Hybrid Methods for Protein Loop Modelling

Claire Marks
Wolfson College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2016

Acknowledgements

Well, where to start? Massive thanks must first go to Charlotte - I can't overestimate how much I appreciate the help and guidance she has provided over the last four years. Thanks for teaching me to be an approximation of a real scientist - oh, and for helping me learn to do cryptic crosswords!

I have to also thank all my industrial collaborators: Stefan and Guy at Roche; and Jiye, Seb and Terry at UCB. I appreciate all the time they have given up to talk to me and the many ideas they have offered for my project.

The last four years wouldn't have been the same without some brilliant friends. To my lovely housemates Anna, Adam and the recently-moved-back-to-Finland Hanna - thanks for keeping me sane(ish!) and for being excellent TV-watching buddies. Thanks to all the members of OPIG past and present, especially Rey, Hannah (old), James, Nick, Jin, Eleanor, Hannah (new)... I could go on, you're all awesome!

Last but by no means least, my wonderful family. As much as they like to pick on me for my 'posh' accent when I go back to visit, I know they're always there for me and there's no way I'd be where I am today without them. The clan is too big to name everyone, but special mentions have to go to Anthony, Heather and Michael.

Then there's my three favourite living things in the entire world. Alfie, for providing puppy dog cuddles when I need them most; Mum, for being the voice of reason (yes Mum, I'll have a break now) and just the best mum in the world (even if you refuse to help me protect myself against my sister); and Amelia, the most infuriating yet the most fun little sister ever. Love you! xxx

Abstract

Loops are often vital for protein function, and therefore accurate prediction of their structures is highly desirable. A particularly important example is the H3 loop of antibodies. Antibodies are proteins of the immune system that are able to bind to a huge variety of different substances, in order to initiate their removal from the body. The binding characteristics of an antibody are mainly determined by the six loops, or complementarity determining regions, that make up their binding site. The most important of these is the H3 loop - however, since it is extremely variable in structure, the accuracy of H3 structure prediction is often poor.

Current loop modelling algorithms can mostly be divided into two categories: knowledge-based, where databases of fragments are searched to find suitable conformations; and *ab initio*, where conformations are generated computationally. In this thesis, we test the ability of such methods to predict H3 structures using one of each: the previously published, knowledge-based algorithm FREAD; and our own new *ab initio* method MECHANO.

Existing knowledge-based methods only use fragments that are the same length as the target, even though loops of slightly different lengths may adopt similar conformations. We describe the development of a novel algorithm, Sphinx, which combines *ab initio* techniques with the potential extra structural information contained within loops of a different length to improve structure prediction.

Finally we look at protein flexibility, by identifying loops for which there are multiple structures deposited in the PDB. We examine the outcome of performing structure prediction on loops with varying amounts of flexibility, and investigate differences between those loops that show a high degree of structural variability and those that do not.

Declaration

I declare that no parts of this thesis or its research herein have been reproduced or accepted for another award or degree or diploma at any other university or learning institution. This thesis contains no other person's work except where stated in the text.

Claire Marks

6th October 2016

Contents

List of Figures

List of Tables

Pre-Chapter Image: An antibody using the six loops of its binding site to bind to a protein found in the HIV virus.

1 Introduction

Contents

Proteins are the macromolecular machines of the cell. They are responsible for most of the biological processes required for an organism to survive. Their roles are many and varied. As enzymes, they act as catalysts, speeding up chemical reactions — for example amylase, which breaks down carbohydrates. They form structurally important elements, for example actin, which maintains cellular shape by forming the cytoskeleton and has a role in muscle contraction. Essential processes involving DNA, like transcription and replication, are controlled by proteins such as polymerases,

helicases and transcription factors. They are also used to transport objects to different locations — the protein dynein, for example, is able to 'carry' various types of cargo along the filaments of the cytoskeleton to where they are required.

The function of a protein is dependent on its three-dimensional structure. Porins, for example, allow the transport of substances in and out of the cell, and are able to do this because they form cylindrical tubes that make channels crossing the membrane.

Proteins are polymers; long chains made up of smaller molecules linked together. These molecules, called amino acids, interact with each other, causing most proteins to fold up into a specific three-dimensional shape determined by the ordering of the amino acids in the chain. Since there are a vast number of possible sequences, many differently shaped proteins can be made.

In this chapter, we discuss the fundamentals of protein structure and its prediction; the specific problem of loop prediction; and introduce antibodies, an important family of proteins that are involved in the immune response, and whose biological functions are highly dependent on their loop regions. Finally, we give an overview of the work presented in this thesis.

1.1 Protein Structure

Protein structure is commonly divided into four levels: primary, secondary, tertiary and quaternary. A pictorial representation of this is shown in Figure 1.1.

1.1.1 Amino Acids

Protein chains are polymers, made up of smaller molecules (monomers) called amino acids. Each amino acid has the same basic chemical structure (Figure 1.2) — a central carbon atom (the alpha carbon, or Cα), with four bonded groups. Three of these groups are always the same: a carboxylic acid group, an amine group, and a hydrogen atom. The fourth group, known as the sidechain or 'R' group, varies between amino acid types.

