a dataset of 91 presumably non-redundant chains, see Figure 102. Whilst only this dataset is considered here, the consideration of different representative datasets, with a variety of higher chain lengths, would be desirable in future for purposes of cross-validation; the general results should be independent of the actual dataset chosen. Importantly, the trends observed, in terms of random fragment-pair comparison, seemed generally the same as that observed

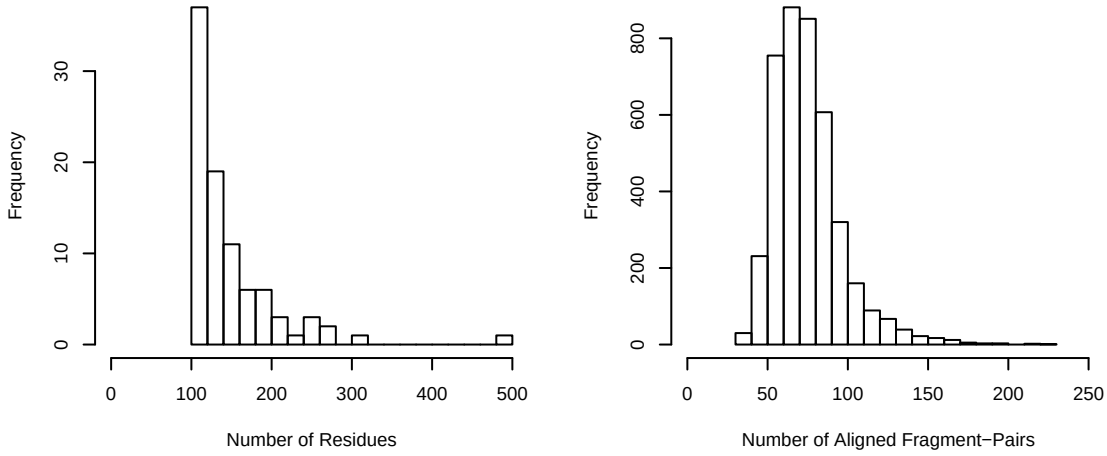


Figure 102: Histograms of the number of residues per chain (left) and the number of aligned fragment-pairs resulting from the all-on-all *ProSMART* alignment of chains in the dataset (right), for the 91 chains selected for inclusion in the non-redundant dataset, with 10% sequence similarity threshold, by *PDBselect*.

for the previous representative dataset.

There are various parameters involved in the construction of the standardised Procrustes score $\hat{d}$, most notably the eigenvalue filtering threshold, the number of boxes in the $(\bar{\lambda}_1, \bar{\lambda}_2)$ grid, the minimum number of observations per considered box, and the number of boxes used for smoothing. For the purposes of the present exploration of global alignment scoring, these four parameters were set to 0.1, 200, 5, and 9, respectively. Other important considerations include those of methodology; specifically, the choice of non-redundant dataset, the method of smoothing, and the method of alignment. Whilst these are fixed for the present study, it should be noted that other scenarios should be considered in future.

4.4.1 Pairwise Chain Scoring with the Standardised Procrustes Score

Since the alignment procedure is subject to constraints, as determined by the specifics of the alignment methodology, the resultant alignment of fragments cannot be considered to be equivalent to the comparison of a list of random fragment-pairs in the context considered above. One major distinction is that, in the practical alignment of structures, the eigenvalues of compared fragments cannot be considered to be equal. This means that the observed average standardised Procrustes score would generally be approximately greater than zero, with the equality occurring when all aligned fragments have approximately equal eigenvalues. Consequently, the consideration of the scores arising from a fragment alignment is far more complicated than the consideration of those arising from random fragment-pairs. Factors such as the alignment length, and the relative length of the compared chains, may have a substantial influence on global scores. It is necessary to

acknowledge the potential influence of such factors in order to have a meaningful interpretation of observed scores.

The average Procrustes dissimilarity score D has the useful property that identical structures score zero (one of the conditions for being a metric). However, no actual interpretation is given to non-zero values, past the mere notion of dissimilarity. Whilst it would be possible to determine the expected Procrustes score, doing so would have very limited purpose due to the vast variability of scores and heterogeneity of conformation space; e.g. the expected score would likely be very different to that of a pair of helices, or a pair of strands, and thus may not provide overly useful information. In contrast, the standardised Procrustes score effectively considers the conditional expectation $\mathrm{E}(d \mid \bar{\lambda}_1, \bar{\lambda}_2)$ of the score of random fragment-pairs. As we have seen, consideration of the principle and second eigenvalues allows some separation of major regions of conformation space, and thus such a conditional expectation might provide useful information that makes consideration of the expected score a practical possibility. Indeed, there may be other quantities that would increase the informative description further, allowing considerable scope for further exploration in this field.

Fundamental to global scoring is the alignment methodology. The usefulness and interpretation of any global score is highly dependent on the protocol used to align the structures. The philosophical approach behind the presented tool *ProSMART* is well-defined; being interested in the conformation-invariant comparison of structures according to the net agreement of the maximal alignment of fragments at a given level of structural resolution. The method is dependent on the underlying definition of the dissimilarity of structural fragments. Whilst the Procrustes score is the default employed method of fragment comparison, there is no reason why other measures could not be used, whilst still conforming to the method's general approach. In particular, we consider the standardised Procrustes score as a measure of dissimilarity, complementary to the traditional Procrustes score. Use of the traditional Procrustes score will result in an alignment based on the raw agreement of local structure. In contrast, use of the standardised Procrustes score will result in an alignment influenced by the observation of common regions of structure considered to be less likely to occur by random, given a definition of randomness (i.e. according to the non-redundant dataset). The behaviour of the global score will be different depending on which alignment method is chosen.

Here, we focus primarily on the standardised Procrustes score due to its ability to give a more equal weighting to the different major regions of conformation space. Consequently, unlike in other Chapters, all *ProSMART* protein chain alignments in this Chapter were performed using a standardised Procrustes dissimilarity matrix (scores capped at a maximum value of zero) in place of the usual Procrustes distance matrix for use in the dynamic programming algorithm.

Pairwise Comparison of All Structures in the Non-Redundant Database

The standardised Procrustes score is such that random fragment pairs with comparable eigenvalues should score approximately zero, on average. However, if fragments' eigenvalues are not comparable then their score may be extremely poor. This may occur very commonly for dissimilar structures, as indicated by the positive tendency of average alignment scores of chain-pairs from the non-redundant dataset, as shown in Figure 103. However, this may also occur in similar structures – the unfavourability of a single aligned fragment-pair could have a high negative influence on an otherwise well-scoring alignment. In such cases, the presence of happenstance dissimilarities may counteract any observed significant similarities, which is undesirable. It seems reasonable that there is no utilisable information gain in knowing whether a fragment pair is aligned by random ($\hat{d} \approx 0$) or much worse than random ($\hat{d} >> 0$). Indeed, a score indicating much worse than random would likely be a consequence of incorrectly assuming comparable eigenvalues (which is necessary for our approach). Rather, we should only be interested in the identification/scoring of any similarities that might be present. Consequently, we propose for any positive scores to be set to zero, so that a random fragment-pair is given the same score as one identified as worse than random. Accordingly, the presented global dissimilarity score is given by:

$$S = \frac{1}{N} \sum_{i=1}^{N} \min(\hat{d}_i, 0) \tag{4.14}$$

This means that, for both alignment and scoring, the signal corresponding to significant similarities should be less hampered by the overly negative influence of random fragment-pairs.

The global scores arising from an all-on-all comparison of chains in the non-redundant database are displayed in Figure 103. Scores corresponding to the average standardised Procrustes scores with (S) and without ($\hat{D}$) setting positive scores $\hat{d}_i$ to zero. The minimum global score given equal eigenvalues (shown in red) is given by:

$$\hat{D}_0 = \frac{1}{N} \sum_{i=1}^{N} \frac{-\mu(\bar{\lambda}_{1i}, \bar{\lambda}_{2i})}{\sigma(\bar{\lambda}_{1i}, \bar{\lambda}_{2i})} \tag{4.15}$$

This value is the score that would be achieved were the structures identical, or, more correctly, were all aligned fragments identical ($\sum_{i=1}^{N} d_i = 0$), given knowledge of the eigenvalues of all aligned fragments. However, if the eigenvalues of the fragments are very different then this ideal minimum will not reliably estimate the minimum that could possibly exist, given the particular fragments in the alignment. Consequently, rather than using the average eigenvalues, the individual raw fragment eigenvalues are used:

$$\hat{D}_1 = \frac{1}{N} \sum_{i=1}^{N} \max \left(\frac{-\mu(\lambda_{11i}, \lambda_{12i})}{\sigma(\lambda_{11i}, \lambda_{12i})}, \frac{-\mu(\lambda_{21i}, \lambda_{22i})}{\sigma(\lambda_{21i}, \lambda_{22i})} \right) \tag{4.16}$$

In practice, this value (shown in blue) is unattainable unless the structures are identical, although it does provide a more sensible value to represent the lower bound of the global score.

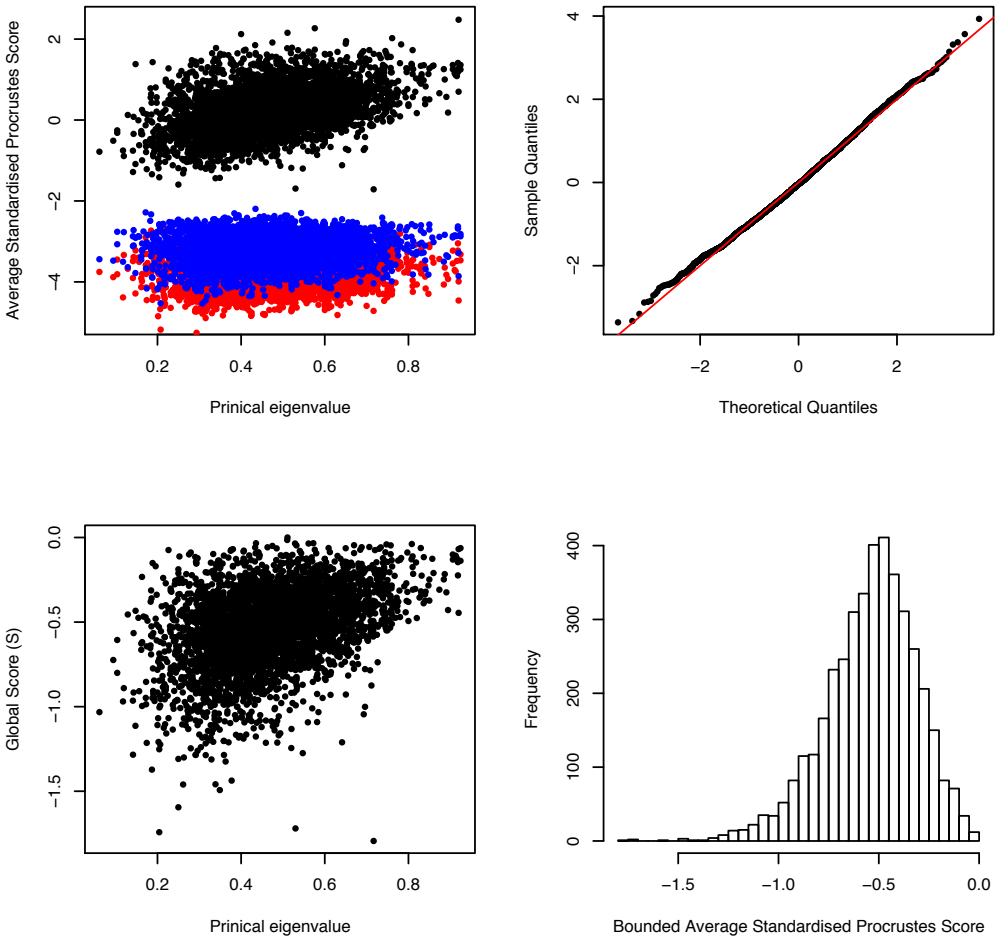


Figure 103: Average standardised Procrustes scores against average of average principle eigenvalues, for the all-on-all comparison of the 91 chains in the non-redundant database. Chains were aligned using the standardised Procrustes score dissimilarity matrix in place of the usual Procrustes score distance matrix. Upper left: black points correspond to the average standardised Procrustes scores, red points to minimum score based on average eigenvalues, and blue points to minimum score based on individual eigenvalues (see text). Upper right: Normal quantile-quantile plot corresponding to the mean and variance-standardised version of the average standardised Procrustes score (black points in upper left graph), demonstrating approximate Normality. Lower left: black points correspond to the bounded average standardised Procrustes score (identified as the global score S in the text). Lower right: histogram of the bounded average standardised Procrustes score.

Behaviour of the Global Score for Random Chain-Pairs

Many factors determine the distribution of the presumed random global scores. A noticeable, but unclear, relationship exists between score and average of average principle eigenvalues. Specifically, chain-pairs with low eigenvalues (mainly helical) tend to have lower global scores. This makes sense, since helices are commonly occurring fragments, which often locate consecutively in sequence (are repetitive), and so helix fragments may be favourably aligned in many configurations between ran-

dom chains. For example, consider the alignment of a short helix with a long helix – in this case there will be a large number of relatively well-scoring alignment configurations; the best-scoring configuration will be chosen, and thus the resultant aligned fragment-pairs cannot be considered random. The same is not true for other fragment types, or at least not to the same extent. Consequently, it would be of benefit to further account for the density of fragment space by considering the frequency of fragments with given eigenvalues. This is not considered here, and is left as an avenue for further exploration.

We have already seen that the majority of scores $\hat{d}$ arising from the comparison of random fragment-pairs are positive; the standardised Procrustes score has zero mean only for fragment-pairs with similar eigenvalues. This means that an alignment of random fragment-pairs (ignoring sequential relatedness) will have a significantly positive average score. However, the alignment process causes the pairing of fragments in a way that results in a collection of seemingly better-than-random fragment-pairs. This is because the objective function used for alignment optimisation is the net fragment score. This will cause the global alignment score to, in general, be better than would be expected from a set of random fragment-pairs, whilst being worse than would be expected from a set of random fragment-pairs with equal eigenvalues. The extent of this will be dependent on the density of conformation space, which is why we observe chain-pairs with low average principle eigenvalues to generally have better global scores.

Furthermore, the chain lengths will be expected to have an influence on the global score arising from the comparison of random structures. Due to the alignment maximisation criteria, there will be a tendency for the alignment length to approach the length of the shorter chain. If the chains are of equal length, then the effective potential number of alignment permutations will be heavily restricted, in practice, by the enforced sequentiality criteria. In contrast, if the chains are of very different length then individual fragments may adopt multiple potential alignment configurations without even influencing the alignment of other fragments. Consequently, the resultant alignment would score better than that of a more heavily restrained random chain-pair of a more similar length.

This effect may be observed by considering the relationship between global score S and the proportion of aligned fragments, out of the number of fragments in the shorter chain (see Figure 104). As expected, there is a strong tendency for lower global scores to be randomly realised when only a small proportion of fragments are aligned. This can be accounted for, removing some of this bias and resulting in an improved estimate of the distribution of random scores. The trend is modelled using a simple linear model. Removal of this trend multiplicatively allows the realisation of an adjusted global score with mean unity, and maximum value zero:

$$S_{\text{adj}} = \frac{\frac{-1}{N}\sum_{i=1}^{N}\min(\hat{d}_i, 0)}{\alpha + \beta\frac{N}{\min(N_1, N_2)}} \tag{4.17}$$

where N is the alignment length, N_1 and N_2 are the numbers of fragments in the two chains, and parameters were estimated as $\alpha = -0.8672$ and $\beta = 0.6980$. As seen in Figure 104, the adjusted

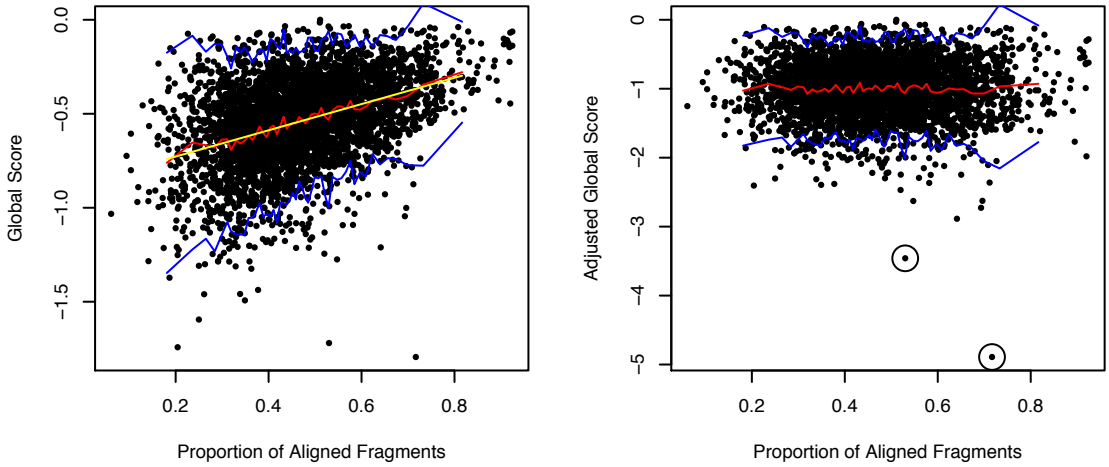


Figure 104: Left: relationship between global score S and the proportion of aligned fragments, given by $\frac{N}{\min(N_1,N_2)}$, for the all-on-all comparison of the 91 chains in the non-redundant database. The red line corresponds to the sampled moving average, blue lines to two sampled standard deviations from the moving average (this is for illustration of variability trends only, and does not imply any statistical significance), and the yellow line to the simple linear regression model of average global score. Right: black points correspond to the adjusted global score S_{adj}, with red lines showing the sampled moving average, and blue lines showing two sampled standard deviations from the moving average. The two most extreme outliers are identified with black circles.

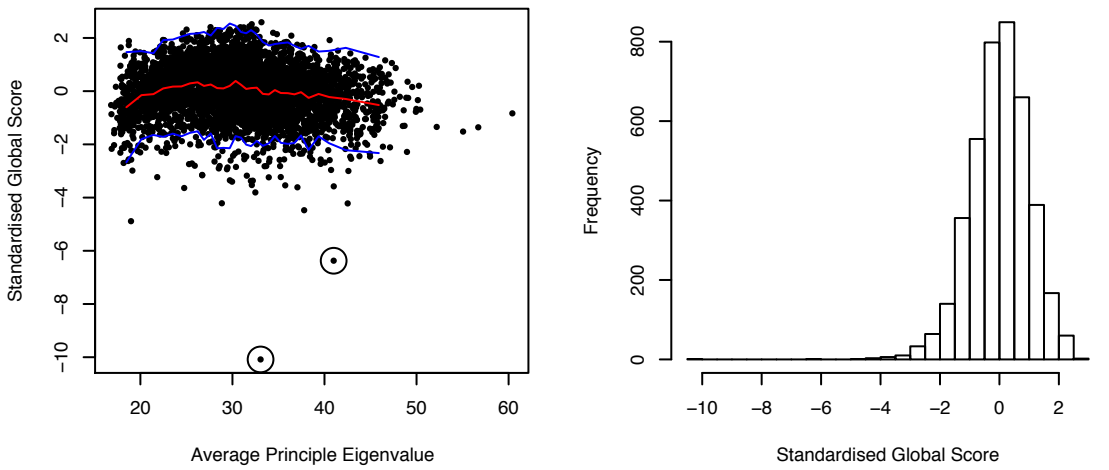


Figure 105: Left: standardised global score $\hat{S}$ against average of average principle eigenvalues, for the all-on-all comparison of the 91 chains in the non-redundant database. Black points correspond to $\hat{S}$, the red line to the sampled moving average, and blue lines to two sampled standard deviations from the moving average (the two outliers were removed for these calculations). Two extreme outliers are identified with black circles. Right: histogram depicting the distribution of $\hat{S}$.

global score S_{adj} removes the observed trend from the data, and results in the score having reduced and stabilised variability for random chain-pairs.

Finally, the global score is then standardised so as to have zero mean and variance unity (a z-score), resulting in the standardised global score:

$$\hat{S} = \frac{S_{\text{adj}} - \mu_{S_{\text{adj}}}}{\sigma_{S_{\text{adj}}}} \tag{4.18}$$

where $\mu_{S_{\text{adj}}} = -1$ is assumed due to the derivation, and the standard deviation is estimated as $\sigma_{S_{\text{adj}}} = 0.3859$. This ensures different scores, such as those arising from the consideration of different fragment lengths, to be on the same scale and thus are comparable.

The distribution of the final standardised global score is illustrated in Figure 105. This score seems better behaved than the original score and interpretation is clear, with large values $\hat{S} \gtrsim 0$ indicating a chain-pair to score worse than random structures in the non-redundant dataset. The number of standard deviations lower than the mean gives a loose indication of significance.

4.4.2 Example: Identified Similarities in the Non-Redundant Dataset

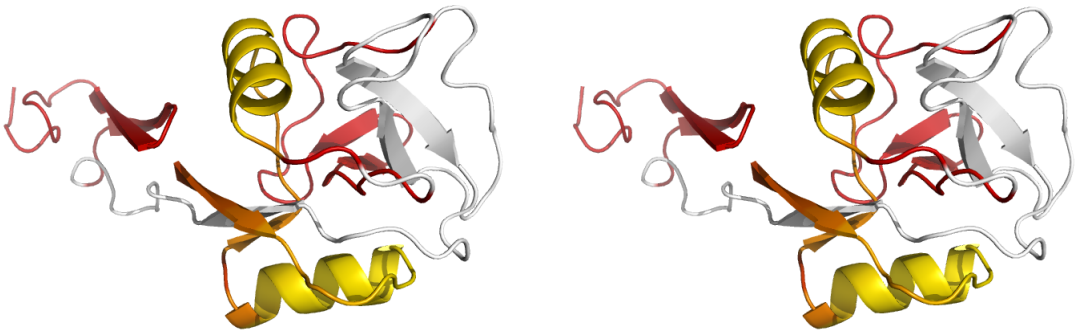
Two extreme outliers were observed when performing an all-on-all comparison of a presumably non-redundant dataset. Both outliers had very significant standardised global scores $\hat{S} < -6$, suggesting each of the two chain-pairs to be significantly more similar than would be expected by random, despite both being present in the non-redundant dataset. These cases will be discussed in this section.