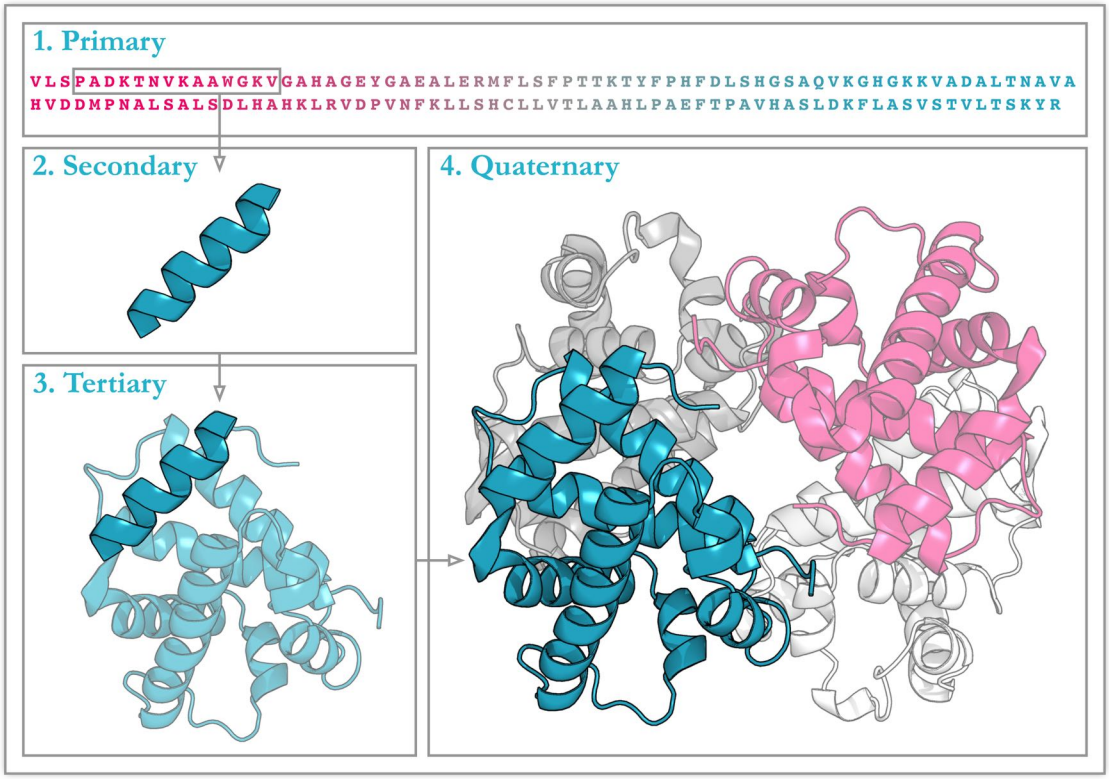

Figure 1.1: The four levels of protein structure, using haemoglobin (PDB code 1BUW) as an example case. The primary structure of a protein is its amino acid sequence. The secondary structure refers to local regions of regular conformation. The way the whole protein chain folds up into a three-dimensional structure is known as tertiary structure. Quaternary structure is only applicable to proteins formed of multiple chains, and refers to how these chains interact with each other.

The alpha carbon is a chiral centre, giving rise to two possible isomers (or enantiomers) which differ in the way the four groups are arranged in space. The two forms are labelled L (levorotatory) and D (dextrorotatory) depending upon how they rotate plane polarised light; the vast majority of naturally occurring amino acids in proteins are of the L form.

There are twenty naturally occurring amino acids, each with their own sidechain (shown in Figure 1.3). Each has both a three-letter and one-letter code; for example glycine is commonly abbreviated to GLY or just G. Sidechains are made up of hydrogen, carbon, nitrogen and sulphur, and range from simple groups, such as the hydrogen atom of glycine, to larger ones such as tryptophan with its fused aromatic rings. The chemical nature of the sidechain determines how the amino acids (or residues) interact, and therefore how they fold up to form a three-dimensional structure when part of a chain. Some amino acids are hydrophobic, such as leucine and valine. Conversely, some

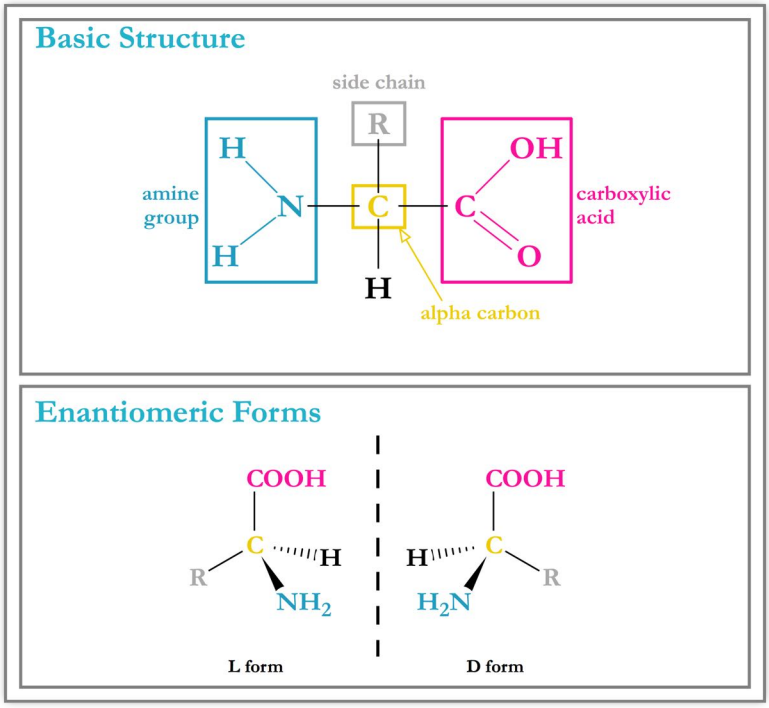

Figure 1.2: The basic structure of an amino acid (top) features a central carbon (the alpha carbon, or Cα) joined to four groups: a hydrogen atom, an amine group, a carboxylic acid group and an R group or sidechain, which varies between the different types. The alpha carbon is a chiral centre, meaning there are two possible isomers (enantiomers) that are mirror images of each other. The majority of amino acids in nature are of the L form.

are hydrophilic — like lysine and arginine, whose charged sidechains make interactions with water favourable. Soluble proteins, therefore, tend to have hydrophobic sidechains buried in their core, and hydrophilic ones on their surface.

There are two amino acids that are slightly different in structure to the rest. The sidechain of glycine is just a hydrogen atom, making it achiral and allowing more backbone conformational flexibility than is possible for other residues. Proline has a cyclic structure, formed by the bonding of the sidechain to the backbone amido nitrogen atom. In contrast to glycine, this makes proline much more rigid than other amino acids; it is commonly found in regions where the direction of the protein chain changes ('turns').

Amino acids can be divided into groups depending on their chemical properties (Brunning, 2014). Aliphatic residues have sidechains that only contain carbon and hydrogen (alanine, glycine, isoleucine, leucine, proline and valine). Aromatic sidechains have stable, planar, aromatic ring systems (phenylalanine, tryptophan and tyrosine).

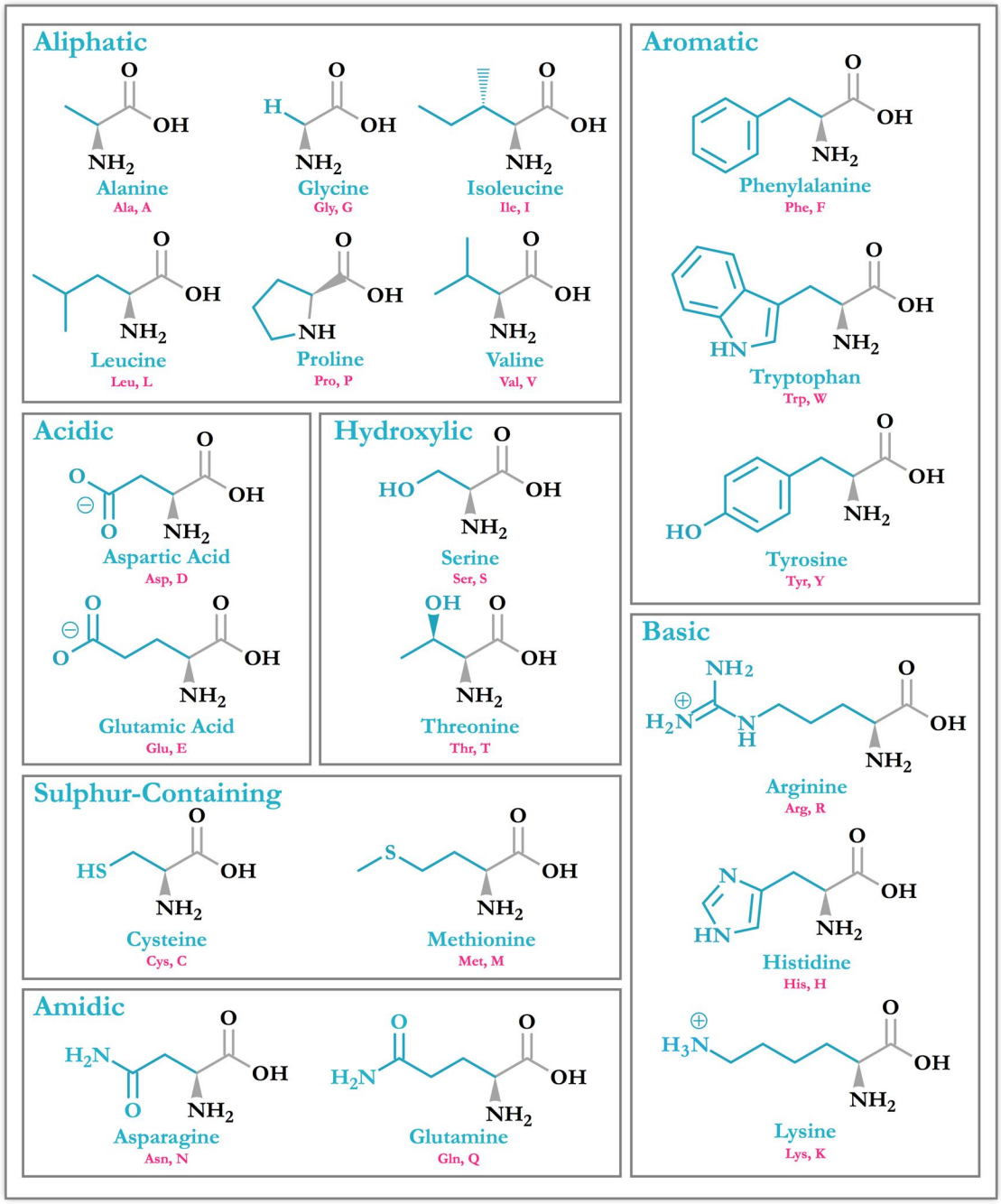

Figure 1.3: The different naturally-occurring R groups give rise to the twenty amino acids that form proteins. These are displayed here, grouped by their chemical properties. Three- and one-letter codes are given beneath each structure in pink.

Some sidechains are acidic, with carboxylic groups that can donate their hydrogen ions making them negatively charged (aspartic acid and glutamic acid), while others are basic, with nitrogen atoms that can accept protons to give a positive charge (arginine, histidine, and lysine). There are two amino acids that contain sulphur (cysteine and

methionine). Other classes are characterised by the presence of particular functional groups: some have hydroxylic (OH-containing) sidechains (serine and threonine), and others have amidic sidechains (asparagine and glutamine). The amino acids can also be grouped according to the polarity of their sidechains (and hence their hydrophobicity). The polar amino acids are arginine, asparagine, aspartic acid, glutamic acid, glutamine, histidine, lysine, serine, threonine and tyrosine (these amino acids therefore tend to be hydrophilic); alanine, cysteine, glycine, isoleucine, leucine, methionine, phenylalanine, proline, tryptophan and valine are non-polar (and therefore tend to be hydrophobic).

1.1.2 Primary Structure

The primary structure of a protein is the sequence of amino acids that link together to make it. The genome contains sections of DNA that code for proteins (genes). Each amino acid is encoded by a set of three bases, of which there are four types: adenine, guanine, thymine, and cytosine. Through DNA transcription and translation, this code is turned into a chain of amino acids, covalently bonded together by peptide bonds. These bonds are formed by a condensation reaction, catalysed by the peptidyl transferase enzyme — a hydrogen atom from the amine group of one amino acid and the -OH from the carboxylic group of another is lost as water, and a covalent bond between the two residues is formed (Figure 1.4A).

The structure of a protein can be described in terms of bond lengths, angles and dihedral angles. Bond lengths and angles are relatively constant; ideal values for each bond type have been determined from experimental data (Engh and Huber, 1991). Three dihedral (or torsion) angles are defined for the protein backbone: ϕ (phi), ψ (psi) and ω (omega). These are shown in Figure 1.4B. Each dihedral angle is defined by four atoms and describes the rotation about the central bond: for residue i, ϕ, ψ and ω depend on the following atoms:

- ϕ — C_{i-1}, N_i, $C\alpha_i$, C_i
- ψ — N_i, $C\alpha_i$, C_i, N_{i+1}
- ω — $C\alpha_i$, C_i, N_{i+1}, $C\alpha_{i+1}$

Peptide bonds are planar, and therefore the ω angle is restricted to roughly 180° (*trans* conformation), or occasionally 0° (*cis* conformation). The other two angles, ϕ and ψ, are restricted to particular combinations of values that do not cause clashes between sidechains. Ramachandran plots are used to visualise this — an example is shown in Figure 1.4C, displaying the generally allowed combinations of angles.

Sequences of amino acids are conventionally written from the N-terminus to the C-terminus, in the order that they are translated by the ribosome. The length of a whole protein sequence can vary widely, from just a few amino acids for short peptides to many thousands, but on average is 335 (EMBL-EBI, 2016).