In fact, both of the outliers do indeed correspond to the comparison of similar structures. Both cases are examples of structures with relatively low sequence similarity but high structural similarity. The considered non-redundant dataset was chosen specifically because it only excludes redundancies based on sequence similarity; had it utilised structural information then the results would have been biased according to the particular definition of structural similarity and employed methodology, and thus would have been unsuitable for our purposes. Consideration of the developed standardised Procrustes score enabled these chain-pairs to be easily discriminated from the set of dissimilar chains (although note that it may have also been possible to identify these chain-pairs as outliers using other methods).

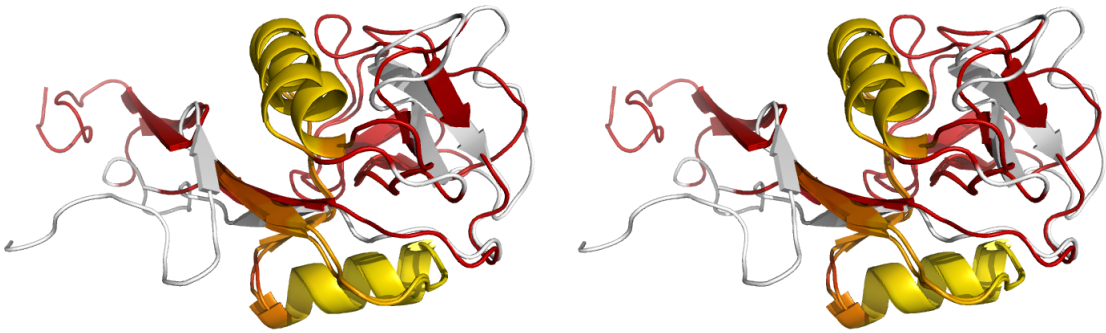
First Extreme Outlier

The first identified outlier ($\hat{S} = -10.08$) was the pairing of structures with PDB codes 3hup(A), an extracellular domain of human CD69 C-type lectin protein (Kolenko et al., 2009), and 1g1t(A), a human E-selectin receptor comprising lectin and EGF domains (Somers et al., 2000). Both of these structures share one common domain, 'Mannose-Binding Protein; Chain A', according to *CATH* (Cuff et al., 2011; Orengo et al., 1997). As can be seen in Figure 106, there is a highly conserved strand-loop-helix-loop-helix motif. However, the rest of the structure is either badly or incorrectly

(a) Stereo view of 3hup(A).



(b) Stereo view of 1g1t(A).



(c) Stereo superposition of 3hup(A) and 1g1t(A).

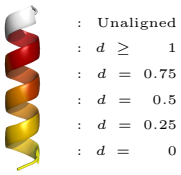
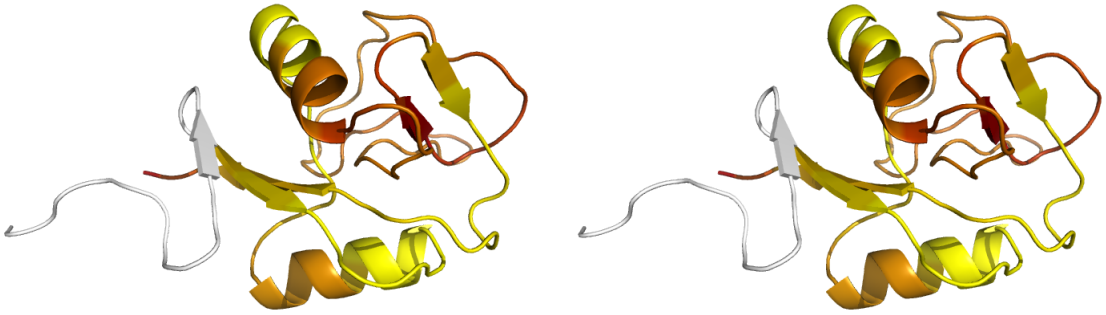
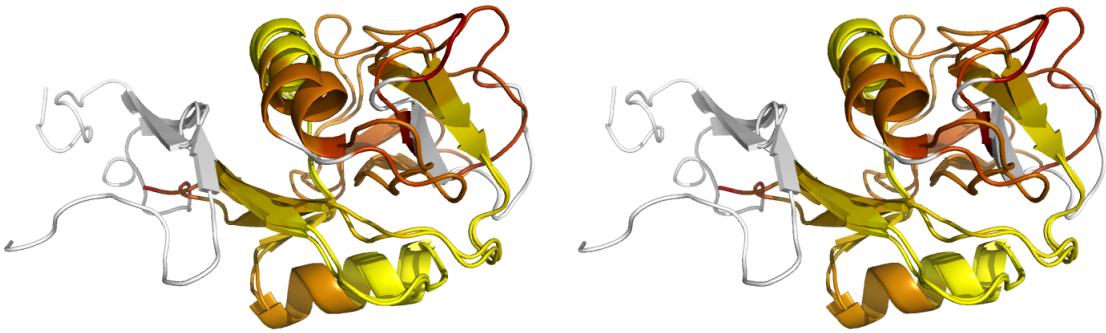


Figure 106: Depiction of structures (a) 3hup(A), (b) 1g1t(A), and (c) superposition of 3hup(A) and 1g1t(A). Residues are coloured by the traditional Procrustes score (minimum score); yellow suggests similarity, red dissimilarity, and white unaligned.



(a) Stereo view of 3hup(A).

(b) Stereo view of 1g1t(A).



(c) Stereo superposition of 3hup(A) and 1g1t(A).

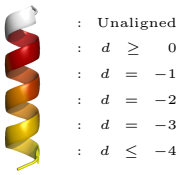
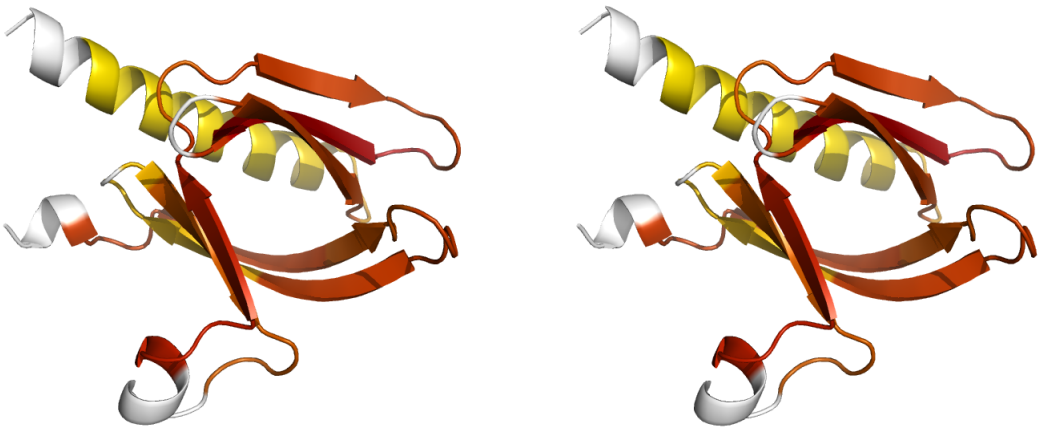
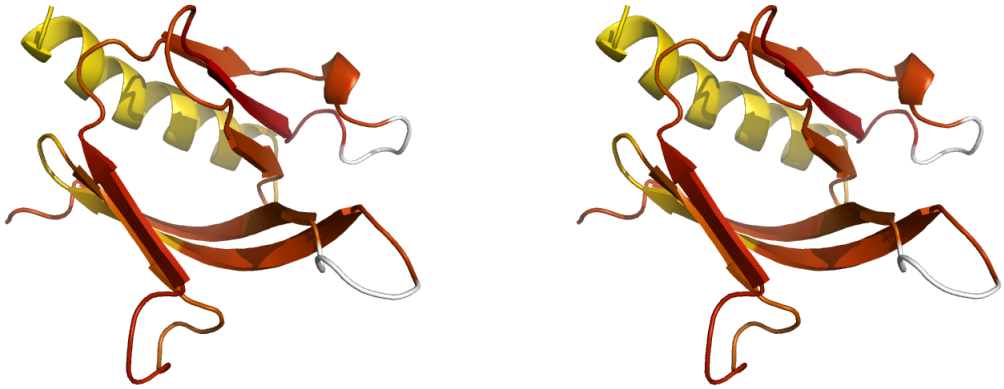


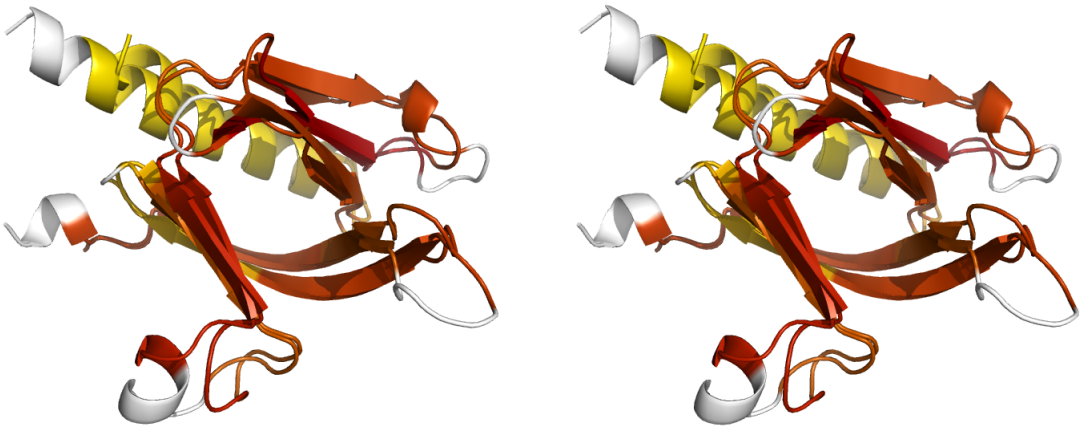
Figure 107: Depiction of structures (a) 3hup(A), (b) 1g1t(A), and (c) superposition of 3hup(A) and 1g1t(A). Residues are coloured by the standardised Procrustes score; yellow suggests similarity, red dissimilarity, and white unaligned.



(a) Stereo view of 1unq(A).



(b) Stereo view of 1u5d(A).



(c) Stereo superposition of 1unq(A) and 1u5d(A).

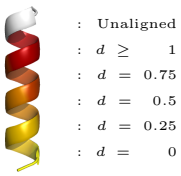
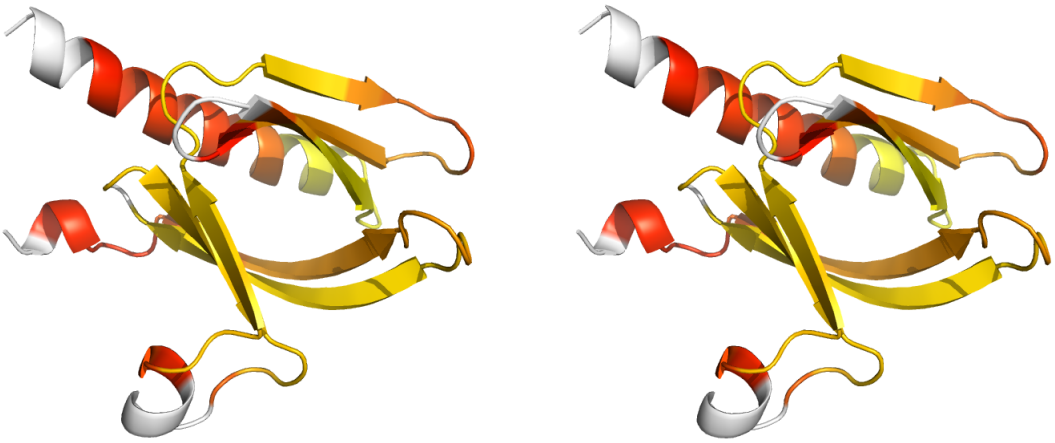
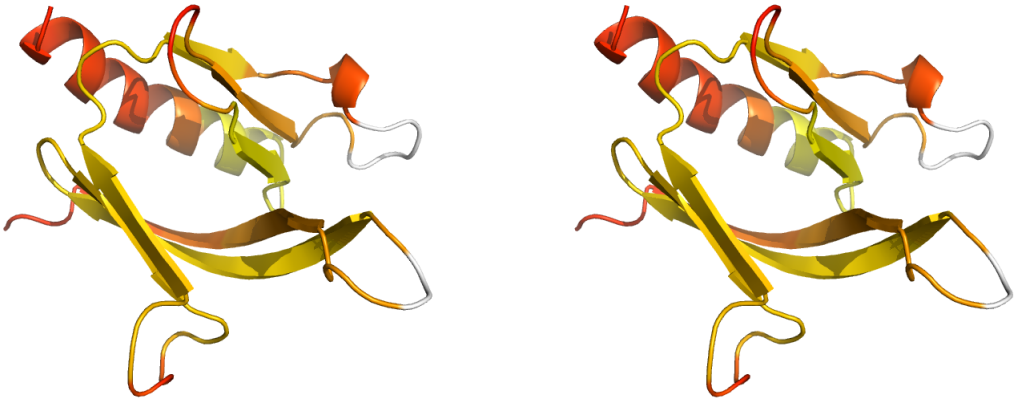


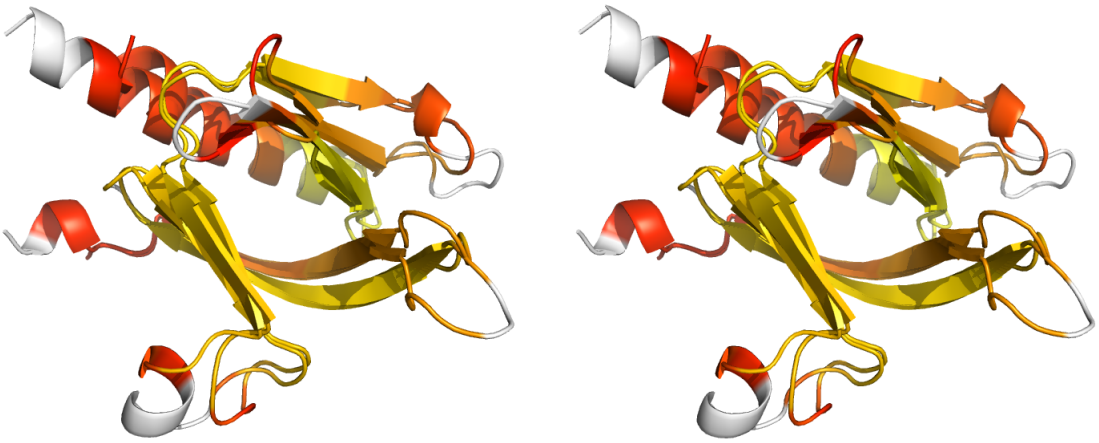
Figure 108: Depiction of structures (a) 1unq(A), (b) 1u5d(A), and (c) superposition of 1unq(A) and 1u5d(A). Residues are coloured by the traditional Procrustes score (minimum score); yellow suggests similarity, red dissimilarity, and white unaligned.



(a) Stereo view of 1unq(A).



(b) Stereo view of 1u5d(A).



(c) Stereo superposition of 1unq(A) and 1u5d(A).

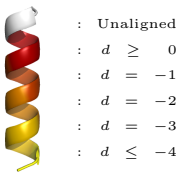


Figure 109: Depiction of structures (a) 1unq(A), (b) 1u5d(A), and (c) superposition of 1unq(A) and 1u5d(A). Residues are coloured by the standardised Procrustes score; yellow suggests similarity, red dissimilarity, and white unaligned.

aligned, when using the default Procrustes score as the fragment dissimilarity measure. The presence of the EGF domain in 1g1t causes particular issues, as part of 3hup is identified as more locally similar to the EGF domain than to the correct region of the lectin domain, due to insertions/deletions and high conformational flexibility of loops. Consequently, this might be considered a hard alignment problem for local methods. For chains with such intuitively similar structures, according to topological agreement, the resultant average Procrustes score was surprisingly poor ($D = 1.485$). This highlights the necessity for the consideration of the standardised Procrustes score in this case.

Figure 107 displays the corresponding alignment based on the standardised Procrustes score. In contrast, much more structural similarity is identified, and the whole lectin domain is aligned sensibly. Due to the very conformationally flexible loops, the traditional RMSD method does not identify equivalent loops as similar, giving a higher weight to the incorrect alignment of other regions. However, the standardised Procrustes method is able to identify the loops as significantly more similar than other random fragment-pairs with similar shape properties. Of the 87% aligned residues (relative to the shorter chain), 14% were identical in sequence.

Second Extreme Outlier

The second identified outlier was the pairing of 1unq(A), a pleckstrin homolog domain of protein kinase B (Milburn et al., 2003), and 1u5d(A), which is also a PH domain. As can be seen in Figure 108, the default Procrustes score is able to align the chains in a seemingly sensible way. However, relatively few of the aligned fragment-pairs score particularly well; as is typical, the alignment of the helix stands out as well-scoring. The core is conserved, although surface loops display some differences. Unlike for the first outlier, here the resultant average Procrustes score suggests similarity ($D = 0.776$). However, it should be noted that this score is heavily influenced by the alignment of the long helix; if this single helix was not present then the average score would be much worse.

Looking at Figure 109, we see a stark contrast – β-strands and conserved loops score well, part of the helix being considered no more similar than would be expected by random. Of the 94% of aligned residues (relative to the shorter chain), 15% were sequence-identical. This example serves to demonstrate how the two scores provide different, complementary, information.

4.4.3 Example: Comparison of Structures from Different Classes

We now consider an example of the global scores that might arise from intra and inter-class comparisons. The purpose of this is only to build further intuition regarding the nature of the developed alignment method and scoring system, not to provide any general and thorough results. Consequently, we demonstrate some typical relationships using a very small and simply-designed sample. Specifically, we consider five arbitrarily selected chains from each of four families: 'L-arabinose

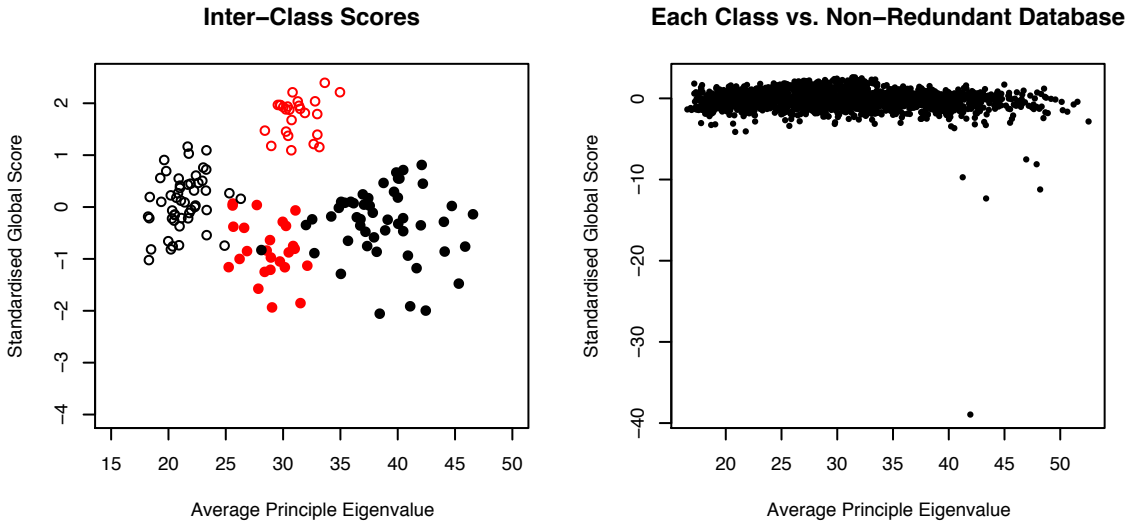


Figure 110: Standardised global score versus average of average principle eigenvalues, showing scores arising from some inter-class comparisons. Left: solid red bullets represent the comparison between the two α/β classes; red open circles the comparison of the all-α class with the all-β class; solid black bullets the comparison of the all-β class with the two α/β classes; and black open circles the comparison of the all-α class with the two α/β classes. Right: black points show scores resulting from the comparison of all four classes with the non-redundant dataset.

binding protein-like', 'Phosphate binding protein-like', 'V set domain' and 'HD domain', according to *SCOP* (Murzin et al., 1995; Andreeva et al., 2008). The first two of these families belong to the same superfamily, and are classified as α/β structures. The third family is classified as all-β, and the fourth as all-α. Specific families and chains were selected arbitrarily.

As can be seen in Figure 110, expected behaviour is observed for the comparison of structures from different classes, whereby the resultant scores are of magnitude similar to that expected from random chain-pairs. Note that the achieved view illustrated in Figures 110 and 111, which considers the standardised global score versus average principle eigenvalue, represents a simple implicit description of protein fold space. Specifically, structure-pairs with a high degree of aligned α-helical content congregate towards the left (low $\frac{1}{N}\sum\bar{\lambda}_1$), and those comprising mainly strands congregate towards the right (high $\frac{1}{N}\sum\bar{\lambda}_1$), with a smooth transition between the two extremes. Observations corresponding to the comparison of all-α structures with all-β structures intuitively congregate in the centre, and are characterised by generally having positive (worse than random) scores. It is interesting that there are six extreme outliers identified between the considered classes and the non-redundant dataset; these will be considered below.