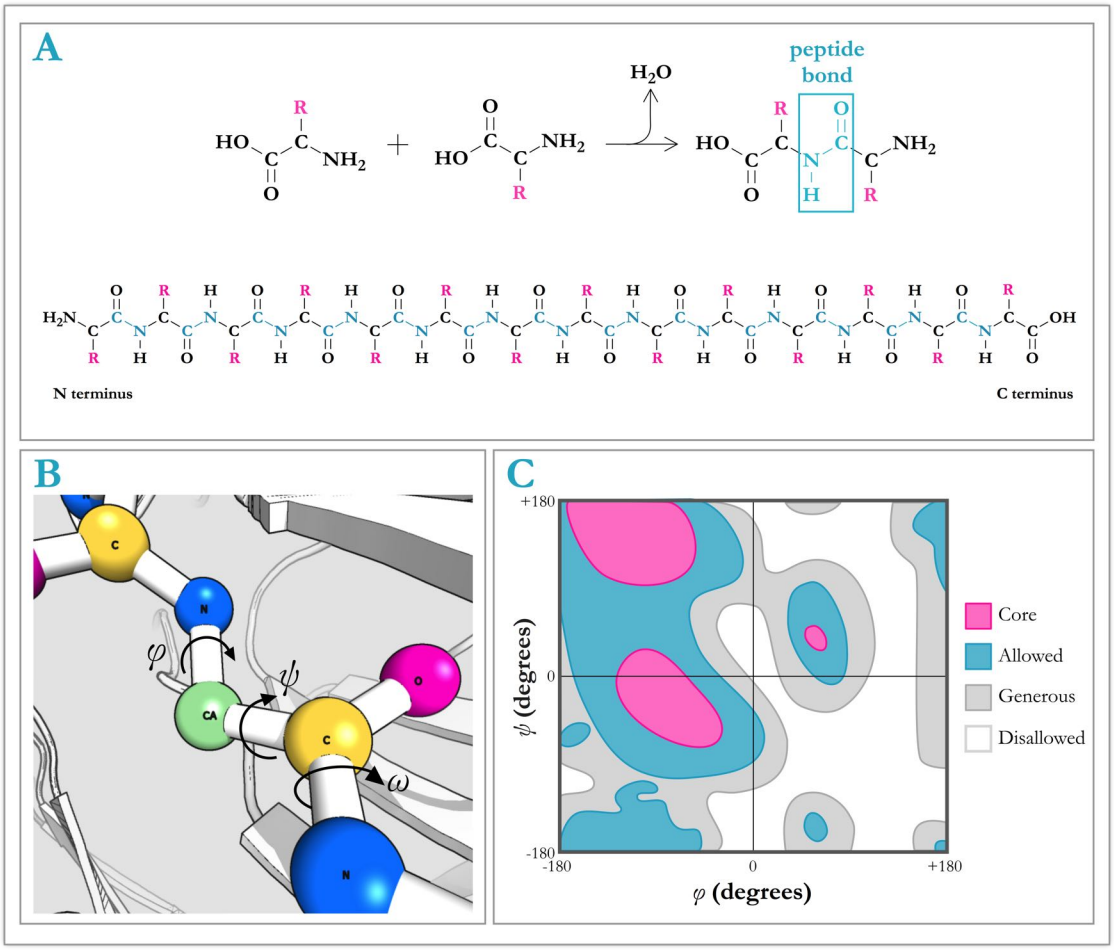

Figure 1.4: Peptide bonds and dihedral angles. A — amino acids join together to form long chains, linked through the formation of peptide bonds via a condensation reaction. B — The backbone dihedral angles, ϕ, ψ and ω. C — A Ramachandran plot showing which pairs of ϕ/ψ angles form the core, allowed, generously allowed and disallowed regions. Only certain pairs of ϕ and ψ angles are permissible, since some combinations introduce clashes between sidechains. Data taken from PROCHECK (Laskowski *et al.*, 1993).

1.1.3 Secondary Structure

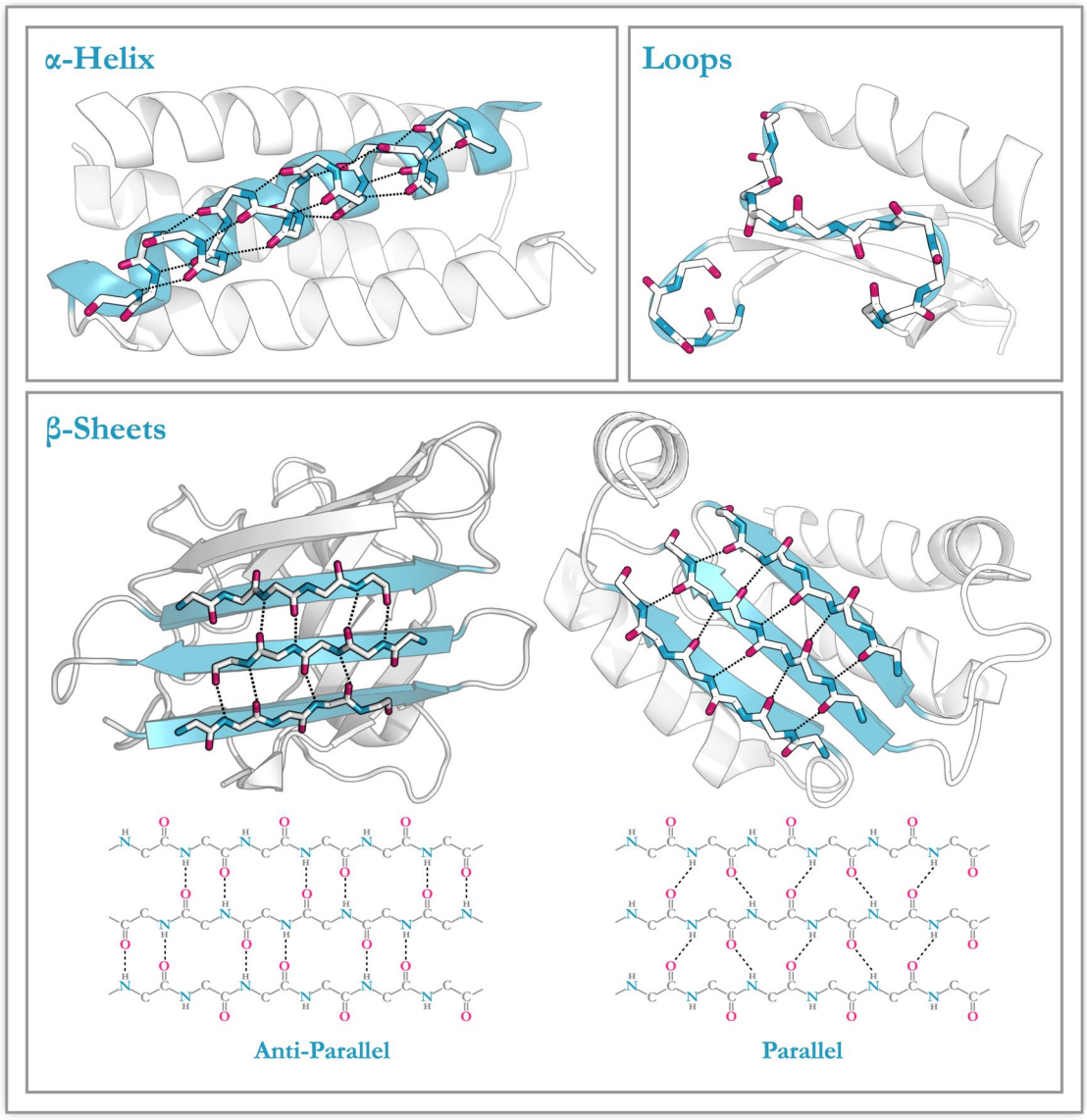

Figure 1.5: Secondary structure motifs: α helices, β-sheets and loops (used in this thesis to refer to everything else).