Figure 111 illustrates the intra-family score variability for the four considered samples. It would seem that, based on only this very small sample, score magnitude and variability may depend to some degree on structural class. However, an important point is that the intra-class scores are

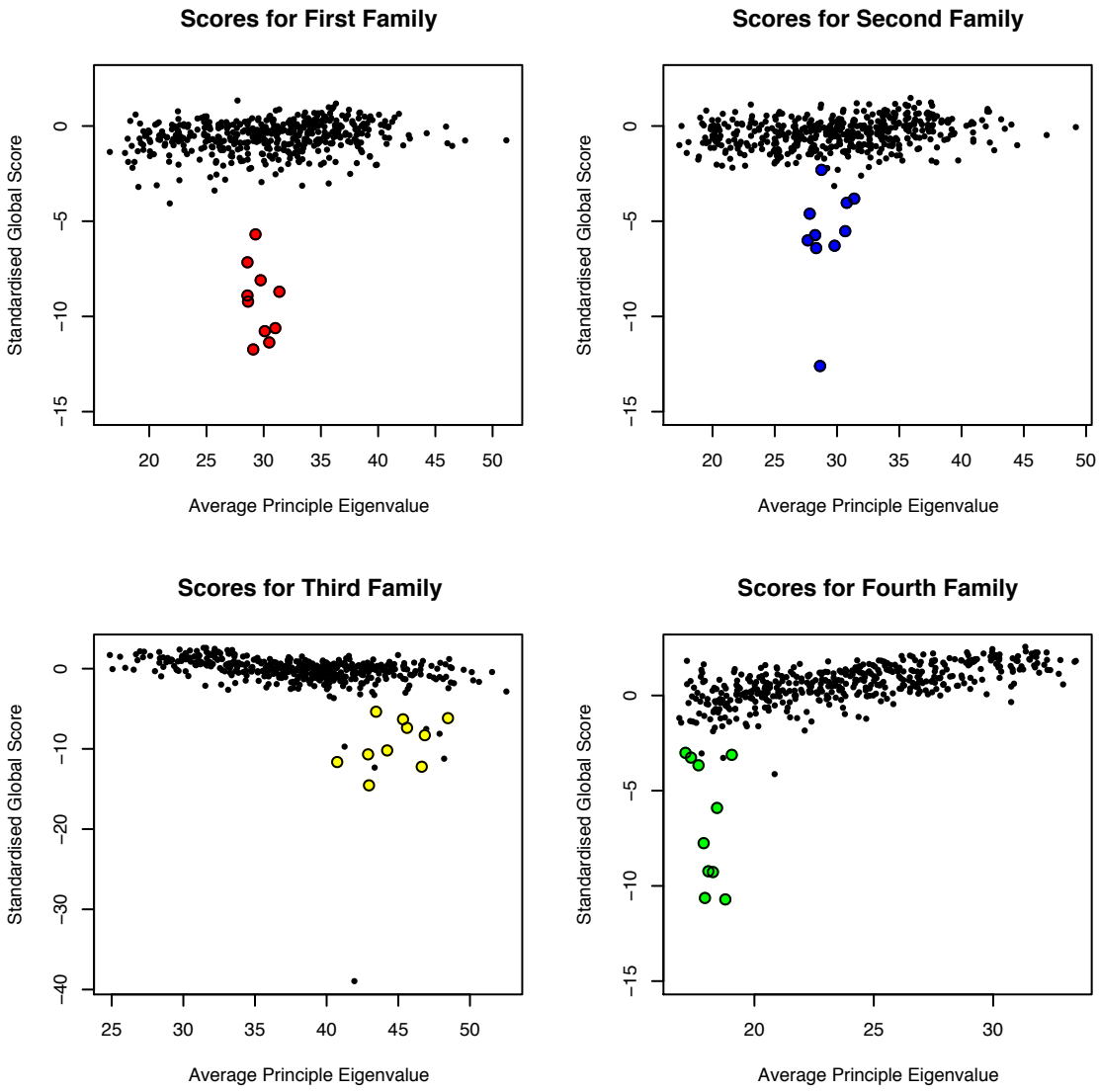
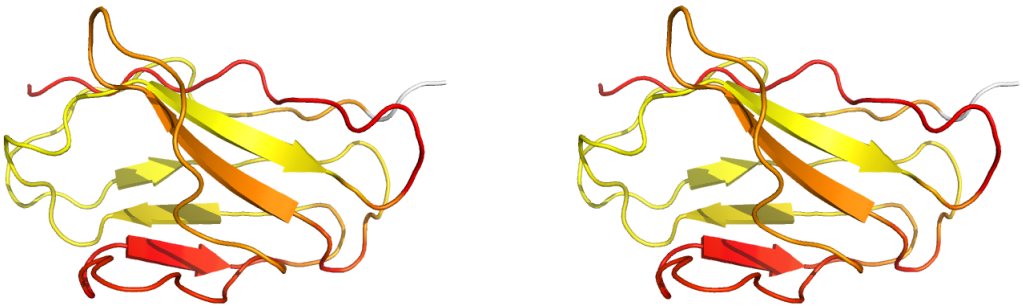


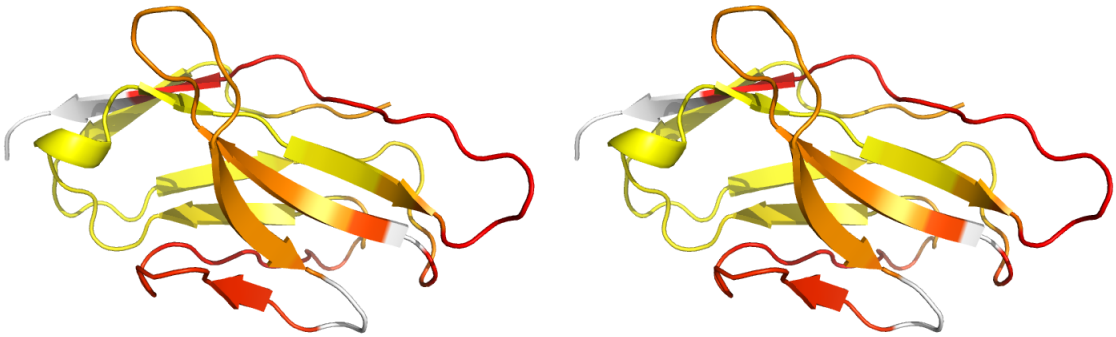
Figure 111: Standardised global score versus average of average principle eigenvalues, within each of the considered four classes, and between each target class and the non-redundant dataset. Intra-class scores are depicted by the large coloured points, whilst scores between each class and the non-redundant dataset are shown as small black points. Graphs are shown corresponding to the sample of the first α/β class (upper left; red points), the second α/β class (upper right; blue points), the all-β class (lower left; yellow points), and the all-α class (lower right; green points).

generally on a much more comparable scale when compared with that achieved using the average Procrustes score. As hoped, the intra-class scores are generally lower than would be expected by random. It is confirmed that the noticeably extreme outliers identified above, each having $\hat{S} < -7$, correspond to the comparison of chains between the all-β class and the non-redundant dataset.

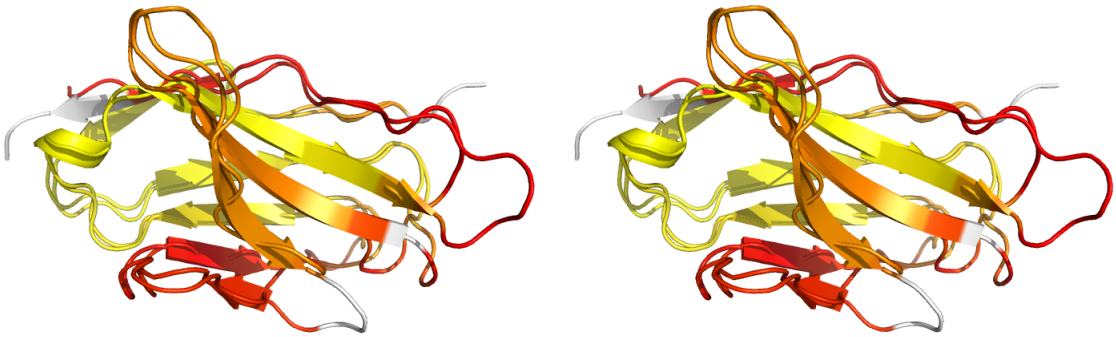
The six most extreme outliers are illustrated in Figures 112, 113, 114, 115, 116 and 117 (in no particular order). It would seem that one of the chains in the non-redundant database, 2q20(A), belongs to the same structural family as the five chains in the considered all-β family, since all five



(a) Stereo view of 2q20(A).



(b) Stereo view of 3bp5(A).



(c) Stereo superposition of 2q20(A) and 3bp5(A).

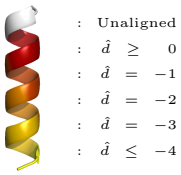
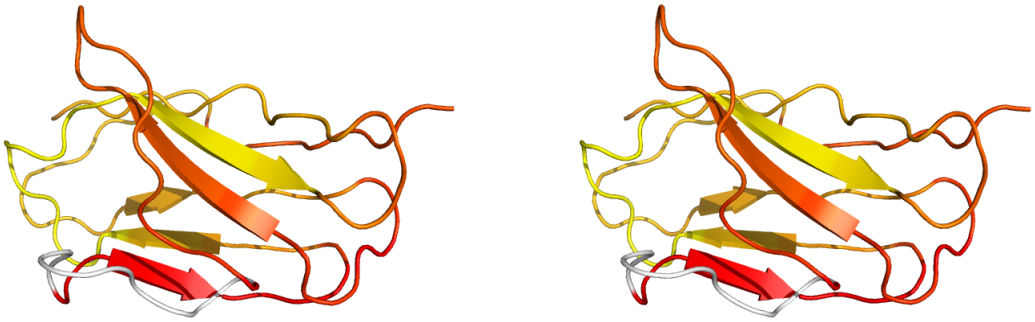
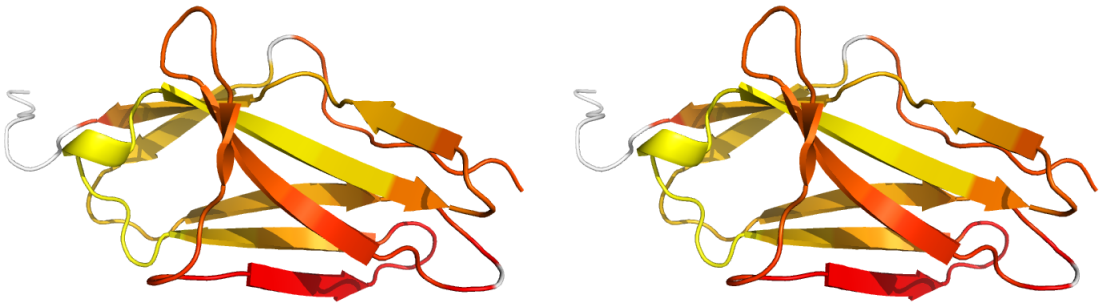


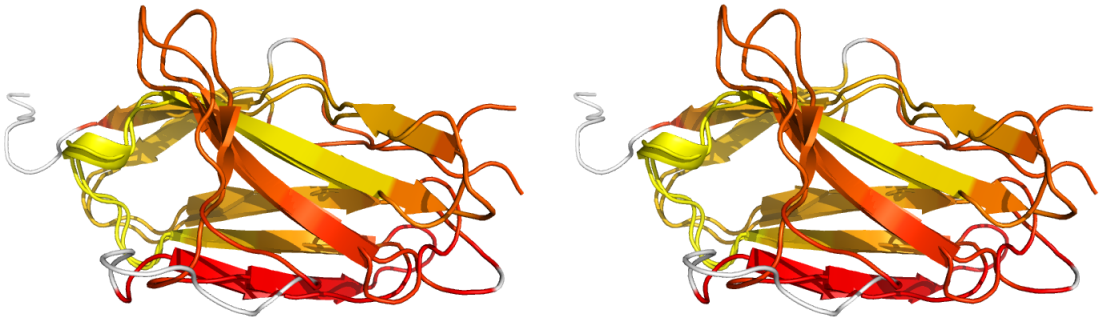
Figure 112: Images depicting alignment and scoring of 2q20(A) and 3bp5(A), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 20%, and standardised global score $\hat{S} = -12.33$. Individual (a,b) and superposed (c) chains are shown.



(a) Stereo view of 2q20(A).



(b) Stereo view of 2aw2(A).



(c) Stereo superposition of 2q20(A) and 2aw2(A).

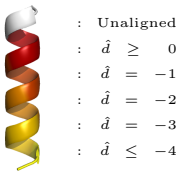
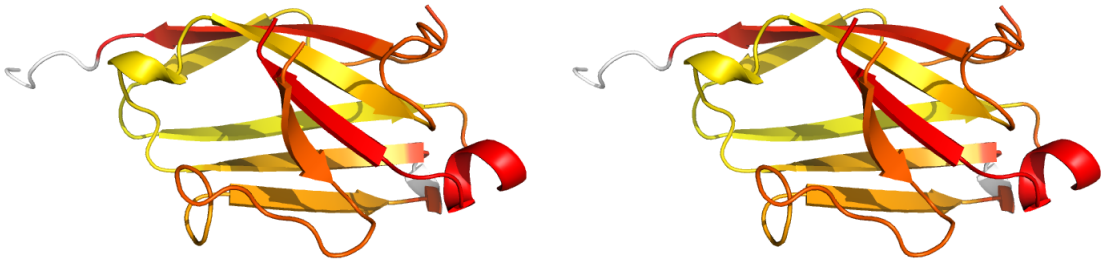
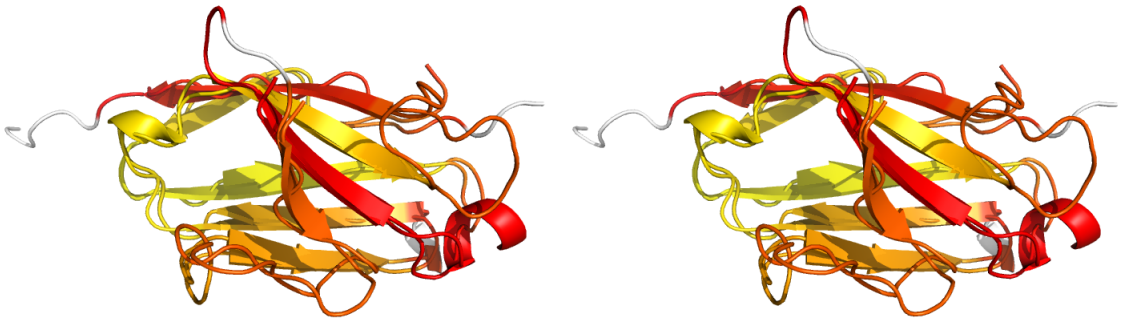


Figure 113: Images depicting alignment and scoring of 2q20(A) and 2aw2(A), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 17%, and standardised global score $\hat{S} = -7.517$. Individual (a,b) and superposed (c) chains are shown.

(a) Stereo view of 2q20(A).



(b) Stereo view of 1xed(A).



(c) Stereo superposition of 2q20(A) and 1xed(A).

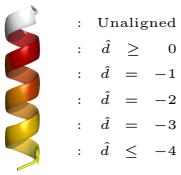
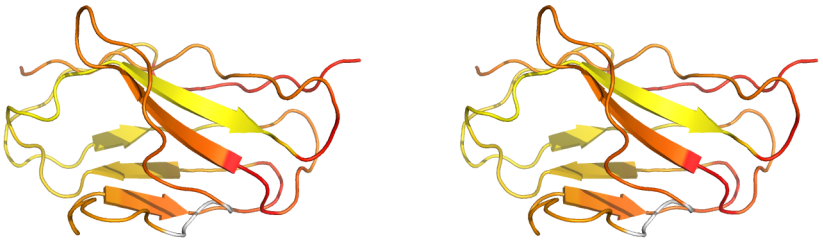
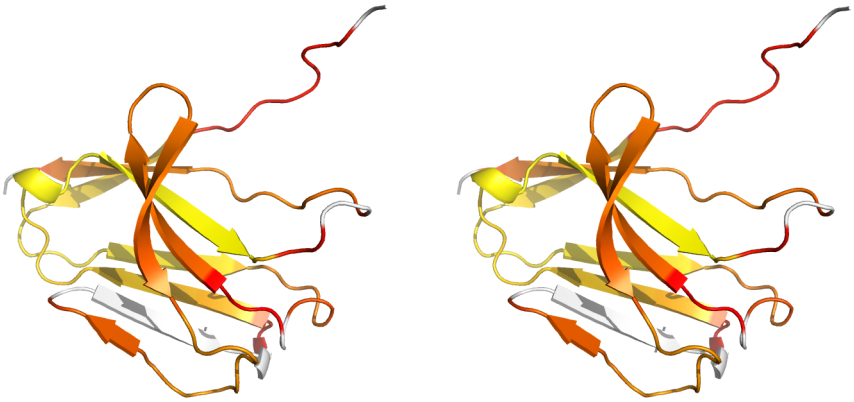


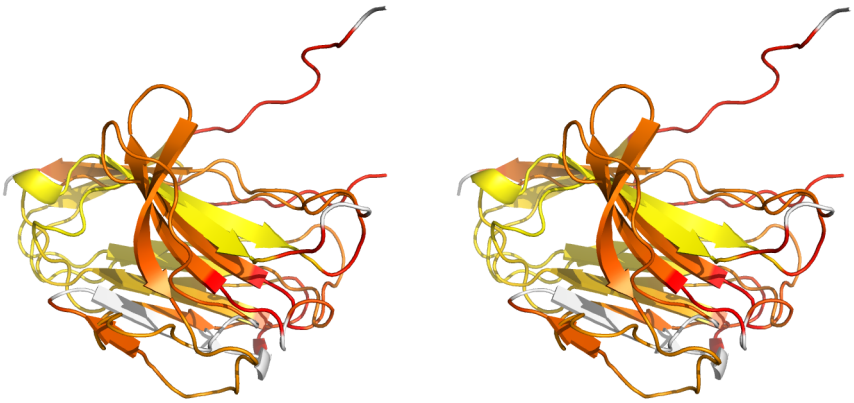
Figure 114: Images depicting alignment and scoring of 2q20(A) and 1xed(A), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 17%, and standardised global score $\hat{S} = -9.1722$. Individual (a,b) and superposed (c) chains are shown.



(a) Stereo view of 2q20(A).



(b) Stereo view of 1q8m(A).



(c) Stereo superposition of 2q20(A) and 1q8m(A).

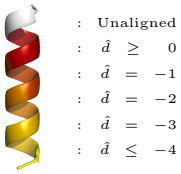
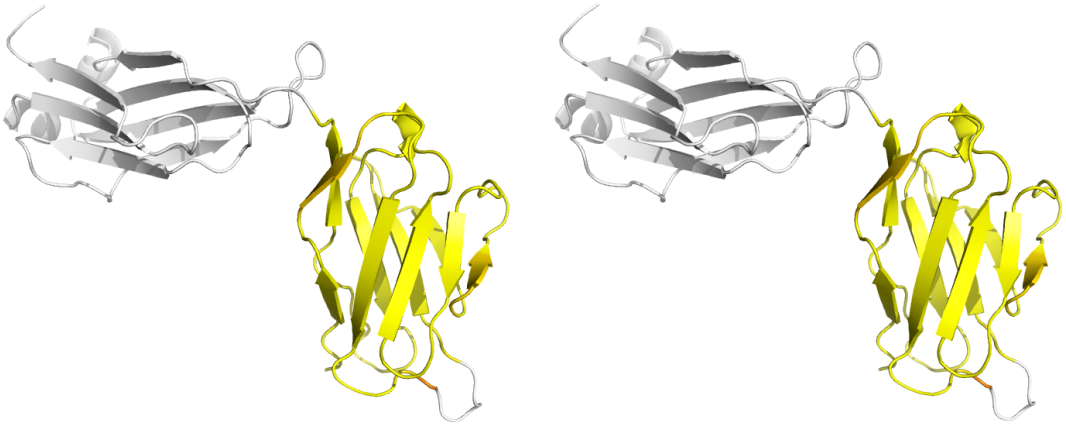
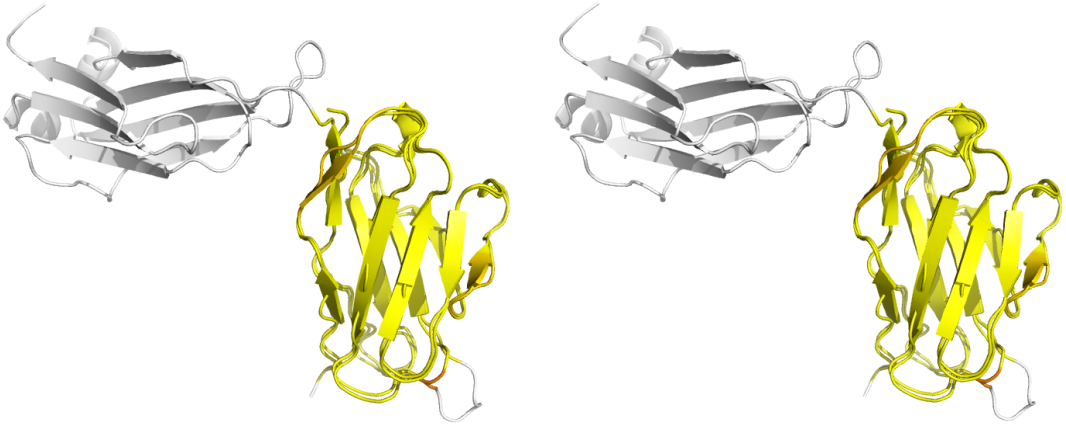


Figure 115: Images depicting alignment and scoring of 2q20(A) and 1q8m(A), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 16%, and standardised global score $\hat{S} = -8.118$. Individual (a,b) and superposed (c) chains are shown.

(a) Stereo view of 2q20(A).



(b) Stereo view of 2ok0(L).



(c) Stereo superposition of 2q20(A) and 2ok0(L).

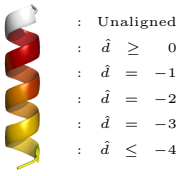
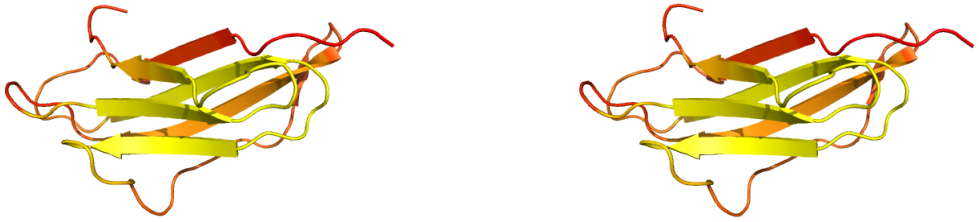
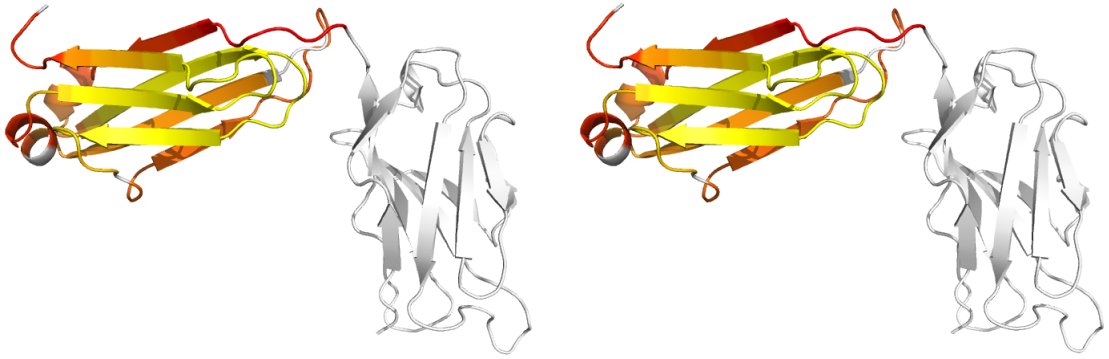


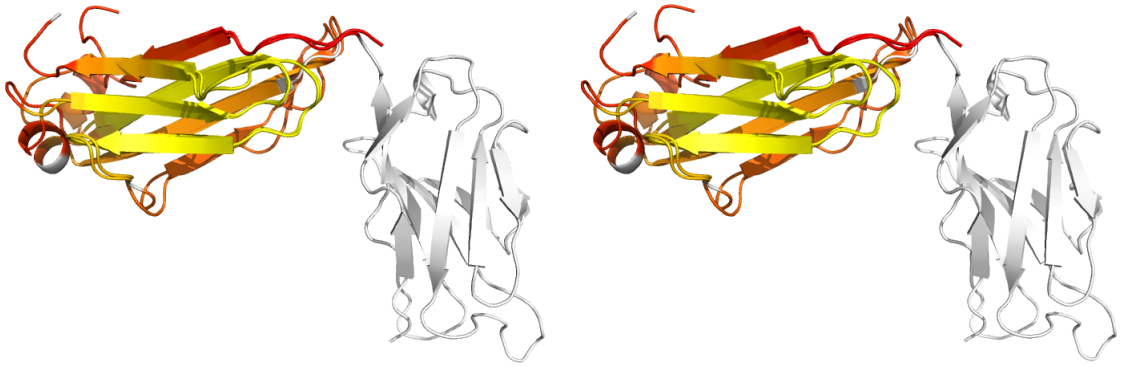
Figure 116: Images depicting alignment and scoring of 2q20(A) and 2ok0(L), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 60%, and standardised global score $\hat{S} = -38.95$. Individual (a,b) and superposed (c) chains are shown.



(a) Stereo view of 1kjv(A).



(b) Stereo view of 2ok0(L).



(c) Stereo superposition of 1kjv(A) and 2ok0(L).

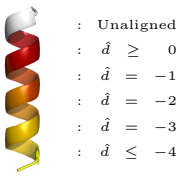


Figure 117: Images depicting alignment and scoring of 1kjv(A) and 2ok0(L), one of the extreme outliers between the considered all-β family and non-redundant dataset. The structures were aligned with sequence identity of aligned residues 13%, and standardised global score $\hat{S} = -11.22$. Individual (a,b) and superposed (c) chains are shown.

pairings are identified as outliers. It is clear that there is a common core shared between 2q20(A) and the five chains, whilst not sharing an overly large overall sequence identity (pairwise sequence identity on aligned residues: 16%, 17%, 17%, 20%, 60%). Fragments in the common core are identified as being significantly more similar than would be expected for random fragments having the same overall shape properties, as depicted by the yellow hues. All chains display the same overall domain topology, apart from an additional β-strand present in 1q8m(A). The high sequence identity between 2q20(A) and 2ok0(L) (60% on aligned residues) is reflected in the identification of their near-identical structural conservation. Since 2ok0(L) comprises two domains, the other domain is identified as similar to another chain in the non-redundant database, 1kjv(B).

Future Improvements

In future, utilisation of the density of fragment conformation space should be considered; taking this into account, perhaps using a likelihood-based approach, may improve separability of random scores. More generally, the method could be refined by considering how the score function could be improved by the use of more descriptors, or using a heuristic approach (as suggested above), with the intention of achieving a better description of fold space by reducing multimodality. A dynamic density-dependent smoothing technique would be preferable to the current homogenous method. Robustness with respect to parameters and specifics of the methodology should also be investigated.

4.4.4 Towards a Multiresolution Approach

So far, we have only considered one specific fragment length $n = 9$. This choice has been reasonably arbitrary; short enough to describe local conformation rather than tertiary structure, whilst long enough to capture information regarding conformation of the chain rather than only high-resolution details. As discussed in the introduction to this chapter, there may be benefits to the consideration of various levels of structural resolution. Consequently, we now consider the generation global scores using a variety of fragment lengths. The purpose of this is to preliminarily investigate: whether similar trends to those observed for $n = 9$ occur for other fragment lengths; whether such relationships in fragment conformation space appear reasonably smooth; and whether similar information regarding global alignment scores is achieved when using different fragment lengths.

Trends in Fragment Conformation Space

Figures 118 and 119 display the relationship between smoothed average Procrustes score and average principle and second eigenvalues for fragment lengths $n = 3, 5, 7$ and 9, and $11, 13, 15$ and 17, respectively. This range was chosen since n must be greater than or equal to 3, and n must be odd (according to implementation in *ProSMART*).

It is clear that qualitative change occurs as fragment length increases. In particular, the highest-

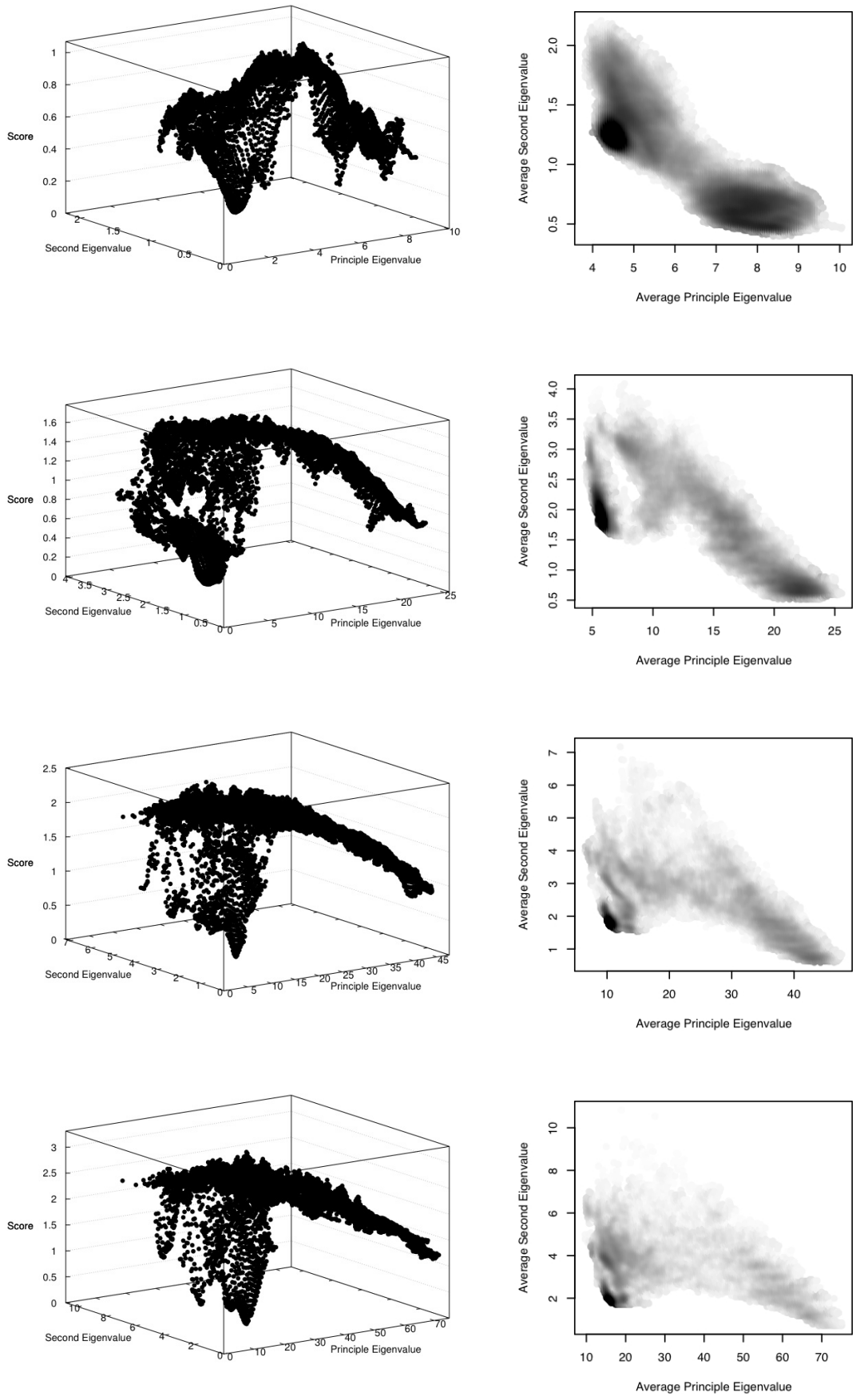


Figure 118: Smoothed Procrustes score against principle and second eigenvalues (left graphs) and corresponding greyscale map of log-density (right images; white: observations < 5, off-white: log(observations) < 4, black: log(observations) > 10), for $n = 3, 5, 7, 9$ (top to bottom).

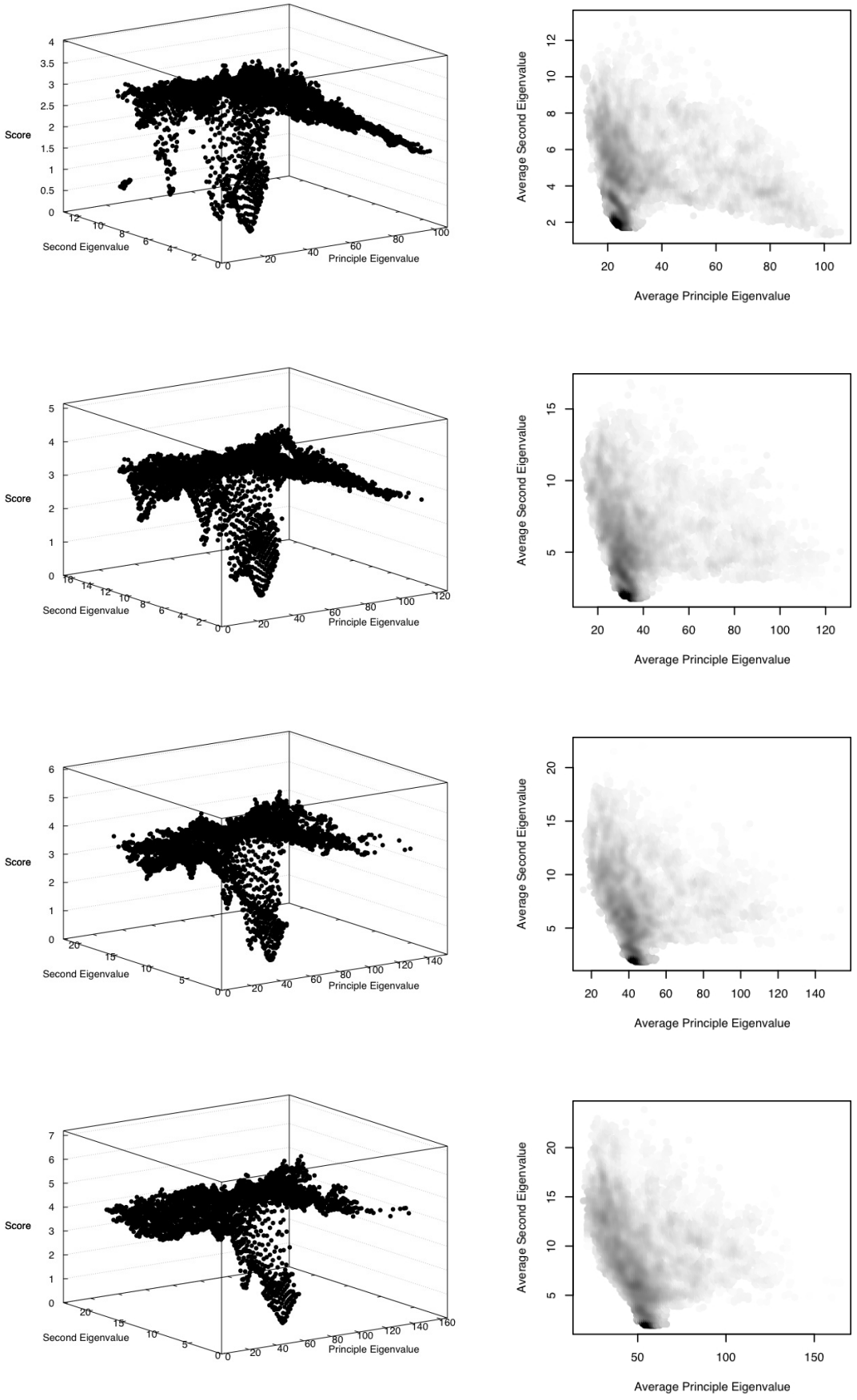


Figure 119: Smoothed Procrustes score against principle and second eigenvalues (left graphs) and corresponding greyscale map of log-density (right images; white: observations < 5, off-white: log(observations) < 4, black: log(observations) > 10), for $n = 11$, 13, 15, 17 (top to bottom).

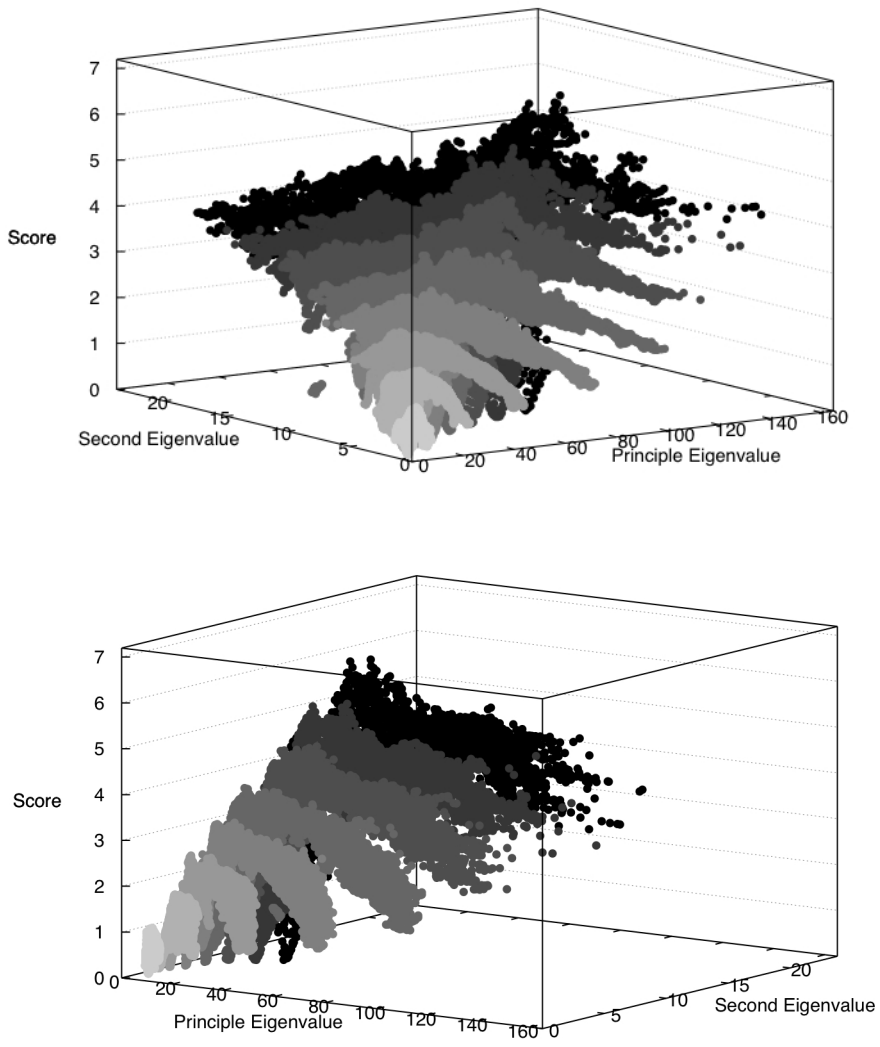


Figure 120: Two views of smoothed Procrustes score against principle and second eigenvalues, corresponding to the all-on-all comparison of fragment-pairs in the non-redundant dataset such that principle, second and third eigenvalues are sufficiently similar ($\alpha = 0.1$). Data are shown for fragment lengths $n = 3, 5, 7, 9, 11, 13, 15, 17$, depicted in greyscale from lightest to darkest, respectively.

resolution length $n = 3$ exhibits noticeably different behaviour. In this case of limited conformational states due to reduced dimensionality, Procrustes score represents extremely detailed structural information.

For fragment lengths of $n \geq 5$, a reasonably smooth qualitative transition is observed as n increases. This behaviour is reflected in the maps of density against eigenvalues. Regions of high density corresponding to helices and strands are observed. The strand attractor is clear for low fragment lengths, with density diminishing as n increases until no longer clearly visible for $n \geq 11$. In contrast the helix attractor (and also relatively high density corresponding to helix-loop conformations) remains for all considered fragment lengths. This behaviour would change were n increased further, past the number of residues in commonly occurring helices.

More generally, overall density of fragment space appears to diminish as n increases, apart

from for favourable helical fragments. This can be attributed to there being fewer favourable strand/loop conformations due to the increased sparseness of the spaces of higher dimensionality (note that cells comprising less than 5 observations are omitted from the Figures 118 and 119). It is for this reason that the qualitative difference is observed for the lowest fragment length $n = 3$, where there are a higher number of favourable conformations in regions of $(\bar{\lambda}_1, \bar{\lambda}_2)$-space. Figure 120 illustrates all the relations shown in Figures 118 and 119, allowing visualisation of the relative scale of the eigenvalues of the fragments of different lengths, and visualisation of the smooth qualitative transition as n increases.

Global Score Profile Vectors

Using the same protocol, and the same non-redundant dataset used for calculating the adjusted score for $n = 9$ above, standardised global scores were calculated for all considered fragment lengths. Quantitative properties of the global score were found to depend on fragment length n. Consequently, it was necessary to estimate the parameters α, β, and $\sigma_{S_{\text{adj}}}$ from Equations (4.17) and (4.18) separately for each considered fragment length. Parameter estimates are shown in Table 5.

The standardised global score $\hat{S}$ has, approximately, mean zero and unitary standard deviation for all fragment lengths, according to the all-on-all comparison of structures in the non-redundant dataset. Consequently, it is possible to directly compare scores arising from the alignment/scoring of chain-pairs using different fragment lengths.

Figure 121 shows the global profile vectors resulting from the all-on-all comparison of chain-pairs in the non-redundant database. The two previously identified extreme outliers are highlighted. The profile vectors of these outliers are qualitatively different. Chains 1unq(A) and 1u5d(A) are identified as similar only for shorter fragment lengths, indicating that details of the structures are similar,

Fragment Length (n)	α	β	$\sigma_{S_{\text{adj}}}$
3	-0.7025	0.4200	0.2617
5	-0.8260	0.5919	0.3310
7	-0.8556	0.6790	0.3710
9	-0.8672	0.6980	0.3859
11	-0.8710	0.6878	0.4078
13	-0.8646	0.6603	0.4267
15	-0.8535	0.6040	0.4392
17	-0.8718	0.6014	0.4494

Table 5: List of the estimated parameter values of α and β from Equation (4.17), and standard deviation of the resultant distribution of the adjusted global score, for considered fragment lengths.