The hydrogen connected to the nitrogen of the backbone peptide bond can interact with the backbone carbonyl oxygen of another residue — if several of these hydrogen bonds occur in a consistent pattern, the protein backbone folds up into local segments of regular structure. This is referred to as secondary structure. The two main types of structural motif formed in this way are α-helices and β-sheets (Figure 1.5).

Helices are formed when, for a section of the protein sequence, the N-H of one residue's backbone forms a hydrogen bond with the carbonyl group a few residues

ahead, producing a coiled structure with sidechains on the outside. The most common type of helix (the α-helix) is created when the latter residue is four ahead in the sequence, leading to a coil with 3.6 residues per turn. Other helix types exist with a gap of three or five residues (for the 3_{10} and π helices, respectively).

β-sheets consist of multiple extended sections of the protein backbone (β-strands) that are located side-by-side and held together by hydrogen bonds. Depending on the direction of the neighbouring strands, they are classified as either parallel or anti-parallel. Parallel β-sheets are formed by strands with their N- to C-terminus directions aligned. When arranged anti-parallel, adjacent strands are opposite to one another in direction. Anti-parallel β-sheets are the most common of the two (see Figure 1.5). Strands can be found consecutively in a sequence, or may come together from different part of the protein chain, to form a sheet.

Regions of the protein that do not display regular hydrogen bond patterns are referred to as loops. Loops play an important role in protein structures, connecting the different secondary structure elements together, and are often crucial to biological function. Since they lack the repetitive bonding patterns, they are often more conformationally flexible than other regions. While α-helices and β-sheets normally form the structural core of a protein, loops are most often found on the protein surface and as such are normally rich in hydrophilic amino acids.

1.1.4 Tertiary Structure

Tertiary structure refers to how the whole protein chain folds up into its three-dimensional shape. Examples of the interactions that form during this process are:

- Hydrogen bonds — these occur when a hydrogen atom (the donor), which is covalently bonded to an electronegative atom such as nitrogen and hence is slightly positively charged, comes in close proximity to another electronegative atom (the acceptor).

- Electrostatic interactions (salt bridges) — occur between amino acid sidechains with opposite charges, such as lysine (positive) and aspartic acid (negative).

- Disulphide bridges — when two cysteine residues come into contact, they can form a covalent S-S bond.
- Hydrophobic interactions — non-polar groups in soluble proteins tend to cluster together in the protein core, away from the surface and hence the polar solvent.
- van der Waals interactions — occur when a charged or polar group induces a dipole in a non-polar group.

When combined, these interactions enable most proteins to adopt a stable, specific structure.

Sections of proteins that form distinct structural (or functional) units are known as domains. They are considered to be the building blocks of a protein, that can fold and exist independently of the rest. The same domain, with its characteristic structure, can appear in many different proteins alongside others in different arrangements.

1.1.5 Quaternary Structure

For protein complexes, that are made up of multiple chains, the way they interact with one another to form a whole is called the quaternary structure. An example is the protein haemoglobin, which is responsible for transporting oxygen in the blood. Haemoglobin is a complex made of four protein chains in an approximately tetrahedral arrangement (Figure 1.1), with two types of subunit, each of which has a unique sequence. The four chains are held together with electrostatic and hydrophobic interactions, along with hydrogen bonding.

1.1.6 Experimental Structure Determination

Knowledge of a protein's structure can give a crucial insight into how it achieves its function (for example, enabling the identification of binding sites, ligand interactions, and catalytic mechanisms). Structures are important for many applications, such as guiding drug discovery, studying protein dynamics and interactions, and predicting binding affinities. Solved structures are stored in the Protein Data Bank, or PDB (Berman *et al.*, 2000); at present it contains over 120,000 entries, with nearly 39,000 unique

sequences. There are several experimental methods that are used to determine protein structures — the most common are X-ray crystallography and NMR spectroscopy.

X-ray Crystallography

Structures solved using X-ray crystallography make up the majority of the PDB (89%, The Protein Data Bank 2016). This approach uses crystals of the protein, in which the molecules are ordered in a repeating array, and all have the same orientation. When an X-ray beam is fired at the crystal, some of the rays are scattered by electrons within it, producing a diffraction pattern that is dependent on the positions of the protein atoms. The crystal is rotated, with a different diffraction pattern collected at various orientations. By analysing the position and intensity of all the spots in the diffraction patterns, a map of the electron density, and hence the protein structure, can be elucidated (Ilari and Savino, 2008).

X-ray crystallography can produce high quality structures, providing enough data to make it clear where each atom is located (Wlodawer *et al.*, 2008). However, it has its drawbacks. In particular, it can be very challenging to form high-quality crystals for some proteins; despite a great deal of research, it remains primarily a trial-and-error process (Ilari and Savino, 2008; Krauss *et al.*, 2013). As proteins are usually found in solution, the structure achieved using this approach may not reflect its true nature — for example, contacts may form between proteins in the crystal that induce non-natural structures to be adopted (Pokkuluri *et al.*, 2012). It is also not useful for determining the structure of flexible regions; the atoms must be in approximately the same place in every molecule of the crystal to achieve a good diffraction pattern (Wlodawer *et al.*, 2008). On occasion, it may be possible to extract multiple conformations from X-ray crystallography (for instance if the sidechain of a residue is found in two conformations in the different members of the crystal), but only if they can be distinguished in the electron density map. For truly flexible (disordered) regions, the electron density map becomes too 'blurred' to obtain any meaningful structures.