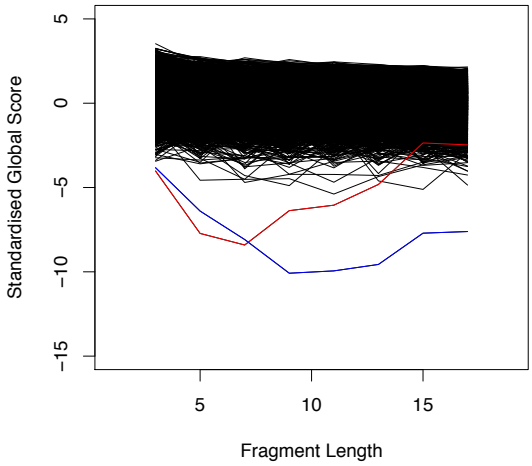


Figure 121: Relationship between standardised global score and fragment length, for all possible chain-pairs in the non-redundant database. Lines represent chain-pair profile vectors. The two extreme outliers are identified in colour; the red line corresponds to the alignment of chains with PDB codes 1unq(A) and 1u5d(A), and the blue line to 3hup(A) and 1g1t(A).

whilst not exhibiting similarity at a lower level of structural resolution. This is indicative of chain-pairs that are similar but conformationally flexible, or those that have many insertions/deletions. In contrast, 3hup(A) and 1g1t(A) are less significantly similar at the local level, on average, whilst being identified as very similar at lower levels of structural resolution. This is indicative of chain-pairs sharing substantial in-sequence regions of extremely well-conserved structure.

The alignment, superposition, and scoring of 1unq(A) and 1u5d(A) are illustrated in Figure 122. The conserved core is identified as significantly similar (yellow) for fragment lengths 7–11. Even though $n = 7$ was found to identify the overall most significant similarities, it is evident that there are regions where $n = 7$ is not ideal; consider the β-strand in the core that is coloured red for $n = 7$, but yellow for $n = 9$. In this case, the ability to combine information regarding the significance of structural similarities from fragment lengths $n = 7$ and $n = 9$ would be beneficial. Consequently, it seems reasonable to conclude that different fragment lengths provide different, complimentary information, and that more information could be gained from the co-consideration of a variety of fragment lengths.

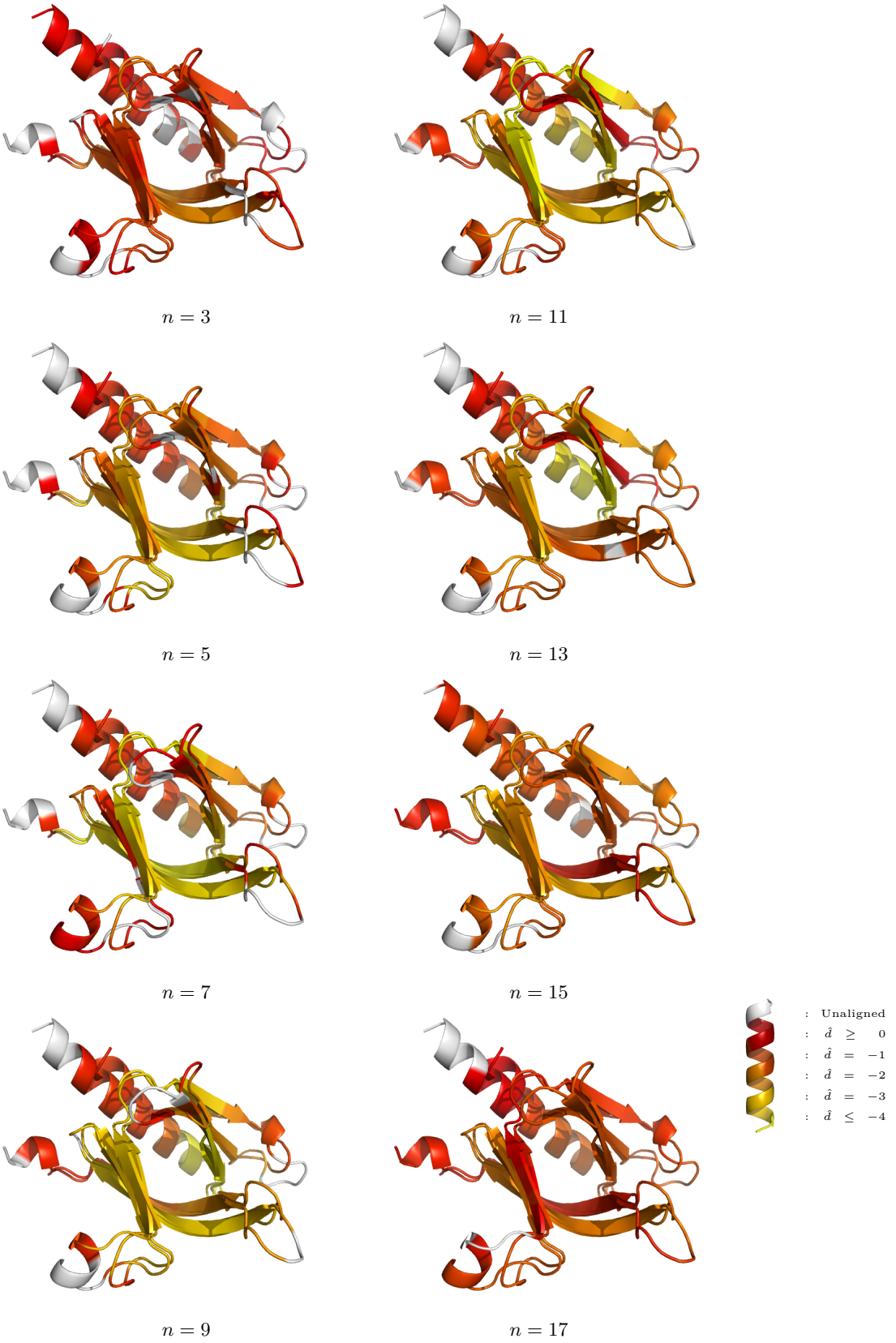


Figure 122: Superposed structures with PDB codes 1unq(A) and 1u5d(A), with residues coloured according to standardised Procrustes score. Images are shown for fragment lengths 3–17.

Profile Vectors for Different Classes

Figure 123 illustrates the global score profile vectors arising from the intra-family comparison of each of the four structural families considered above. The general trends observed are different for each family.

All structures in the first α/β family (red) are identified as similar, with score converging for high fragment lengths. This suggests that, details aside, the (flexible) topology of all of these chains is similar, with no chain-pairs that are particularly more similar than the others.

In contrast, the second α/β family (blue) exhibits a chain-pair that is much more similar than the others. Also, one chain, 1sbp(A), is much less similar to the other four considered members of the family, resulting in scores that might be considered to have marginal significance, having the same magnitude of those observed for some chain-pairs between the family and the non-redundant dataset. This situation typically occurs when structures have relatively low average local similarity, whilst having the same global topology, thus being classified as belonging to the same structural family. Such chain-pairs may be identified as significantly similar using this local method only when using a much longer fragment length, depending on the influence of insertions/deletions.

The all-β class (yellow) exhibits different behaviour, having much better scores for medium fragment lengths 7–13 than for either extreme, and also displaying much more overall intra-family score variability.

The tendency for chain-pairs in the all-α class (green) to score better for higher fragment-lengths is more pronounced than that for the α/β families. This is because the poor significance scores corresponding to helices dominates the global score for low fragment lengths; at higher fragment lengths the structural features that describe the family are identified as significantly different to those found in unrelated helical structures. One chain, 2pq7(A), is far less similar to other members of the family, resulting in a cluster of less significant scores.

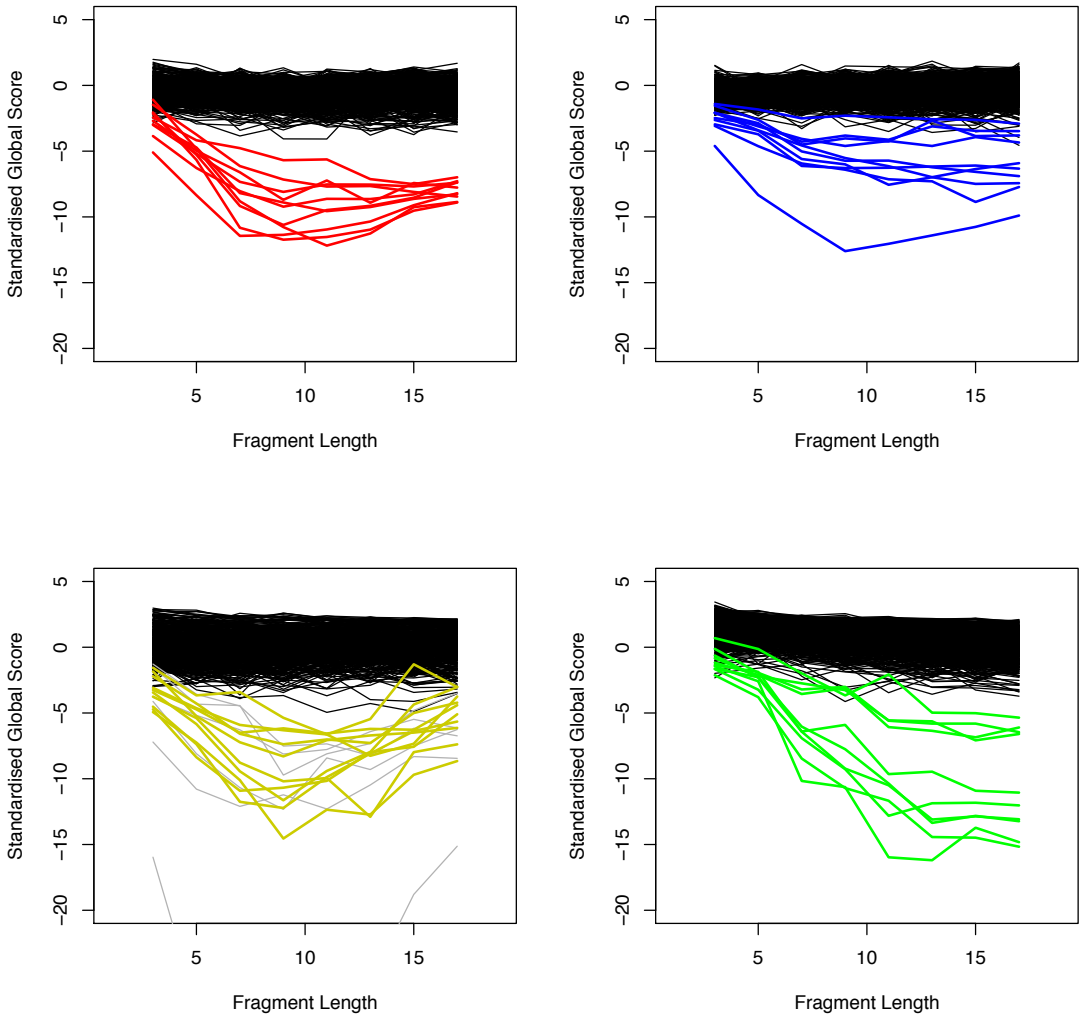


Figure 123: Relationship between standardised global score and fragment length, for each of the four considered families. In concert with Figure 111, the four graphs correspond to the all-on-all intra-family comparison for each of the four considered families. Graphs are shown corresponding to the sample of the first α/β class (upper left; red lines), the second α/β class (upper right; blue lines), the all-β class (lower left; yellow lines), and the all-α class (lower right; green lines). Black lines represent the global score profile vector between chains in the family and the non-redundant database. Grey lines represent the six outliers previously identified.

Chapter 5

Conclusions and Future Directions

The software package *ProSMART* has been produced for the comparison of protein structures in a way that is independent of global conformation. Functionality includes alignment, residue-based and global scoring, rigid substructure identification, visualisation of results by means of superposition and colour-coding, and the generation of external restraints for use in crystallographic refinement. Trends in fragment conformation space have been considered in order to normalise fragment-pair scores, for use in alignment. New methods of scoring the global agreement between protein structures, including the consideration of a multi-resolution feature vector, have been realised. However, such work is preliminary and has much scope for future investigation and improvement.

The software is currently implemented as an automated command line tool. Future work may include attempts to broaden user accessibility. For example, this would be greatly facilitated by integration into the *CCP*4 software suite (Winn et al., 2011). Proper integration with visualisation software (specifically *CCP*4*mg*; McNicholas et al., 2011) would be highly desirable, and allow for powerful improvements over the existing output and functionality. Future ambitions for the standalone tool may include the implementation of an interactive command line version, perhaps with a graphical user interface, and an online service.

Regarding the Flexibility of Local Structure

Variability of atomic positions leads to structural uncertainty, whether systematic, experimental, or dynamic. We have concluded that the raw Procrustes score may not alone be sufficient to unambiguously determine similarity/dissimilarity (at least, in some cases), without also incorporating information regarding uncertainty/flexibility. The consideration of conformational flexibility has been considered elsewhere by utilising thermal parameters during alignment (Mosca et al., 2008). Here, we take a different approach, considering the significance of observed scores, utilising information from a database of known structures, merely taking advantage of general trends.

One of the major problems is that structures are dynamic, not static, yet are often recorded as

adopting fixed conformations following experimental observation. Whilst thermal parameters are often available (whether or not they are reliable), this description is a simplification of the actual system, due to distributional assumptions, and does not capture information regarding true (local) conformational variability, since it does not account for correlations between atomic positions. Indeed, further to the complexity of the high-dimensional problem of structural comparison (using atomic coordinates as raw features), the problem is further complicated by the presence of a temporal component. It is anticipated for future comparison methods to focus on this aspect, perhaps utilising descriptions of fold space or knowledge/simulations of molecular dynamics (although this is not a new idea; e.g. see Kedem et al., 1999; Roach et al., 2005).

Future Additions to the Fragment Scoring and Alignment Approaches

The method of fragment-pair scoring considered here, i.e. Procrustes distance, is only one potential way of assessing the dissimilarity of structural fragments. The consideration of other, meaningful, metrics may provide complementary information to that considered here, e.g. scale invariant measures or other descriptors used in shape analysis.

It may be possible to extend the alignment refinement/optimisation process by including additional stages, providing such stages are found to improve alignments without incurring too much extra computational cost. For example, one possible extension to the rigid segment refinement process would be to allow pairs of fragments to be simultaneously translated in order to increase the number of possible segment configurations tested per refinement iteration. However, such an approach would inherently greatly increase the number of combinations to consider, and may consequently be too computationally expensive for our purpose. Such approaches may be explored in future.

Future Improvements to the Rigid Substructure Identification Method

Our rigid substructure identification method is inherently limited/constrained by the presumed alignment, which, in the present implementation, requires the maintenance of sequence ordering. The removal of the requirement of a prior alignment would be possible. For example, the potential alignment of any fragment-pair could be allowed. However, such an approach would be very computationally expensive, being intractable without further heuristics to speed up the process.

In future, in order to deal with more challenging situations where there are multiple substructures with different variabilities, it may be possible to achieve an improved method by combining the current approach with a hierarchical method of clustering. Specifically, if more sophisticated methods were to be developed in future, the employed single linkage approach may be most well-suited to being used as a clustering pre-filter.

At present, the dissimilarity measure (cosine distance) between pairs of fragment-pairs used during the clustering of fragment-pairs utilises only rotational information. The additional use

of translational information may be beneficial, whether in series or in parallel with the existing approach (e.g. such information could be utilised as a pre- or post-filter). Note that, if the translational component were to be considered, the features of interest would be differences in intra-chain fragment translations, in contrast with the inter-chain fragment rotations employed here for quantifying rotational dissimilarity. Otherwise, results would not be invariant to the protein chains' original coordinate frames.

Furthermore, the translational components of the final rigid substructure coordinate frames are defined so as to optimise superposition of the fragment-pairs used in cluster definition. Whilst this is the most simple and intuitive method, other more sophisticated approaches could be developed in order to further improve the visual appearance of the superposition, similar to that achieved for the rotational component.

Whilst the default parameters seem reasonable for the examples considered, other parameter values may be suitable in different cases, particularly for the identification of less rigidly conserved substructures. Consequently, in future, many cases should be considered in order to develop a method of automatically selecting appropriate parameters. This may be investigated by considering the distribution of inter-fragment-pair cosine distances, and intra-cluster variability. Furthermore, effects of the translation-invariant approach should be investigated. Complementary translation-based clustering may be suitable, if deemed necessary.

External Restraints for Use in Crystallographic Refinement

The use of external restraints has been observed to dramatically improve refinement in some cases, and is expected to be a commonly used technique for dealing with challenging low-resolution structures in future. In order for such restraints to have a positive effect, it is necessary for the restraints to be generated from sensible sources, and applied appropriately. Consequently, this technique is not yet at a stage where it can be applied automatically, since we cannot ensure that the information has a positive effect. The way in which such information is incorporated into the likelihood function needs to be considered further, most notably choice of suitable weight parameters. This choice is not obvious, since it may differ depending on the particular case. It would also be of benefit to explore other methods of sigma estimation (or restraint filtering) in the presence of conformational change, and also outlier detection. The suitability of utilising other variables, such as local alignment scores and/or B-factors, during sigma estimation could also be investigated.

Procrustes Score Normalisation Using Shape Descriptors

The desire to describe fragment conformation space stems from the observation/proposition that different regions (e.g. helices, strands, and loops) require different score weighting, if fragments are to be scored on the same scale. However, we did not want to discretise/categorise fragment conformation space by classifying fragments. Rather, we wanted a sensible continuous approach.

Our major result is that we are able to distinguish these problematic regions (helices and strands) from other regions (commonly categorised as loops) using only the principle and second eigenvalues as descriptors. Importantly, any such fragment categorisation/classification is continuous and implicit; one powerful property of this approach is that it is not necessary to explicitly identify, categorise, or separate such regions. Rather, this approach corresponds to a comprehensive general description/view of fragment conformation space according to the general trends that may be observed using just these two shape descriptors.

Note that, under the current framework, using more descriptors (e.g. third eigenvalue) would add extra dimensions to the problem. This would require the calculation and storage of average and standard deviation of Procrustes scores in 100^k cells, rather than the current 100^2 cells, where k is the number of descriptors. Consequently, using a third descriptor would dramatically effect computation time and storage requirements. This could be handled by the use of extra techniques or heuristics. For example, if many descriptors were to be used, it may be possible to reduce dimensionality using dimension reduction techniques. It is important not to forget that it is intended for this approach to be implemented as part of the alignment methodology of *ProSMART*, and so the proposed method must be simple enough to be computationally feasible. Consequently, given that using just the principle and second eigenvalues seems sufficient to distinguish between the regions of major concern (helices, strands and loops), only the principle and second eigenvalues were considered in the present work.

Towards a Meaningful Global Score Between Structure-Pairs

One key aspect of our approach is that the global score is still meaningful (approximate) even when the residue alignment is not intuitively perfect. Furthermore, the method is consistently applicable for any chain-pair, regardless of their similarity. This arguably sensible abstraction allows meaningful scores to be obtained even for non-homologous structures, leading towards the potential for a new description of fold space (e.g. classification).

Further to producing a global score based on net local conformational agreement, we have demonstrated a global score based on the significance of those similarities. It is further proposed that a multiresolution approach using the global score profile vector would provide more useful information than that which can be achieved considering a single fragment length alone. At the simplest level, the profile vector can be used to determine the optimal score between two chains, over considered fragment lengths. This might be used to identify the most appropriate fragment length for use in alignment, according to the net significance of the resultant aligned fragment-pairs, and may prove to be an indicator of alignment quality. Furthermore, the overall shape of the vector provides useful information about the types of similarity observed between the chain pair, e.g. similarity of details or local topology.

We have observed that even unideal fragment lengths provide useful information, sometimes

complementary to that observed for the ideal fragment length. Consequently, it may be beneficial to combine the fragment scores from different length fragments in some way. To achieve this, it would be necessary to consider relationships between the scores of hierarchically-related fragments, towards a method of utilising multi-resolution relationships for the construction of new descriptors. Such descriptors could be used complimentarily to the shape descriptors suggested above, allowing a more unified description and utilisation of multiple fragment fold spaces. Following an investigation into the hierarchical nature of fragment conformation space, it may be possible to combine scores from the comparison of individual fragment-pairs performed at different levels of structural resolution. Ideally, this should be achieved in a natural way, and not based on arbitrary weighting parameters. This would achieve a different way of scoring fragment-pairs, utilising multiple fragment distance matrices, and achieving a single multi-resolution distance matrix for use in alignment by dynamic programming. Whilst being slower, this extension to the existing method may result in an improved method, at least from a conceptual point of view, since any particular chain pair may exhibit regional similarities at different levels of structural resolution.

Final Words

The aphorisms of Taleb (2010), shown on page 7, have interesting relevance to the present work. Of course, on the surface, our fundamental use of Procrustes analysis complements Taleb's use of the Procrustes metaphor. Note that, in Chapter 2, by removing scale invariance criteria we avoid squeezing structural fragments into a Procrustean bed, making their comparison tangible. Later, in Chapter 4 we avoid the discrete classification of fragments, choosing to suspend categorisation in favour of a more continuous description of fold space, allowing the maintenance of information rather than imposing the subjective narrative that accompanies classification. At the same time, it is important to acknowledge that by adopting specific methods, our heuristics impose subjectivity (e.g. our fundamental definition of similarity; choice of alignment refinement/optimisation algorithms; use of eigenvalues as shape descriptors), and thus cannot be considered comprehensive. Consequently, we must ensure that we do not suffer from "*mistaking the unobserved for the unobservable*", and conclude that it will always be possible to extract more information from such systems. Furthermore, we must be aware that any conclusions drawn using a particular method are simply a narrative, and thus should aim for a succinct interpretation of results, and an appropriate realisation of limitations.

Structural comparison methods may produce seemingly false-positive and false-negative results, in a human-readable context. There may be dissimilar structures that are randomly identified as similar (in some way, whether quantitatively or qualitatively). Also, there may be seemingly similar chains that are not identified as such by a given method. However, the inherent presence of such false results could be used to an advantage rather than seen as a flaw. For example, the identification of similarities between presumed dissimilar structures may raise our awareness of the

existence of such phenomena, resulting in us asking why such similarities are present.

There is no single alignment/comparison method that can identify all types of similarity. Indeed, the identification of similarity is an unbounded problem, and so the combination of information from any number of methods could still in theory be complemented by new approaches. Therefore, when interested in the identification of similarities between structures, it would be of benefit to combine information from a variety of sources, allowing sensible interpretation of the results from each source. If, on the other hand, the identification of a very specific type of similarity is desired, then careful consideration should go into the selection of a method, or methods, that will produce the desired output.

Appendix A

Structural Alignment Approaches and Tools

Here, various existing approaches to structural alignment and scoring are briefly summarised. Some of the implemented structural alignment methods (and other closely related tools) are listed, in order of publication date:

- *SSAP:* Taylor and Orengo (1989b) Globally rigid method, which uses double dynamic programming on a residue-based vector distance matrix. Vectors are constructed using C_β atoms, so as to utilise information regarding side chain orientation. An alignment score is constructed for the favorability of each residue from one structure being aligned to the other, resulting from the optimal path (calculated using dynamic programming) through the matrix of distances between interatomic vectors, relative to the target residue pair. A second round of dynamic programming is performed on the resultant matrix of scores, achieving the optimal alignment. The method was extended to multiple alignment (Taylor et al., 1994). Used for hierarchical fold classification scheme *CATH*.

- *SSAPe:* Taylor and Orengo (1989a) Extension of *SSAP*, which includes other types of information regarding residues properties, including hydrogen bonding, solvent exposure, torsion angles, amino acid type.

- *SARF:* Alexandrov et al. (1992); Alexandrov and Gō (1994)

  Aims to identify similarities between overall folds by detecting rigid substructures common to pairs of structures. Does not require the maintenance of sequence order. First considers how the distribution of the RMSD of random continuous fragments (from random protein structures) increases nonlinearly as the fragment length increases. Uses this information to define a similarity score as a function of RMSD (linear) and fragment length (nonlinear). For a given pair of structures being compared, all C_α backbone fragments (length 6 or 7) that are sufficiently similar (according to the score) between the two structures are identified. Pairs of

fragments (not necessarily consecutive) are iteratively combined, and score recalculated, until the further combination of fragments does not improve the similarity score. This results in potentially large, possibly unconnected, fragments that superpose sufficiently well given their size.

- *STAMP:* Russell and Barton (1992)

  Used for multiple structure alignment and superposition. Performs multiple sequence alignment on all structures, then pairwise superposes the initial alignments. For each pair, using the initial superpositions, structural information (a function of inter-chain RMSD of individual residues and their neighbours) is used to iteratively refine the alignment and superposition. Using a global score representing pairwise superposition quality, all alignments are clustered (using single linkage clustering), allowing superpositions to be sequentially combined according to the cluster hierarchy (clusters are represented by average position of superposed coordinates).

- *Dali:* Holm and Sander (1993); Holm and Sander (1996); Holm and Sander (1998)

  Constructs a C_α distance matrix, then performs a similarity search using a sliding window of 6×6 submatrices, in order to identify similar local structure. Monte Carlo simulations are used to optimise the alignment. Whilst identifying global rigidity, residue pairs at longer distances are down-weighted so as to reduce their contributions to the overall score, thus allowing more flexibility (in score) than methods that score using the traditional global RMSD. A z-score is reported, which normalises the similarity score with respect to the chain lengths. Specifically, distribution (mean and standard deviation) of scores from random structure-pairs is estimated as a cubic polynomial, a function of average chain length $\sqrt{L_A, L_B}$. Used for the *FSSP* classification scheme. The standalone version is called *DaliLite* (Holm and Park, 2000).

- *Protein3Dfit:* Lessel and Schomburg (1994)

  Compares fragments using a C_α distance matrix.

- *VAST:* Gibrat et al. (1996); Madej et al. (1995)

  Globally rigid SSE-based graph theoretical approach, with a focus on applications in sequence-structure threading for use in protein prediction. Identifies possible pairs of corresponding SSEs by considering the probability of the pair occurring by chance, given the number of helices and strands they contain. A common subgraph isomorphism is then calculated, and optimal corresponding C_α alignment is achieved using a Monte Carlo approach. Outputs the global RMSD and number of aligned residues.

- *ProFit:* Martin (1996)

  Tool to superpose structures, using the McLachlan algorithm (McLachlan, 1982). Initial alignment is specified by user, an then a subsequent iterative refinement procedure achieves

the superposition. *ProFit V2.0* allows multiple structures to be superposed, by iteratively averaging the coordinates of superposed structures until convergence.

- *SARF2:* Alexandrov (1996); Alexandrov and Fischer (1996)

  Modification of *SARF* that uses information regarding SSEs. Gives focus to, and examples of, the identification of non-topological structural resemblances, suggesting that such cases indicate energetically stable and favourable conformations, and may be of use for sequence-structure threading protein prediction models. Rather than using a traditional definition of SSEs, calculates fragments sufficiently similar to a typical helix and strand from 1bp2 (helix cutoff: 0.4Å, and strand cutoff: 0.8Å). Identifies pairs of similar SSEs (according to some criteria) before using a graph theoretical approach for clique detection (only allowing links between spatially close SSEs). Using this initial alignment, a C_α alignment is realised and extended, subject to maximising the similarity score (as used in *SARF*).

- *STRUCTAL:* Gerstein and Levitt (1996); Levitt and Gerstein (1998)

  Uses dynamic programming to align structures using residue-based scores based on inter-molecular interatomic distances, after superposition, similar to the approach of Subbiah et al. (1993). The superposition, and thus the alignment, is iteratively refined by re-scoring. At each iteration, the alignment is achieved and structures are re-superposed. Allows the use of C^β atoms to represent residues, instead of C^α atoms. The overall similarity score is proportional to the sum of the residue-based scores, subtracting a gap penalty. A z-score, dependent on the number of aligned residues, is used to normalise the score with respect to the distribution of scores for random structure-pairs. These z-scores are found to follow an extreme value distribution.

- *LOCK:* Singh and Brutlag (1997)

  SSE-based method. Represents SSEs as vectors (position and orientation) and scores the positional and orientational agreement between pairs of SSEs between the two structures, using various criteria. Scores include those which are dependent on, and those which are independent of the particular coordinate frame. Uses a dynamic programming algorithm to find the optimal alignment of SSEs (according to their scores) and then superposes according to the achieved initial alignment. Iterative refinement is performed until a satisfactory initial superposition is achieved. Atomic alignment is achieved by identifying closest atoms between the two structures, providing they are sufficiently close, thus identifying a core. Alignment is then filtered in order to insure that only atoms considered to be well-aligned are used in the final alignment. Specifics of the method were updated in *LOCK2* (Shapiro and Brutlag, 2004).

- *DEJAVU:* Kleywegt and Jones (1997)

SSE-based method. Compares relative positions (centre) and orientations (cosine distance) of pairs of SSEs, and also utilises information about the number of residues within, and the length of, SSEs.

- *CE:* Shindyalov and Bourne (1998)

  Fragment-based method which is concerned with the identification of local similarities. A default fragment length of 8 is used. Only considers fragments that are identified as sufficiently similar between the two structures. Employs various heuristics for combining such fragments in order to extend the alignment, in search of an optimal result. The general strategy is to fix one fragment, then iteratively fix additional fragments, providing they agree (do not clash) with the existing alignment, and satisfy some criteria. All aligned (sufficiently similar) fragments may be used as the starting point, potentially resulting in multiple alignments. The longest alignment path is (preliminarily) considered to be the best one. Statistical significance of this (conformation invariant) alignment is represented using a z-score, which considers the distributions of distances (dissimilarity scores) and gaps (number of gaps in the alignment) observed from the comparison of random pairs of non-redundant structures, given that the alignment length is the same as the one achieved. If the z-score is sufficiently high then the structure pair is considered similar, and further refinement is employed by considering global alignment rigidity. Specifically, the RMSDs of the best 20 alignment paths are considered; the one with the lowest RMSD is selected as the best alignment. Various refinement stages are employed in order to optimise the global RMSD, including refinement of fragment positions, and dynamic programming on the distance matrix (with gap penalties for alignment initiation and extension).

- *SAP:* Taylor (1999) Modified version of *SSAP*, which uses various types of information to complicate the scoring function. Also introduces a stochastic element, allowing an iterative approach towards refinement of the optimal alignment.

- *SPASM:* Kleywegt (1999)

  Identifies occurrences of a given structural motif within a database of structures. Such motifs do not necessarily need to consecutive in sequence. Allows the forcing of some residues to be similar in amino acid type, and allows control over allowed residue substitutions. The motif (and structures in the database) are represented using the coordinates of residues C_α atoms and the centre of mass of side chains (or combinations therein), allowing both main chain and side chain (dis)similarities to be considered if desired. All potential hits are superposed with the target, and are considered a match if their RMSD is lower than some threshold.

- *RIGOR:* Kleywegt (1999)

  Counterpart software to *SPASM*, which rather than scanning a database of structures for a given motif, scans a particular protein structure for the presence of a motifs from a database.

- *PrISM:* Yang and Honig (2000)

  Globally rigid SSE-based method. They previously (Yang and Honig, 1999) state that it was designed with structure prediction in mind, in order to identify potential homology models. Consequently, the main aim is to achieve a quantitative dissimilarity score, rather than necessarily describe similarity. Double dynamic programming (as in *SSAP*) is used to align SSEs (scored by positional and orientational information). Iterative global superposition is then used to refine a C_α alignment alignment is maximised, then reduced until RMSD converges. A dissimilarity score is achieved (PSD), which they claim to be applicable for both similar and dissimilar structures. This score utilises achieved information regarding the global RMSD and SSE scores; parameters were chosen by optimising agreement with the classification scheme *SCOP*. They claim that protein fold space is continuous (whilst acknowledging that classification schemes are useful).

- *KENOBI/K2/K2SA:* Szustakowski and Weng (2000)

  Globally rigid SSE-based method, implemented with and without requiring sequence connectivity. Considers an atomic distance matrix, and focuses on regions representing SSEs. SSEs are aligned using a genetic algorithm, which is a stochastic approach, rather than an exhaustive search. A residue correspondence is achieved by iteratively identifying equivalent residues after superposition. The alignment is then extended, subject to rigidity criteria, and scored using the flexible similarity score from Dali. *K2* and *K2SA* are evolutions to this general method. *K2* adopts a vector based approach for representation of SSEs, rather than using an atomic distance matrix. *K2SA* uses a faster simulated annealing algorithm instead of the genetic algorithm.

- *SHEBA:* Jung and Lee (2000)

  Globally rigid method that utilises non-structural information. Construct an aligned residue pair-based score which is a weighted sum of components: sequence homology; similarity of the SSE that the residues belong to; similarity of solvent accessibility and polarity. The overall alignment score is the sum of the residue scores, along with a gap penalty. Alignment is achieved by optimising score using the Needleman-Wunsch dynamic programming algorithm. Aligned residues are subsequently superposed, and alignment is iteratively refined so that superposed residues have RMSD below some threshold (3.5Å). If the achieved superposition is bad, tries giving different weights to different portions of the chain in an attempt to improve the global superposition. Clustering of the z-scores in order to identify clusters in fold space is attempted.

- *Matras:* Kawabata and Nishikawa (2000)

  Uses a Markov transition model in order to score structures at a feature-based level in a way that reflects/allows potential evolutionary changes. Aligns structures using a common ap-

proach: first aligns SSEs, then iteratively refines a deduced residue alignment (using dynamic programming). The alignment is scored in various ways, utilising information about discretised residue distances (using C_β atoms), SSEs pairs, and residue environment. The score is the log-odds of a feature evolving into another, given prior knowledge, calculated using a Markov transition model (similarly to that previously used in the context of amino acid substitution). This score quantifies the chance of any observed differences between aligned features occurring during the natural process of evolution. This also means that it captures information regarding the degree to which observed similarities are significant (e.g. helical conformations are common and do not necessarily imply a well-aligned region). In order to identify probabilities (parameters of the model), the PDB is clustered at 40% sequence identity to define homology, and the number of intra-cluster transition occurrences (of each transition type) is considered. Significance of overall score is provided as a z-score, calculated by performing a 1-on-all comparison of non-redundant structures. Allows multiple structure alignment (based on pair-wise alignments), and identification of repetitions (maximum two).

- *MAMMOTH:* Ortiz et al. (2002)

  Globally rigid method, which uses heptapeptide fragments, compared using the URMSD. Focuses on the comparison of low-resolution or theoretical models with high-resolution models. Performs dynamic programming (with a gap penalty) on an matrix of fragment similarity scores, which are based on the URMSD. The resultant alignment is filtered so that the RMSD of the alignment superposition is less than 4Å. The percentage of aligned residues is used to represent alignment quality. This score is found to follow an extreme-value distribution. By means of log-log-linear regression, using a non-redundant database, the probability of an observed z-score is estimated.

- *PRIDE:* Carugo and Pongor (2002)

  Compares the distribution of intra-molecular interatomic C^α distances. The method is not interested in aligning structures, being more interested in producing a significance-based score for the similarity of structures. Considers the distribution of distances between C^α atoms separated by a given number of residues. Separations of 3–30 residues are considered. For each separation, the distributions are compared using contingency table analysis, resulting in a χ^2 statistic. This allows calculation of the probability of identity of distributions. The average probability out of the 28 distributions is taken as the result. The approach was later modified to use a Kolmogorov-Smirnov test in *PRIDE2* (Gáspári et al., 2005), rather than contingency table analysis.

- *FlexProt:* Shatsky et al. (2002)

  Considers locally similar regions in order to identify rigid substructures shared between two molecules. Allows multiple substructures, and hinges between them, to be identified. Identi-

fies runs of consecutive residues that superpose sufficiently well, resulting in a list of potential alignment segments of at least 12 residues (which may clash). The final alignment is achieved using a graph-based approach in order to identify the optimal path, according to some criteria. The resulting alignment maintains sequence order, and the identified continuous in sequence alignment segments are rigid substructures. The implied hinges between them are also identified.

- *MASS:* Dror et al. (2003)

  Identifies the presence of rigid substructures, according to conservation of SSE orientation and position, between multiple structures. Uses a vector representation of SSEs, and constructs a fingerprint based on geometric properties SSE-pairs, before using geometric hashing to identify larger ensembles of corresponding SSE pairs. This method does not require the maintenance of sequence order; allows detection of non-topological similarities.

- *FATCAT:* Ye and Godzik (2003)

  Uses dynamic programming on structural fragments, allowing gaps and twists, in order to identify a globally flexible alignment. Uses the concept of Aligned Fragment Pairs (AFPs), as in *CE*, which are fragments (length 8) that may be superposed between the two structures, within an RMSD threshold. All AFPs are identified. Those which are compatible (consecutive and within a RMSD threshold) are combined into extended fragments. Dynamic programming is used to find the optimal path of AFPs. The score matrix penalises gaps and AFPs with larger RMSDs, and rewards longer extended fragments. The alignment is allowed to have 'twists' wherever consecutive aligned fragment pairs are incompatible. After achieving the optimal path, twists are added or removed, depending on the resultant effect on the RMSD. The alignment is further iteratively refined.

- *MolCom:* OHearn et al. (2003)

  Uses an octree to partition space, allowing a multi-resolution comparison of protein structures without the explicit identification of feature-based correspondences. Rather, the octree cubes are presumed to correspond, and any similarities between structures that may be described by such a framework are identified. Consequently, this method requires global rigidity, and for the overall size and shape of the molecules to be similar. Also, requires the structures to be pre-superposed – this method assumes that an optimal spatial alignment has already achieved, being more interested in the identification and scoring of similarities. Various properties are considered, such as SSE type, polarity and aromaticity. These properties are split into groups (e.g. aromatic/aliphatic residues), and the proportion of residues of these types is calculated, to be used as descriptors. Different groups are assigned different weights, dependent on the octree resolution level, so that low-resolution features (e.g. SSEs) have higher weight at low octree levels, and high-resolution features (e.g. polarity) have higher weight at low octree

levels. A score is given to each resolution level based on the proportion of cubes considered to be sufficiently similar between the two structures. The overall score is given by the geometric mean of the scores for all considered octree resolutions.

- *FLASH:* Shih and Hwang (2003)

  Globally rigid SSE-based method, which represents SSEs as vectors. Calculates scores for the alignment of both individual SSEs and SSE-pairs. The potential alignment of each SSE from one structure with each SSE from the other structure is scored. Sufficiently high-scoring matches are identified. The highest scoring match is fixed; other SSEs are added to the alignment in order of rank, subject to compatibility with the existing SSEs in the alignment. After identification of the first alignment, this process is then repeated using the highest-scoring unaligned SSE as the seed, and repeated until all high-scoring SSEs are part of an alignment. All identified alignments comprising at least three SSEs are selected for further refinement. Alignments are superposed, and a C^{α} correspondence deduced, refined using dynamic programming, and a z-score reported.

- *CTSS:* Can and Wang (2003)

  Uses differential geometry to describe differences between chains. First, data points (C^{α} atomic coordinates) are smoothed using approximate splines. Shape signatures are calculated corresponding to each residue, which comprise information regarding the backbone curvature, torsion, and secondary structure type. Pairs of signatures are scored according to their agreement. Dynamic programming is used to identify the optimal path through the signature score matrix, using a gap penalty. All rigid local alignments are considered; the one with the best local alignment, according to RMSD criteria, is taken.

- *SSM:* Krissinel and Henrick (2004)

  Globally rigid SSE-based graph theoretical approach, allowing detection of non-topological similarities. SSE Vertices are represented by vectors, along with knowledge of SSE type, and edges between vertices include information such as SSE centres, and various angles between the SSE vectors. All properties of the vertices and edges, which represent information regarding the relative position and orientation of SSE pairs, must be sufficiently similar in order to be considered comparable. Multiple common subgraphs may be identified between the structures. For each subgraph, an alignment of C^{α} atoms is realised by superposing aligned SSEs and identifying closest C^{α} atoms. This is done by first aligning atoms within aligned SSEs, then within non-aligned SSEs, then extending the alignment. Short aligned sections are removed. The alignment is then refined by iteratively superposing and re-scoring the alignment, until the optimal alignment is found according to a similarity score. This score is based on RMSD and number of aligned residues. The common subgraph with the best score is chosen. Significance of scores is estimated in a similar way to *VAST*, considering the

distribution of scores, given knowledge of the number of aligned SSEs and the total number of SSEs, using an non-redundant database. Ultimately, a z-score is reported representing the significance of the similarity score.

- *TOPOFIT:* Ilyin et al. (2004)

  Performs Delaunay tessellation of all C^{α} atoms. This allows the considered features to be tetrahedrons, rather than residues, which can be compared by shape (in various ways). All tetrahedrons are compared. They are then clustered, using the best match as a seed (multiple seeds may be used), subject to satisfying some RMSD criteria. Given an alignment/correspondence of residues/tetrahedrons, the quantities of importance are residue RMSD and the number of aligned tetrahedrons (representing volume of matching regions). z-scores are estimated, using a function of RMSD.

- *MALECON:* Ochagavía and Wodak (2004)

  Globally rigid multiple alignment method. Requires pairwise alignment(s) to be previously calculated and provided as input. Identifies residues that are consistently aligned between all structures. Using this consensus alignment, the median structure is identified as the one that has lowest RMSD to all other structures. All other structures are then superposed into the coordinate frame of the median structure. The geometric centres of corresponding C^{α} atoms are calculated, and all structures are superposed to this set of averaged coordinates.

- *MultiProt:* Shatsky et al. (2004)

  Multiple alignment tool that identifies rigid substructures. This is performed simultaneously for all structures, searching all coordinate frames. Detects all fragments of maximal length, subject to RMSD criteria. These fragments are combined if compatible, and thus can detect non-topoligical similarities.

- *SCALI:* Yuan and Bystroff (2005)

  Globally rigid method. Uses a sequence profile in order to determine scores, which are based on sequence-structure Markov state probabilities. Pairs of fragments are identified, such that all fragments are extended maximally subject to optimising the score. Having similarities to methods such as *SARF* and *CE*, pairs of fragments are combined in a favourable way (using a tree search approach) in order to identify an initial alignment. This alignment is superposed, and iteratively refined in order to optimise the alignment based on RMSD criteria.

- *FAST:* Zhu and Weng (2005)

  Globally rigid method, that uses a graph-based approach to align C^{α} atoms. Local structural environments are compared in order to dramatically reduce the number of potential aligned residue-pairs, thus reducing the number of vertices in the graph. Edges between pairs of vertices are scored using properties of the corresponding residues' local structural environments.

Vertices are then scored by summing the scores of all connected edges. Poor scoring vertices are removed, further reducing the size of the graph. Dynamic programming is then performed on the graph, using the vertex scores, in order to identify the optimal path. The overall score is normalised by the geometric average of the chain lengths; this score is found to follow an extreme value distribution.

- *POSA:* Ye and Godzik (2005)

  Extension to the method of the globally flexible tool *FATCAT* that allows multiple alignment. A residue-based directed acyclic graph representation is adopted, which effectively allows a chains to merge when aligned, and split away when there is a gap, without loss of information. This contrasts with other methods that identify a simple residue correspondence. This requires maintenance of sequence order. Alignments are pairwise combined until all alignments are merged.

- *MAMMOTH-mult:* Lupyan et al. (2005)

  Uses pairwise alignments from *MAMMOTH* in order to achieve a multiple alignment. Alignments are iteratively considered in order of their pairwise *MAMMOTH* scores. First achieves a C^{α} alignment by considering the agreement of unit-vectors. Then reassigns C^{α} correspondences based on global superposition, before refining the core. This process is iterated until all molecules have been aligned.

- *REVOLVER:* Sandelin (2005)

  Performs multiple structural alignment, taking pairwise alignments as input. Constructs a graph, where nodes are residues and edges are equivalences. Identifies consensus agreement between pairwise alignments, if one exists, to identify multiple alignment.

- *TetraDA:* Roach et al. (2005)

  Performs tetrahedralisation of protein structures, as in *TOPOFIT*. However, considers a sequence representation, rather than volumes of tetrahedrons. Connected edges of the tetrahedrons are used to represent spatial connectivity of residues. This connectivity is expressed as a sequence (specifically two sequences, for forward and backward connectivity). The sequence contains information about spatial relationships. Such sequences are compared using a custom similarity measure, and aligned using dynamic programming. Demonstrates an example of the classification of structures along a molecular dynamics trajectory in high-dimensional space.