Structures attained using X-ray crystallography have a series of values associated with them which indicate their quality (Wlodawer *et al.*, 2008). Resolution, reported in

Ångströms, is a measure of the size of the smallest discernible feature, hence giving an idea of how accurately the atoms are positioned. Atoms in the protein structure that are closer to each other than the resolution value are not separated in the electron density map. A low value therefore means there is greater confidence in the structure (high resolution). Structures with a resolution of around 1 Å come from high-quality crystals, and every atom (including hydrogens) can be positioned accurately. For low-quality structures (above 3 Å resolution), only the conformation of the backbone may be correct.

The amount that the final, deposited protein structure matches the electron density achieved by experiment is quantified in the R value. After a structure is obtained based on the electron density, that structure is used to generate a simulated diffraction pattern. This is compared to the actual diffraction pattern, and their similarity is given as the R value, with a low value indicating that they match well. There is a problem with this, however — since this process is normally used to refine the structure and improve the R value, bias is introduced. A better measure is given by the R-free value. Refinement is performed, only using 90% of the data. After the refinement procedure is complete, the R value calculation described above is carried out on the remaining 10% to give the R-free value (Brünger, 1997).

In addition to resolution, R, and R-free values, each atom in a structure determined by X-ray crystallography has a value associated with it called a B-factor, or temperature factor. This value, given in units of $Å^2$, is calculated using the mean squared displacement of the atom, and hence describes how spread out the electron density of that particular atom is — a high value means the atom is disordered and may not be accurately placed, i.e. it is found in different positions in each copy of the protein found in the crystal (Wlodawer *et al.*, 2008).

NMR Spectroscopy

Nuclear magnetic resonance (NMR) spectroscopy accounts for approximately 9% of the structures in the PDB. By subjecting protein chains in solution to a magnetic field and radio frequency radiation, information can be extracted about the chemical environments of the atoms in the molecule. Depending on the type of NMR, the atoms

investigated include hydrogen, carbon and nitrogen. From the NMR data, a set of constraints can be obtained that impose limits on the relative locations of particular atoms. This in turn leads to a set of possible protein structures.

Unlike X-ray crystallography, crystals of the protein are not required for NMR spectroscopy, and it can therefore be used to study flexible proteins. However, NMR produces lower quality structures than X-ray crystallography (Sikic *et al.*, 2010), cannot be used for very large proteins (Yu, 1999), and depends on the number of constraints that can be determined — fewer constraints leads to a larger set of possible structures.

1.1.7 Protein Structure Prediction

Due to the difficulties and time-consuming nature of protein structure determination, there is a large and increasing gap between the number of known sequences and the number of experimentally solved structures (Lee *et al.*, 2009; Moult *et al.*, 2014). This means that researchers often try to predict the structure of a protein from its sequence instead. The accuracy of protein structure prediction is monitored by an assessment exercise held every two years called CASP (Moult *et al.*, 1995), where researchers around the world are asked to model proteins not yet experimentally solved from their sequence. Algorithms for protein structure prediction can be classified into two groups: template-based methods, and template-free methods.

Template-Based Modelling

Template-based (or homology) modelling methods use proteins whose structures have already been determined to predict the conformation of a target protein. As organisms evolve, mutations occur in the genes that encode proteins. However, the function of a protein is likely to stay the same, and hence structure will be conserved. Evolutionarily related (or homologous) proteins, therefore, may share the same structure and function, but have different sequences. Proteins that have sequences similar to a target sequence are therefore likely to have similar structures to that target, and can be used as templates with which to produce models (Fiser, 2009). Possible template proteins are identified

by aligning residues in the sequence of known structure to their equivalents in the target sequence, and looking at their similarity.

The performance of template-based modelling is primarily dependent on the accuracy of the alignment, as well as the selection and quality of the template structure (Martin *et al.*, 1997). Aligning multiple sequences instead of just one can reduce errors associated with the former issue (Fiser, 2009). However, there may still be gaps in the alignment, associated with less well conserved regions (in particular the loops). These segments must be modelled separately. Errors can also occur in modelling the sidechains.

If the wrong template is chosen, i.e. one with a dissimilar structure to the target, it is impossible using a homology approach to produce an accurate model. The choice of template is therefore a key decision. Some protein modelling algorithms are able to use multiple templates, averaging the structures in an attempt to reduce errors (Larsson *et al.*, 2008). High sequence identity (a large number of matching amino acid types) normally means that two sequences have nearly identical structures; it is generally thought that proteins with over 40% sequence identity will be conformationally very similar (Rost, 1999). Below this value, however, structural similarity becomes less clear, choice of template becomes more difficult, and another approach may be necessary. If two templates are available with similar sequence identities, the one with higher resolution is generally chosen as it is likely to make the resulting model more accurate.

Template-Free Modelling

Template-free modelling, also called *de novo* or *ab initio* modelling, refers to structure prediction without the use of previously determined structures (Lee *et al.*, 2009). This approach is used when no known structure with a similar sequence can be found. Models are generated by searching the possible conformational space, guided by techniques such as assembling short peptide fragments, building the protein starting from the N-terminus as it would be *in vivo*, or the prediction of contacts within the protein, as well as monitoring the energy of the model using an energy function. Currently, this approach is challenging, due to the difficulty in searching enough of the conformational

space, the inaccuracy of current energy functions, and the computational power required. Prediction quality is therefore generally poor (Tai *et al.*, 2014).

1.2 Protein Loop Modelling

Many residues in a given protein will form regions of regular structure, in α-helices and β-sheets. The segments of the protein that join these secondary structure elements together, that do not have easily observable regular patterns in their structure, are referred to as loops. This does not mean, though, that loops are only minor components of a protein structure – on average, half of the residues in a protein are found in loops (Regad *et al.*, 2010), and they are typically found on the surface of the protein, which is largely responsible for its shape, dynamics and physiochemical properties (Fiser and Sali, 2003).

Loops are often vitally important to a protein's function. For example, they are known to play a role in protein-protein interactions (Gavenonis *et al.*, 2015), signalling cascades (Bernstein *et al.*, 2004), ligand binding (Saraste *et al.*, 1990), DNA binding (Tainer *et al.*, 1995), and enzyme catalysis (Pineda *et al.*, 2007). Knowledge of loop structures is therefore extremely useful, enabling predictions to be made about a protein's behaviour.

More insertions, deletions and substitutions occur in loop regions than in the more conserved α-helices and β-sheets (Panchenko and Madej, 2005). This means that, for a homologous set of proteins, the loop regions are the parts that vary the most between structures, in both length and sequence. While this often makes the protein's function possible, it leads to unaligned regions in a sequence alignment; standard homology modelling techniques can therefore not be used. This makes prediction of their structure difficult – as a result, it is frequently the loop regions that are the least accurate parts of a protein model (Baker and Sali, 2001; Gront *et al.*, 2012). The aim of the work presented in this thesis is to improve the accuracy of loop structure prediction. The following sections describe current algorithms and their accuracy.