- *YAKUSA:* Carpentier et al. (2005)

  Attempts to identify common substructures, based on backbone angles. Considers an angle based on the geometry of four consecutive C^{α} atoms. These angles are discretised, so that they can be written as a sequence. Runs of angles (of fixed length) are considered, and

compared with equivalent sized patterns from structures in a database, in order to identify potentially matching structures. These potential matches are extended, if favourable to do so. Hits are ranked in order of similarity score. Various similarity scores are used, including one that measures the global agreement of matches within the structure-pair, enforcing rigidity of identified substructures. A z-score is provided, based on the distribution of scores in the database.

- *SP^3:* Zhou and Zhou (2005)

  Structural fragment-based method, using fragments of length 9. Scores fragments by weighting RMSD and solvent exposure. Use a structural fragment library from non-homologous high-resolution structures. For each fragment in the target structure, the top 25 hits from the library are considered. For each residue, calculate a frequency distribution of amino acid types that occur at that position, as observed in the corresponding (up to 25×9) fragments. A score is created that combines structural, sequence, and secondary structure information, and dynamic programming is used to align the structures.

- *MUSTANG:* Konagurthu et al. (2006)

  Fragment-based approach that allows multiple structure alignments. Identifies superposable fragments (of length 6) within an RMSD threshold (1.75Å). These are combined into longer fragments where possible, subject to maintenance of the the RMSD criteria, and subject to specific heuristics. Scores residues using an elastic similarity function using interatomic C^{α} distances (similar to *Dali*). Residue scores are constructed using intra-fragment information. Uses dynamic programming to align residues, based on the residue scores. Multiple alignment involves merging pairwise alignments along a guide tree, according to their similarity.

- *CBA:* Ebert and Brutlag (2006)

  Multiple structure alignment. Uses *LOCK2* to perform pairwise alignments. Structures are then superposed, based on the alignments. Consistent correspondences are recognised, and multiple alignment is identified.

- *GANGSTA:* Kolbeck et al. (2006); Guerler and Knapp (2008)

  Globally rigid SSE-based method that allows detection of non-topological similarities. Uses a graph-based representation of SSEs, utilising spatial information in the form of supposed residue contacts (sufficiently close residues). Uses a genetic algorithm to align SSEs. Then optimises residue contacts in order to achieve a residue alignment. Constructs a score based on both residue RMSD and SSE agreement. Rather than the genetic algorithm, a combinatorial approach is used in the updated version, *GANGSTA+*.

- *MatAlign:* Aung and Tan (2006)

Considers intramolecular interatomic distance matrices. Compares rows from the two matrices using a dynamic programming algorithm, achieving row-based scores. Dynamic programming is then used on the row-row distance matrix in order to identify the initial alignment. The alignment is then refined in order to optimise an RMSD-based similarity score (from *SARF2*).

- *CURVE:* Zhi et al. (2006)

  This approach considers backbone curvature, avoiding the explicit consideration of torsion. Smoothes C^{α} positions by averaging over a given number of residues. Considers angles between points along the smoothed backbone. The points are separated by a chosen number of residues. Both the smoothing and point separation choices effectively allow alteration of the structural resolution. Alignment is achieved using dynamic programming (with a gap penalty) using a score based on the agreement of angles. Provides a z-score.

- *SSGS:* Wainreb et al. (2006)

  Identifies similar substructures (long fragments) according to structural rigidity, allowing more flexibility in loop regions, and penalising the alignment of residues belonging to different predicted secondary structure types. Focus on structure prediction by fragment assembly. The method is more concerned with comparing fragments of structures, rather than necessarily whole structures. A library of fragments is constructed, in which representatives exhibit conformational stability, assembled from a non-redundant dataset. Fragments are large, at least 15 residues long. A heuristic approach to alignment is taken. Fragments are superposed, giving lower weight to loops. Dynamic programming is performed on intermolecular interatomic distances, giving lower weight to loops. The method of scoring is dependent on predicted secondary structure type transitions.

- *FASE:* Vesterstrøm and Taylor (2006)

  Aligns SSEs, by considering every SSE-pair between the two structures. The best aligned SSE-pairs are selected as alignment seeds. These are combined into one alignment. Residues between SSEs are aligned using dynamic programming. An iterative alignment and superposition method is employed, which involves superposing the structures, and identifying closest residues. Allows detection of non-topological similarities.

- *ComSubstruct:* Morikawa (2006)

  Local method based on differential geometry, encoding fragments in order to achieve structural sequence. Represents the chain as a sequence of (identical) tetrahedron blocks. Tetrahedrons are oriented according to the vector between the C^{α} atoms of residues $i-1$ and $i+1$, thus capturing information regarding curvature. Considers fragments of length 5, thus considers 5-tetrahedron sequences. A form of geometric hashing allows these to be described as a single number, based on the contact pattern. Sequences are then compared for common subsequences.

- *LOVOALIGN:* Martínez et al. (2007)

  Optimises an alignment by optimising a superposition according to a score, rather than calculating the superposition that minimises the RMSD. Specifically, uses the similarity score from *STRUCTAL*. Given a superposition, two methods of alignment are employed, one which uses dynamic programming, and another which identifies spatially closest residues. Transformation is optimised using an iterative Newtonian line search algorithm.

- *Vorolign:* Birzele et al. (2007)

  Constructs similarity score between residues using weighted sum of amino acid and SSE type exchange scores. Performs Voronoi tessellation in order to identify residue spatial contacts. Performs double dynamic programming, first to achieve a score for the similarity of Voronoi cells using the residue-based similarity scores, the second to achieve the overall alignment. Pairwise alignments may be combined into a multiple alignment.

- *CAALIGN:* Oldfield (2007)

  Considers a pseudo-torsion angle based on the geometry of four consecutive residues. These angles are discretized into bins. Consecutive angles are combined into a word. This word is represented using a single hash value, constructed using a power series. Words with identical hash values are used as alignment seeds. Helices are not used as seeds, due to their abundance. For each seed, iterative superposition and alignment refinement is performed, refining the alignment using intermolecular interatomic distances. Pairwise alignments may be combined into multiple alignments.

- *Matchprot:* Bhattacharya et al. (2007)

  Considers structural neighborhoods. This may be in-sequence (same as a structural fragment) or spatial, defined as the k nearest residues. However, in the case of spatial neighborhoods, there is an addition problem of ordering. All neighborhood-pairs are aligned using a graph theoretical approach. Aligned neighborhoods are optimally superposed, and transformations stored. For each transformation, all residues are transformed and a score is achieved. An alignment is then achieved by performing dynamic programming on a sorted distance matrix. This allows detection of non-topological similarities.

- *SARST:* Lo et al. (2007)

  Considers torsion angles. Ramachandran plot is discretised, and clustered into 20 groups. A transition score matrix is constructed, using a database of similar structures. This is then used for sequence alignment.

- *SABERTOOTH:* Teichert et al. (2007)

  Considers a binary matrix of contacts, representing spatial relationships. Principle eigenvector of the contact matrix is considered. The weighted sum of eigenvectors is considered for less

rigid structures. This is used to create a score function for the agreement of vectors. These vectors are then aligned using dynamic programming.

- *PROMALS3D:* Pei et al. (2008)

  Performs multiple sequence alignment. Generates sequence constraints for the corresponding residue alignment. Identifies homologues (according to SCOP) based on a target structure, and generates constraints based on their structural conservation. Scores consistency of sequence and structure-based constraints in order to determine final alignment.

- *TALI:* Miao et al. (2008)

  Aligns structures using torsion angles. Creates a score for the similarity of two pairs of torsion angles by using knowledge of the density of the Ramachandran plot, which is only defined if the two compared residues are of the same amino acid type (so requires a reasonable amount of sequence homology). Performs dynamic programming on a distance matrix of these scores in order to achieve the alignment.

- *Matt:* Menke et al. (2008)

  Fragment-based method, using fragments of length 5–9. The method is flexible, being concerned with local dissimilarities only. However, during the alignment process, consecutive fragments are penalised if their relative transformations vary. Fragments are scored using (negative logarithm of) an RMSD-based p-value, calculated from a database of non-redundant structures. Uses dynamic programming to align the fragments, where the fragment-based score comprises components of RMSD, displacement and relative angles of sequentially aligned fragments. Fragment-pairs must meet certain criteria (cutoffs on RMSD, displacement and relative angles) in order to be aligned. Three rounds of dynamic programming is applied, fixing the alignment after each round, with weaker criteria each iteration. Pairwise alignments may be combined into a multiple alignment.

- *RAPIDO:* Mosca et al. (2008)

  Flexible fragment-based method. The approach is similar to that of *FATCAT*, identifying matching fragments (length 8) and lengthening them subject to RMSD criteria, adjusted by B-factors. However, a graph-based approach allows the fragments to be extended spatially, allowing identification of non-topological similarities. Rigid regions are identified.

- *Fr-TM-align:* Pandit and Skolnick (2008)

  Uses structural fragments to get an initial superpositon, then iteratively refines using intermolecular interatomic distances. Consider only non-overlapping fragments (length 12). Using a fragment-based similarity score, performs dynamic programming, also generating suboptimal alignments, so that there are multiple alignment seeds. Superpose the structures, and

calculate intermolecular interatomic distance matrix. Align residues using dynamic programming, using info about SSE type. Use iterative dynamic programming to maximise a score.

- *TOPS+strings:* Veeramalai and Gilbert (2008)

  Uses the TOPS cartoon representation of protein structures. This graph representation indicates the spatial connectivity of SSEs. The TOPS+ representation includes ligands as nodes, includes loops as SSEs, and also includes extra information such as SSE length. Rather than using a graph theoretical approach to alignment, the chain is represented as a 1D sequence, which captures information from the graph representation, e.g. SSE properties, number of interactions. Dynamic programming is used to align strings. The final score is the sum of dynamic programming scores, normalised with respect to chain lengths.

- *TOPS++FATCAT:* Veeramalai et al. (2008)

  Modification of *FATCAT*, for purposes of speed. Uses information about spatial connectivity, from the TOPS+ graph representation of SSEs, in order to reduce the search space of *FATCAT* when combining aligned fragment pairs.

- *TableauSearch:* Konagurthu et al. (2008)

  Represents protein folding patterns as tableaux. Relative orientations of SSE-pairs are discretised. This encoding allows a form of geometric hashing. Tableaux and subtableux are considered, searching for clusters of similarly oriented SSEs. Tableaux may be aligned using dynamic programming, by assuming a score function.

- *STON:* Eslahchi et al. (2009)

  Represents local structure using 3 angles, representing backbone curvature and torsion. Construct a binary matrix, identifying sufficiently similar local regions between two structures, based on having sufficiently similar angles. Use a greedy approach to finding a path through the matrix. Alignment is then filtered to identify the final alignment subject to global RMSD criteria.

- *SABIC:* Shen et al. (2010)

  Flexible alignment method, by outputting multiple rigid alignments. Constructs a feature-based score at the residue level, using bond lengths, bond angles and torsion angles (i.e. residue-level structural information). Considers the distance matrix of these scores, and finds sufficiently long runs of consecutive residue-pairs with sufficiently small scores. These 'fragments' are used as seed alignments. For each of these seeds, the structures are superposed, an alignment is found using dynamic programming on the matrix of intermolecular interatomic distances, and the superposition and alignment is iteratively refined. Each of the final alignments are then reported.

- *ProBiS:* Konc and Janežič (2010)

  Has particular focus on identifying similar binding sites. Represents protein surfaces as graphs, where nodes are characterised by residues' chemical properties. Common subgraphs are identified. Each identified subgraph is superposed, and further filtering is performed in order to identify structurally rigid cliques.

Bibliography

P.D. Adams, P.V. Afonine, G. Bunkoczi, V.B. Chen, I.W. Davis, N. Echols, J.J. Headd, L.W. Hung, G.J. Kapral, R.W. Grosse-Kunstleve, A.J. McCoy, N.W. Moriarty, R. Oeffner, R.J. Read, D.C. Richardson, J.S. Richardson, T.C. Terwilliger, and P.H. Zwart. Phenix: a comprehensive python-based system for macromolecular structure solution. *Acta Crystallographica Section D: Biological Crystallography*, 66(2):213–221, 2010.

P.V. Afonine, R.W. Grosse-Kunstleve, and P.D. Adams. A robust bulk-solvent correction and anisotropic scaling procedure. *Acta Crystallographica Section D: Biological Crystallography*, 61 (7):850–855, 2005.

N.N. Alexandrov. SARFing the PDB. *Protein Engineering Design and Selection*, 9(9):727, 1996.

N.N. Alexandrov and D. Fischer. Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins: Structure, Function, and Bioinformatics*, 25(3):354–365, 1996.

N.N. Alexandrov and N. Gō. Biological meaning, statistical significance, and classification of local spatial similarities in nonhomologous proteins. *Protein Science*, 3(6):866–875, 1994.

N.N. Alexandrov, K. Takahashi, and N. Go. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. *Journal of Molecular Biology*, 225(1):5–9, 1992.

G.M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, pages 483–485, 1967.

A. Andreeva, D. Howorth, J.M. Chandonia, S.E. Brenner, T.J.P. Hubbard, C. Chothia, and A.G. Murzin. Data growth and its impact on the scop database: new developments. *Nucleic Acids Research*, 36:D419–425, 2008.

K. Arbter, W.E. Snyder, H. Burkhardt, and G. Hirzinger. Application of affine-invariant Fourier descriptors to recognition of 3-D objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):640–647, 1990.

Z. Aung and K.L. Tan. MatAlign: precise protein structure comparison by matrix alignment. *Journal of Bioinformatics and Computational Biology*, 4(6):1197–1216, 2006.

A. Azzalini. A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12(2):171–178, 1985.

K.J. Barker, K. Davis, A. Hoisie, D.J. Kerbyson, M. Lang, S. Pakin, and J.C. Sancho. A performance evaluation of the nehalem quad-core processor for scientific computing. *Parallel Processing Letters*, 18(4):453–469, 2008.

D. Barthel, J.D. Hirst, J. Błażewicz, E.K. Burke, and N. Krasnogor. ProCKSI: a decision support system for protein (structure) comparison, knowledge, similarity and information. *BMC Bioinformatics*, 8(1):416, 2007.

J.L. Bentley. Survey of techniques for fixed radius near neighbor searching. Technical report, Stanford Linear Accelerator Center, Calif.(USA), 1975.

H.M. Berman, T. Battistuz, TN Bhat, W.F. Bluhm, P.E. Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J.D. Westbrook, and C. Zardecki. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

D.P. Bertsekas. *Dynamic Programming and Optimal Control, Vol 1.* Athena Scientific, 2005.

S. Bhattacharya, C. Bhattacharyya, and N.R. Chandra. Comparison of protein structures by growing neighborhood alignments. *BMC Bioinformatics*, 8(1):77, 2007.

F. Birzele, J.E. Gewehr, G. Csaba, and R. Zimmer. Vorolignfast structural alignment using Voronoi contacts. *Bioinformatics*, 23(2):e205, 2007.

E. Blanc, P. Roversi, C. Vonrhein, C. Flensburg, S. M. Lea, and G. Bricogne. Refinement of severely incomplete structures with maximum likelihood in buster-tnt. *Acta Crystallographica Section D: Biological Crystallography*, 60:2210–2221, 2004.

J.U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science(Washington)*, 253(5016):164–164, 1991.

A.T. Brunger, P.D. Adams, G.M. Clore, W.L. DeLano, P. Gros, R.W. Grosse-Kunstleve, J.S. Jiang, J. Kuszewski, M. Nilges, N.S. Pannu, R.J. Read, L.M. Rice, T. Simonson, and G.L. Warren. Crystallography & nmr system: A new software suite for macromolecular structure determination. *Acta Crystallographica Section D: Biological Crystallography*, 54(5):905–921, 1998.

J. Bujnicki, L. Rychlewski, and D. Fischer. Fold-recognition detects an error in the protein data bank. *Bioinformatics*, 18(10):1391–1395, 2002.

T. Can and Y.F. Wang. *CTSS: a robust and efficient method for protein structure alignment based on local geometrical and biological features.* IEEE Computer Society, 2003.

M. Carpentier, S. Brouillet, and J. Pothier. YAKUSA: a fast structural database scanning method. *Proteins: Structure, Function, and Bioinformatics*, 61(1):137–151, 2005.

O. Carugo and S. Pongor. Protein fold similarity estimated by a probabilistic approach based on C [alpha]-C [alpha] distance comparison1. *Journal of Molecular Biology*, 315(4):887–898, 2002.

RB Catell and JR Hurley. The Procrustes program in producing direct rotation to test a hypothesized factor structure. *Behavioural Science*, 7:258–262, 1962.

J.H. Challis. A procedure for determining rigid body transformation parameters. *Journal of Biomechanics*, 28(6):733–737, 1995.

G. Chang. Retraction of "Structure of MsbA from Vibrio cholera: a multidrug resistance ABC transporter homolog in a closed conformation"[J. Mol. Biol.(2003) 330 419-430]. *Journal of Molecular Biology*, 369(2):596, 2007.

Vincent B. Chen, W. Bryan Arendall, III, Jeffrey J. Headd, Daniel A. Keedy, Robert M. Immormino, Gary J. Kapral, Laura W. Murray, Jane S. Richardson, and David C. Richardson. *MolProbity*: all-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D: Biological Crystallography*, 66(1):12–21, 2010.

L.P. Chew, D. Huttenlocher, K. Kedem, and J. Kleinberg. Fast detection of common geometric substructure in proteins. *Journal of Computational Biology*, 6(3-4):313–325, 1999.

P. Clote and R. Backofen. *Computational molecular biology.* John Wiley & Sons Inc, New York, 2000.

A.L. Cuff, I. Sillitoe, T. Lewis, A.B. Clegg, R. Rentzsch, N. Furnham, M. Pellegrini-Calace, D. Jones, J. Thornton, and C.A. Orengo. Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Research*, 39(suppl 1): D420, 2011.

I.W. Davis, A. Leaver-Fay, V.B. Chen, J.N. Block, G.J. Kapral, X. Wang, L.W. Murray, W.B. Arendall, J. Snoeyink, J.S. Richardson, and D.C. Richardson. Molprobity: all-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Research*, 35(suppl 2):W375, 2007.

M.O. Dayhoff and R.M. Schwartz. A model of evolutionary change in proteins. In *Atlas of protein sequence and structure.* Citeseer, 1978.

WL DeLano. MacPyMOL: A PyMOL-based molecular graphics application for MacOS X. DeLano Scientific, Palo Alto, CA, 2007.

J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 2006.

O. Dror, H. Benyamini, R. Nussinov, and H. Wolfson. MASS: multiple structural alignment by secondary structures. *Bioinformatics*, 19(suppl 1):i95, 2003.

I.L. Dryden and K.V. Mardia. *Statistical shape analysis*, volume 4. Wiley New York, 1998.

J. Ebert and D. Brutlag. Development and validation of a consistency based multiple structure alignment algorithm. *Bioinformatics*, 22(9):1080, 2006.

P. Emsley, B. Lohkamp, WG Scott, and K. Cowtan. Features and development of Coot. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):486–501, 2010.

C. Eslahchi, H. Pezeshk, M. Sadeghi, A. Massoud Rahimi, H. Maboudi Afkham, and S. Arab. STON: A novel method for protein three-dimensional structure comparison. *Computers in Biology and Medicine*, 39(2):166–172, 2009.

B.S. Everitt, S. Landau, M. Leese, and D.D. Stahl. *Cluster analysis*. Wiley Series in Probability and Statistics. Wiley, 2011.

A. Falicov and F.E. Cohen. A surface of minimum area metric for the structural comparison of proteins. *Journal of Molecular Biology*, 258(5):871–892, 1996.

A. Folkers and H. Samet. Content-based image retrieval using Fourier descriptors on a logo database. *Pattern Recognition*, 3:30521, 2002.

I. Foster. *Designing and building parallel programs: concepts and tools for parallel software engineering*. Addison-Wesley, 1995.

I. Friedberg, T. Harder, R. Kolodny, E. Sitbon, Z. Li, and A. Godzik. Using an alignment of fragment strings for comparing protein structures. *Bioinformatics*, 23(2):e219, 2007.

J. Garcia-Nafria, L. Burchell, M. Takezawa, N.J. Rzechorzek, M.J. Fogg, and K.S. Wilson. The structure of the genomic bacillus subtilis dutpase: novel features in the phe-lid. *Acta Crystallographica Section D: Biological Crystallography*, 66(9):953–961, 2010.

Z. Gáspári, K. Vlahovicek, and S. Pongor. Efficient recognition of folds in protein 3D structures by the improved PRIDE algorithm. *Bioinformatics*, 21(15):3322, 2005.

M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In *Proc. Int. Conf. Intell. Syst. Mol. Biol*, volume 4, pages 59–67, 1996.

J.F. Gibrat, T. Madej, and S.H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, 6(3):377–385, 1996.

S. Gonin, P. Arnoux, B. Pierru, J. Lavergne, B. Alonso, M. Sabaty, and D. Pignol. Crystal structures of an extracytoplasmic solute receptor from a trap transporter in its open and closed forms reveal a helix-swapped dimer requiring a cation for α-keto acid binding. *BMC Structural Biology*, 7(1): 11, 2007.

J.C. Gower. Procrustes methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4): 503–508, 2010.

J.C. Gower and G.B. Dijksterhuis. *Procrustes problems*. Oxford University Press, USA, 2004.

S. Griep and U. Hobohm. PDBselect 1992–2009 and PDBfilter-select. *Nucleic Acids Research*, 38 (suppl 1):D318, 2010.

A. Guerler and E.W. Knapp. Novel protein folds and their nonsequential structural analogs. *Protein Science*, 17(8):1374–1382, 2008.