1.2.1 Steps in Loop Modelling

The starting point of the loop modelling problem is a series of missing residues in a protein structure, where the sequence of the missing segment is known but the three-dimensional structure of those residues is not. The protein structure used as input may be an experimentally-determined one, or a model. Predicting the structure of the loop requires three main steps: model (or decoy) generation, filtering, and ranking (Figure 1.6). In a similar way to the prediction of whole protein structures, where methods can be template-based or template-free, loop modelling algorithms can be divided into two categories depending on whether known structures are used in the decoy generation step. These categories are known as knowledge-based and *ab initio*.

Decoy Generation

When predicting a loop structure, the first step is to generate a set of candidate conformations, or decoys, that connect the residues on either side of the gap in the protein structure. These neighbouring residues are termed the anchors; specifically the N-anchor for the one closest to the N-terminus of the sequence and C-anchor for the one nearest the C-terminus (Figure 1.6).

As previously stated, methods for predicting protein loop structures are divided into two categories, knowledge-based and *ab initio*, depending on how they generate possible conformations (decoys). Knowledge-based methods rely upon databases of previously observed protein structure fragments. Structures are selected according to certain criteria such as fragment length (*i.e.* they must be the same length as the target loop), fragment-target sequence similarity and how closely the anchor geometry of the fragment matches that of the target loop. Methods of this type are extremely fast, and can be very accurate when the structure of the target loop is similar to one previously observed (Choi and Deane, 2010). However, there is not currently enough structural data to cover the conformational space, especially for long loops (Totrov, 2012). When a similar loop structure has not been observed previously, knowledge-based methods either give poor predictions or fail to return a prediction at all. Examples of this type of algorithm include FREAD (Deane and Blundell, 2001; Choi and Deane,

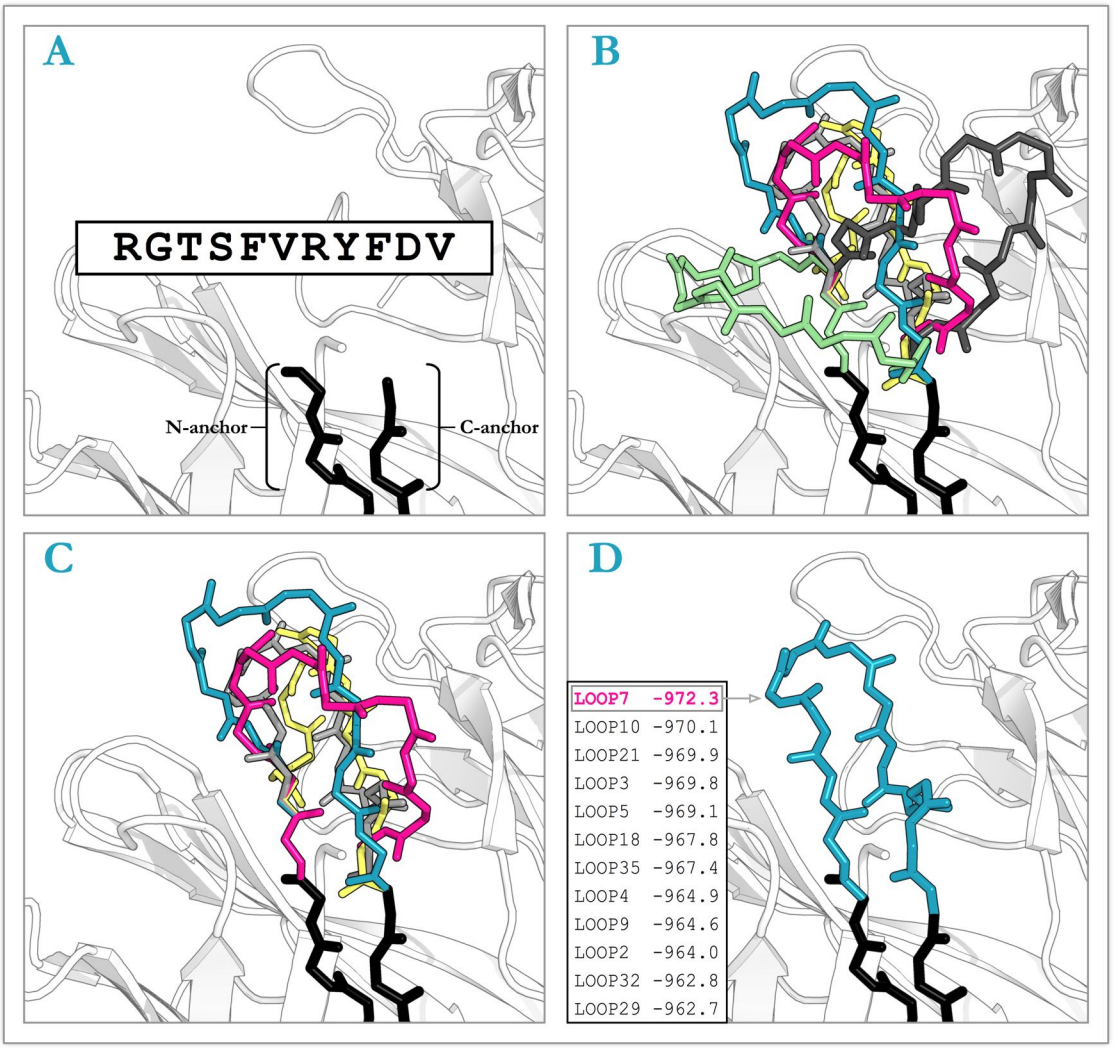

Figure 1.6: The main steps in a loop modelling algorithm. A — the inputs to the algorithm are a protein structure with a missing loop, and the sequence of that loop. B — Decoy generation. C — Filtering of structures that are physically impossible, *e.g.* ones that clash with the rest of the structure. D — Ranking and selection of the final prediction.

2010), SuperLooper (Hildebrand *et al.*, 2009), LoopWeaver (Holtby *et al.*, 2013) and LoopIng (Messih *et al.*, 2015).

Ab initio methods do not rely on previously observed structures; instead, decoys are produced computationally. *Ab initio* methods work by exploring the possible conformational space, for example by randomly sampling the ϕ and ψ dihedral angles of the loop. While this allows novel structures to be generated, like knowledge-based methods *ab initio* algorithms have their limitations: they are computationally expensive, since many decoys must be generated to sample the conformational space sufficiently; and their prediction accuracy decreases with loop length (as the number of degrees

of freedom increases). *Ab initio* algorithms include PLOP (Jacobson *et al.*, 2004), Modeller (Fiser *et al.*, 2000), Loopy (Xiang *et al.*, 2002), LoopBuilder (Soto *et al.*, 2007), LEAP (Liang *et al.*, 2014), and the loop modelling routine within Rosetta (Stein and Kortemme, 2013).