John L. Gustafson. Reevaluating Amdahl's Law. *Communications of the ACM*, 31:532–533, 1988.

S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915, 1992.

B. Herberich, G.Q. Cao, P.P. Chakrabarti, J.R. Falsey, L. Pettus, R.M. Rzasa, A.B. Reed, A. Reichelt, K. Sham, M. Thaman, R.P. Wurz, S. Xu, D. Zhang, F. Hsieh, M.R. Lee, R. Syed, V. Li, D. Grosfeld, M.H. Plant, B. Henkle, L. Sherman, S. Middleton, L.M. Wong, and A.S. Tasker. Discovery of highly selective and potent p38 inhibitors based on a phthalazine scaffold. *Journal of Medicinal Chemistry*, 51(20):6271–6279, 2008.

J. Hicklin, C. Moler, P. Webb, R.F. Boisvert, B. Miller, R. Pozo, and K. Remington. Jama: A Java matrix package. *URL: http://math. nist. gov/javanumerics/jama*, 2000.

U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Science*, 3 (3):522–524, 1994.

U. Hobohm, M. Scharf, R. Schneider, and C. Sander. Selection of representative protein data sets. *Protein Science*, 1(3):409–417, 1992.

L. Holm and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16(6): 566, 2000.

L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–123, 1993.

L. Holm and C. Sander. Mapping the protein universe. *Science*, 273(5275):595, 1996.

L. Holm and C. Sander. Dictionary of recurrent domains in protein structures. *Proteins: Structure, Function, and Bioinformatics*, 33(1):88–96, 1998.

L. Holm, C. Ouzounis, C. Sander, G. Tuparev, and G. Vriend. A database of protein structure families with common folding motifs. *Protein Science*, 1(12):1691, 1992.

RWW Hooft, C. Sander, M. Scharf, and G. Vriend. The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. *Computer Applications in the Biosciences: CABIOS*, 12(6):525, 1996.

S. Hovmoller, T. Zhou, and T. Ohlson. Conformations of amino acids in proteins. *Acta Crystallographica Section D: Biological Crystallography*, 58(5):768–776, 2002.

V.A. Ilyin, A. Abyzov, and C.M. Leslin. Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point. *Protein Science*, 13(7): 1865–1874, 2004.

J.W. Johnston, N.P. Coussens, S. Allen, J.C.D. Houtman, K.H. Turner, A. Zaleski, S. Ramaswamy, B.W. Gibson, and M.A. Apicella. Characterization of the n-acetyl-5-neuraminic acid-binding site of the extracytoplasmic solute receptor (siap) of nontypeable haemophilus influenzae strain 2019. *Journal of Biological Chemistry*, 283(2):855, 2008.

R.P. Joosten, J. Salzemann, V. Bloch, H. Stockinger, A.C. Berglund, C. Blanchet, E. Bongcam-Rudloff, C. Combet, A.L. Da Costa, G. Deleage, M. Diarena, R. Fabbretti, G. Fettahi, V. Flegel, A. Gisel, V. Kasam, T. Kervinen, E. Korpelainen, K. Mattila, M. Pagni, M. Reichstadt, V. Breton, I.J. Tickle, and G. Vriend. PDB_REDO: automated re-refinement of X-ray structure models in the PDB. *Journal of Applied Crystallography*, 42(3):376–384, 2009a.

R.P. Joosten, T. Womack, G. Vriend, and G. Bricogne. Re-refinement from deposited X-ray data can deliver improved models for most PDB entries. *Acta Crystallographica Section D: Biological Crystallography*, 65(2):176–185, 2009b.

J. Jung and B. Lee. Protein structure alignment using environmental profiles. *Protein Engineering Design and Selection*, 13(8):535, 2000.

W. Kabsch. A discussion of the solution for the best rotation to relate two vector sets. *Acta Crystallographica Section A: Foundations of Crystallography*, 34:827–828, 1978.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

J.S. Kavanaugh, P.H. Rogers, and A. Arnone. Crystallographic evidence for a new ensemble of ligand-induced allosteric transitions in hemoglobin: The t-to-thigh quaternary transitions. *Biochemistry*, 44(16):6101–6121, 2005.

T. Kawabata and K. Nishikawa. Protein structure comparison using the markov transition model of evolution. *Proteins: Structure, Function, and Bioinformatics*, 41(1):108–122, 2000.

K. Kedem, L.P. Chew, and R. Elber. Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Structure, Function, and Bioinformatics*, 37(4):554–564, 1999.

Y. Kim, A. Joachimiak, E. Evdokimova, A. Savchenko, and A. Edwards. Putative phosphatase Ynic from escherichia coli K12 (doi=10.2210/pdb1te2/pdb), 2004.

G.J. Kleywegt. Recognition of spatial motifs in protein structures. *Journal of Molecular Biology*, 285(4):1887–1897, 1999.

G.J. Kleywegt and T.A. Jones. A super position. *ESF/CCP4 Newsletter*, 31(9):14, 1994.

G.J. Kleywegt and T.A. Jones. Detecting folding motifs and similarities in protein structures. *Methods in Enzymology*, 277:525, 1997.

B. Kolbeck, P. May, T. Schmidt-Goenner, T. Steinke, and E.W. Knapp. Connectivity independent protein-structure alignment: a hierarchical approach. *BMC Bioinformatics*, 7(1):510, 2006.

P. Kolenko, T. Skalova, O. Vanek, A. Stepankova, J. Duskova, J. Hasek, K. Bezouska, and J. Dohnalek. The high-resolution structure of the extracellular domain of human cd69 using a novel polymer. *Acta Crystallographica Section F: Structural Biology and Crystallization Communications*, 65(12):1258–1260, 2009.

R. Kolodny, D. Petrey, and B. Honig. Protein structure comparison: implications for the nature of 'fold space', and structure and function prediction. *Current Opinion in Structural Biology*, 16 (3):393–398, 2006.

A.S. Konagurthu, J.C. Whisstock, P.J. Stuckey, and A.M. Lesk. MUSTANG: a multiple structural alignment algorithm. *Proteins: Structure, Function, and Bioinformatics*, 64(3):559–574, 2006.

A.S. Konagurthu, P.J. Stuckey, and A.M. Lesk. Structural search and retrieval using a tableau representation of protein folding patterns. *Bioinformatics*, 24(5):645, 2008.

J. Konc and D. Janežič. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160, 2010.

E. Krissinel and K. Henrick. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, 60(12):2256–2268, 2004.

J.B. Kuipers. *Quaternions and rotation sequences*. Princeton university press Princeton, NJ, 1999.

P.M. Larochelle, A.P. Murray, and J. Angeles. A distance metric for finite sets of rigid-body displacements via the polar decomposition. *Journal of Mechanical Design*, 129:883, 2007.

U. Lessel and D. Schomburg. Similarities between protein 3-D structures. *Protein Engineering Design and Selection*, 7(10):1175, 1994.

C. Levinthal. Molecular model-building by computer. *Scientific American*, 214:42–52, 1966.

M. Levitt and M. Gerstein. A unified statistical framework for sequence comparison and structure comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 95(11):5913, 1998.

H. Li and R. Hartley. New 3D Fourier descriptors for genus-zero mesh objects. *Computer Vision*, pages 734–743, 2006.

W.C. Lo, P.J. Huang, C.H. Chang, and P.C. Lyu. Protein structural similarity search by Ramachandran codes. *BMC Bioinformatics*, 8(1):307, 2007.

S.C. Lovell, I.W. Davis, W.B. Arendall III, P.I.W. de Bakker, J.M. Word, M.G. Prisant, J.S. Richardson, and D.C. Richardson. Structure validation by Cα geometry: ϕ, ψ and Cβ deviation. *Proteins: Structure, Function, and Bioinformatics*, 50(3):437–450, 2003.

D. Lupyan, A. Leo-Macias, and A.R. Ortiz. A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics*, 21(15):3255, 2005.

T. Madej, J.F. Gibrat, and S.H. Bryant. Threading a database of protein cores. *Proteins: Structure, Function, and Bioinformatics*, 23(3):356–369, 1995.

V.N. Maiorov and G.M. Crippen. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *Journal of Molecular Biology*, 235(2):625–634, 1994.

A.C.R. Martin. http://www.bioinf.org.uk/software/profit/, 1996.

L. Martínez, R. Andreani, and J.M. Martínez. Convergent algorithms for protein structural alignment. *BMC Bioinformatics*, 8(1):306, 2007.

A.D. McLachlan. Rapid comparison of protein structures. *Acta Crystallographica Section A: Foundations of Crystallography*, 38(6):871–873, 1982.

S. McNicholas, E. Potterton, K. S. Wilson, and M. E. M. Noble. Presenting your structures: the *CCP4mg* molecular-graphics software. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):386–394, 2011.

M. Menke, B. Berger, and L. Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS Computational Biology*, 4(e10):88–99, 2008.

X. Miao, P.J. Waddell, and H. Valafar. TALI: Local alignment of protein structures using backbone torsion angles. *Journal of Bioinformatics and Computational Biology*, 6(1):163–182, 2008.

C.C. Milburn, M. Deak, S.M. Kelly, N.C. Price, D.R. Alessi, and D.M.F. Van Aalten. Binding of phosphatidylinositol 3, 4, 5-trisphosphate to the pleckstrin homology domain of protein kinase b induces a conformational change. *Biochemical Journal*, 375(3):531, 2003.

M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2003.

M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

N. Morikawa. Systematic analysis of local flexibility of multiple-structure proteins. *Nucleic Acids Research*, 34:W239–242, 2006.

R. Mosca, B. Brannetti, and T.R. Schneider. Alignment of protein structures in the presence of domain motions. *BMC Bioinformatics*, 9(1):352, 2008.

A. Muller, E. Severi, C. Mulligan, A.G. Watts, D.J. Kelly, K.S. Wilson, A.J. Wilkinson, and G.H. Thomas. Conservation of structure and mechanism in primary and secondary transporters exemplified by SiaP, a sialic acid binding virulence factor from Haemophilus influenzae. *Journal of Biological Chemistry*, 281(31):22212, 2006.

R.M. Murray, Z. Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994.

GN Murshudov, A.A. Vagin, and E.J. Dodson. Refinement of macromolecular structures by the maximum-likelihood method. *Acta Crystallographica Section D: Biological Crystallography*, 53 (3):240–255, 1997.

G.N. Murshudov, P. Skubák, A.A. Lebedev, N.S. Pannu, R.A. Steiner, R.A. Nicholls, M.D. Winn, F. Long, and A.A. Vagin. Refmac5 for the refinement of macromolecular crystal structures. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):355–367, 2011.

A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, 1995.

C.S. Myers and L.R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, 1981.

S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

R. Nicholls, P.J. Porebski, M.M. Klimecka, M. Chruszcz, K. Murzyn, A. Joachimiak, G. Murshudov, and W. Minor. Crystal structure of dethiobiotin synthetase (BioD) from Helicobacter pylori cocrystallized with ATP (doi=10.2210/pdb3mle/pdb), 2010.

J. Nocedal and S.J. Wright. *Numerical optimization*. Springer verlag, 1999.

M.E. Ochagavía and S. Wodak. Progressive combinatorial algorithm for multiple structural alignments: application to distantly related proteins. *Proteins: Structure, Function, and Bioinformatics*, 55(2):436–454, 2004.

A. O'Hagan and T. Leonard. Bayes estimation subject to uncertainty about parameter constraints. *Biometrika*, 63(1):201, 1976.

TJ Oldfield. CAALIGN: a program for pairwise and multiple protein-structure alignment. *Acta Crystallographica Section D: Biological Crystallography*, 63(4):514–525, 2007.

C.A. Orengo, AD Michie, S. Jones, D.T. Jones, MB Swindells, and J.M. Thornton. CATH – a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

A.R. Ortiz, C.E.M. Strauss, and O. Olmea. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, 11(11):2606–2621, 2002.

SD OHearn, AJ Kusalik, and JF Angel. MolCom: a method to compare protein molecules based on 3-D structural and chemical similarity. *Protein Engineering*, 16(3):169, 2003.

S.B. Pandit and J. Skolnick. Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score. *BMC Bioinformatics*, 9(1):531, 2008.

J. Pei, B.H. Kim, and N.V. Grishin. PROMALS3D: a tool for multiple protein sequence and structure alignments. *Nucleic Acids Research*, 1:6, 2008.

M. Petitjean. Interactive maximal common 3d substructure searching with the combined sdm/rms algorithm. *Computers & Chemistry*, 22(6):463–465, 1998.

R. Pozo. Template Numerical Toolkit for linear algebra: High performance programming with C++ and the Standard Template Library. *International Journal of High Performance Computing Applications*, 11(3):251, 1997.

R. Pozo. JAMA/C++ documentation, 2003. `http://math.nist.gov/tnt/jama_doxygen/`.

R Development Core Team. R: a language and environment for statistical computing, 2011. R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org`.

GN Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7:95, 1963.

J. Roach, S. Sharma, M. Kapustina, and C.W. Carter Jr. Structure alignment via Delaunay tetrahedralization. *Proteins: Structure, Function, and Bioinformatics*, 60(1):66–81, 2005.

M.G. Rossmann and P. Argos. Exploring structural homology of proteins. *Journal of Molecular Biology*, 105(1):75–95, 1976.

R.B. Russell and G.J. Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins: Structure, Function, and Bioinformatics*, 14(2):309–323, 1992.

E. Sandelin. Extracting multiple structural alignments from pairwise alignments: a comparison of a rigorous and a heuristic approach. *Bioinformatics*, 21(7):1002, 2005.

D. Sankoff and J.B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Center for the Study of Language and Information, 1983.

MA Saqi, R.B. Russell, and MJ Sternberg. Misleading local sequence alignments: implications for comparative protein modelling. *Protein Engineering*, 11(8):627, 1998.

C. Savino, A.E. Miele, F. Draghi, K.A. Johnson, G. Sciara, M. Brunori, and B. Vallone. Pattern of cavities in globins: the case of human hemoglobin. *Biopolymers*, 91(12):1097–1107, 2009.

G.F. Schröder, A.T. Brünger, and M. Levitt. Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure*, 15 (12):1630–1641, 2007.

G.F. Schröder, M. Levitt, and A.T. Brunger. Super-resolution biomolecular crystallography with low-resolution data. *Nature*, 464(7292):1218–1222, 2010.

Schrödinger, LLC. The PyMOL molecular graphics system, version 1.3r1, August 2010.

H. Schwalbe, S.B. Grimshaw, A. Spencer, M. Buck, J. Boyd, C.M. Dobson, C. Redfield, and L.J. Smith. A refined solution structure of hen lysozyme determined using residual dipolar coupling data. *Protein Science*, 10(4):677–688, 2001.

J. Shapiro and D. Brutlag. FoldMiner: structural motif discovery using an improved superposition algorithm. *Protein Science*, 13(1):278–294, 2004.

M. Shatsky, R. Nussinov, and H.J. Wolfson. Flexible protein alignment and hinge detection. *Proteins: Structure, Function, and Bioinformatics*, 48(2):242–256, 2002.

M. Shatsky, R. Nussinov, and H.J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins Structure Function and Bioinformatics*, 56(1):143–156, 2004.

George M. Sheldrick. A short history of *SHELX*. *Acta Crystallographica Section A: Foundations of Crystallography*, A64(1):112–122, Jan 2008.

Y.F. Shen, B. Li, and Z.P. Liu. Protein structure alignment based on internal coordinates. *Interdisciplinary Sciences: Computational Life Sciences*, 2(4):308–319, 2010.

E.S.C. Shih and M.J. Hwang. Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics*, 19(6):735, 2003.

I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11(9):739, 1998.

K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3):245–254, 1985.

A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 5(2):284–293, 1997.

M.J. Sippl. On distance and similarity in fold space. *Bioinformatics*, 24(6):872, 2008.

Oliver S. Smart, Thomas O. Womack, Claus Flensburg, Peter Keller, Andrew Sharff, , Paciorek Wlodek, Clemens Vonrhein, and Gerard Bricogne. Exploiting structure similarity in refinement: automated ncs and target structure restraints in *buster*. *Acta Crystallographica Section D: Biological Crystallography*, 68, 2012.

TF Smith and MS Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

W.S. Somers, J. Tang, G.D. Shaw, and R.T. Camphausen. Insights into the molecular basis of leukocyte tethering and rolling revealed by structures of P- and E-selectin bound to SLeX and PSGL-1. *Cell*, 103(3):467–479, 2000.

GW Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4): 551–566, 1993.

S. Subbiah, DV Laurents, and M. Levitt. Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Current Biology*, 3(3):141–148, 1993.

G. Sutton, J.M. Grimes, D.I. Stuart, and P. Roy. Bluetongue virus vp4 is an rna-capping assembly line. *Nature Structural & Molecular Biology*, 14(5):449–451, 2007.

J.D. Szustakowski and Z. Weng. Protein structure alignment using a genetic algorithm. *Proteins: Structure, Function, and Bioinformatics*, 38(4):428–440, 2000.

N.N. Taleb. *The bed of Procrustes: philosophical and practical aphorisms.* Random House, 2010.

W R Taylor and C A Orengo. A holistic approach to protein structure alignment. *Protein Engineering*, 2(7):505–19, 1989a.

W.R. Taylor. Protein structure comparison using iterated double dynamic programming. *Protein Science*, 8(3):654–665, 1999.

W.R. Taylor and C.A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208(1): 1–22, 1989b.

W.R. Taylor, T.P. Flores, and C.A. Orengo. Multiple protein structure alignment. *Protein Science*, 3(10):1858–1870, 1994.

F. Teichert, U. Bastolla, and M. Porto. SABERTOOTH: protein structural alignment based on a vectorial structure representation. *BMC Bioinformatics*, 8(1):425, 2007.

D.L. Theobald and D.S. Wuttke. THESEUS: maximum likelihood superpositioning and analysis of macromolecular structures. *Bioinformatics*, 22(17):2171, 2006.

A.A. Vagin and M.N. Isupov. Spherically averaged phased translation function and its application to the search for molecules and fragments in electron-density maps. *Acta Crystallographica Section D: Biological Crystallography*, 57(10):1451–1456, 2001.

M. Veeramalai and D. Gilbert. A novel method for comparing topological models of protein structures enhanced with ligand information. *Bioinformatics*, 24(23):2698, 2008.

M. Veeramalai, Y. Ye, and A. Godzik. TOPS++FATCAT: fast flexible structural alignment using constraints derived from TOPS+ Strings Model. *BMC Bioinformatics*, 9(1):358, 2008.

V. Venkataramanujam and P.M. Larochelle. A coordinate frame useful for rigid-body displacement metrics. *Journal of Mechanisms*, 2:044503, 2010.

J. Vesterstrøm and W.R. Taylor. Flexible secondary structure based protein structure comparison applied to the detection of circular permutation. *Journal of Computational Biology*, 13(1):43–63, 2006.

D.V. Vranić and D. Saupe. 3D shape descriptor based on 3D Fourier transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001), Budapest, Hungary*. Citeseer, 2001.

G. Wainreb, N. Haspel, H.J. Wolfson, and R. Nussinov. A permissive secondary structure-guided superposition tool for clustering of protein fragments toward protein structure prediction via fragment assembly. *Bioinformatics*, 22(11):1343, 2006.

W.A. Wilson. On semi-metric spaces. *American Journal of Mathematics*, 53(2):361–373, 1931.

M.D. Winn, C.C. Ballard, K.D. Cowtan, E.J. Dodson, P. Emsley, P.R. Evans, R.M. Keegan, E.B. Krissinel, A.G.W. Leslie, A. McCoy, S.J. McNicholas, G.N. Murshudov, N.S. Pannu, E.A. Potterton, H.R. Powell, R.J. Read, A. Vagin, and K.S. Wilson. Overview of the *CCP*4 suite and current developments. *Acta Crystallographica Section D: Biological Crystallography*, 67(4):235–242, 2011.

D.S. Wishart, R.F. Boyko, L. Willard, F.M. Richards, and B.D. Sykes. SEQSEE: a comprehensive program suite for protein sequence analysis. *Bioinformatics*, 10(2):121, 1994.

A. Wlodawer, J. Lubkowski, and W. Minor. Is too 'creative' language acceptable in crystallography? *Acta Crystallographica Section D: Biological Crystallography*, 66(9):1041–1042, 2010.

M.J. Wood and J.D. Hirst. Protein secondary structure prediction with dihedral angles. *Proteins: Structure, Function, and Bioinformatics*, 59(3):476–481, 2005.

A.S. Yang and B. Honig. Sequence to structure alignment in comparative modeling using PrISM. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):66–72, 1999.

A.S. Yang and B. Honig. An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance. *Journal of Molecular Biology*, 301(3):665–678, 2000.

Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19(suppl 2):ii246, 2003.

Y. Ye and A. Godzik. Multiple flexible structure alignment using partial order graphs. *Bioinformatics*, 21(10):2362, 2005.

X. Yuan and C. Bystroff. Non-sequential structure-based alignments reveal topology-independent core packing arrangements in proteins. *Bioinformatics*, 21(7):1010, 2005.

G. Zhang, J. Dai, L. Wang, D. Dunaway-Mariano, L.W. Tremblay, and K.N. Allen. Catalytic cycling in β-phosphoglucomutase: A kinetic and structural analysis. *Biochemistry*, 44(27):9404–9416, 2005.

D. Zhi, S. Krishna, H. Cao, P. Pevzner, and A. Godzik. Representing and comparing protein structures as paths in three-dimensional space. *BMC Bioinformatics*, 7(1):460, 2006.

H. Zhou and Y. Zhou. Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins*, 58(2):321, 2005.

J. Zhu and Z. Weng. FAST: a novel protein structure alignment algorithm. *Proteins: Structure, Function, and Bioinformatics*, 58(3):618–627, 2005.