The idea of a hybrid loop modelling algorithm, combining knowledge-based and *ab initio* approaches, has been explored. CODA (Deane and Blundell, 2001) generates decoys using a knowledge-based method and an *ab initio* method separately, then combines the two decoy sets and makes a consensus prediction. Martin *et al.* (1989), Whitelegg and Rees (2000), and Fasnacht *et al.* (2014) have used similar approaches, and applied it to modelling the complementarity determining regions of antibodies — initial conformations are selected from a database of structures, and the middle section is then remodelled using *ab initio* techniques. An alternative approach using Rosetta is described in Rohl *et al.*, 2004 — this used a Monte Carlo-based fragment assembly method (*i.e.* changing sections of the loop structure to match a randomly selected short fragment, and accepting or rejecting this 'move' based on how the energy of the system changes), in conjunction with a minimisation protocol.

Depending on how the loops are built, the continuity of the protein backbone may need to be enforced through the implementation of a closure algorithm. Alternatively, a minimisation step may be introduced, where the energy function has a term that penalises an 'open' loop. Three types of loop closure algorithm exist: analytical, iterative and build-up. Analytical methods calculate the values of particular degrees of freedom that are required to produce a continuous backbone (for example, ϕ/ψ angles). This approach was first introduced by Go and Scheraga (1970) — they showed that the ϕ/ψ values necessary to close a loop can be solved mathematically for up to six angles. This approach is used to maintain loop closure in the loop modelling routine within Rosetta, in the algorithm called kinematic closure or KIC (Mandell *et al.*, 2009). Similar algorithms are used in robotics, to move multi-jointed 'arms' to specific locations in space (Totrov, 2012).

Iterative methods normally start with an open conformation, and gradually enforce its closure through a series of steps. A key example of this type is cyclic coordinate

descent, or CCD (Canutescu and Dunbrack Jr., 2003) — starting at one end of the loop, each ϕ or ψ angle is altered so that the distance between the free end of the loop and the fixed anchor is minimised. This continues iteratively, until the distance between the two ends is low enough to consider the loop closed. The change in angle required is calculated analytically; CCD can therefore be thought of as both an analytical and an iterative method.

Build-up methods attempt to guide loop building such that a closed loop conformation is automatically generated. RAPPER, for example, builds loops starting from the N-anchor, and places restraints on each Cα atom added to the structure, limiting the distance they are allowed to be from the C-anchor (DePristo *et al.*, 2003). Loop closure is enforced by making the restriction gradually tighter as more residues are added.

Filtering

Some decoys generated will not be physically possible. For example, ϕ/ψ angles of the structure may be in the disallowed regions of the Ramachandran plot (see Figure 1.4), or some atoms may be too close together. A filtering step is therefore required to remove these structures. This step may be combined with the other parts of the loop modelling process; for example some algorithms combine it with decoy generation itself. The Direct Tweak loop closure method, for example, enforces a continuous backbone whilst monitoring the loop for clashes (Xiang *et al.*, 2002).

Ranking

Once all decoys have been generated, it is necessary to use a ranking system to select a final prediction; *i.e.* the one that is predicted to be closest to the true structure of the target (the native structure). This is a vital step; even if decoys close to the native structure have been generated at a previous stage, an ineffective ranking system means that the structure chosen as the final prediction will be inaccurate.

For knowledge-based methods, the ranking system may use properties of the decoy/fragment structure — for example the similarity between the target sequence and the decoy sequence, or between the geometry of the decoy anchors and the anchors

of the target. FREAD, for example, ranks the fragments selected from a database by their anchor RMSD.

More commonly, especially for *ab initio* methods, an energy function is used to predict which structures are lower in energy and therefore more likely to be near-native. There are two main types of energy function: physics-based force fields and statistical potentials (Lazaridis and Karplus, 2000). Force fields are equations with separate terms for the contribution of different properties to the energetics of a system. These include bonded interactions, such as bond lengths, bond angles, and dihedral angles; and non-bonded interactions, like electrostatics and van der Waals' forces (Ponder and Case, 2003). The terms are parameterised using empirical evidence. Some examples of force fields are AMBER (Weiner *et al.*, 1984), CHARMM (Brooks *et al.*, 1983) and OPLS (Jorgensen *et al.*, 1996).

Statistical (or knowledge-based) potentials use pre-observed structures to infer the relative energy of a protein — based on the assumption that the distributions of particular structural features seen in nature reflect energetics (Lazaridis and Karplus, 2000; Totrov, 2012). For example, the carbon to oxygen bond in the carbonyl of the protein backbone is regularly observed in experimentally-determined structures to have a length of 1.23 Å (Engh and Huber, 1991) — a decoy with a C-O length of around this value is therefore likely to be more energetically stable than one with a C-O distance of 2 Å. Statistical potentials are attractive because the protein energetics do not necessarily need to be deciphered — these functions incorporate unknown or poorly-understood interaction terms into their calculation without having to explicitly include them (Lazaridis and Karplus, 2000). In addition, they are often faster to run than force field calculations, automatically include solvation and the potential is smoother — small changes in conformation do not lead to huge differences in energy. Examples of statistical potentials include DFIRE (Zhang *et al.*, 2004), DOPE (Shen and Sali, 2006), GOAP (Zhou and Skolnick, 2011) and SOAP-Loop (Dong *et al.*, 2013).

1.2.2 Measurement of Accuracy

The accuracy of a loop modelling algorithm is normally evaluated using the root mean square difference, or RMSD. This is a measure of the average distance between the equivalent atoms of two structures — a lower value therefore indicates higher accuracy. The RMSD between two structures, a and b, is calculated as follows:

$$\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((a_{ix}-b_{ix})^2+(a_{iy}-b_{iy})^2+(a_{iz}-b_{iz})^2)}$$

where a_{ix} is the x-coordinate of atom i in structure a, and N is the number of atoms considered. Commonly, RMSDs are calculated using backbone atoms (N, Cα, C and O) but different sets of atoms are sometimes used (*e.g.* only Cα atoms).

There are two types of RMSD: global and local. In terms of loop modelling, the global RMSD is the RMSD when the whole protein structure on which the loop has been predicted is structurally aligned onto the native; *i.e.* the RMSD between the two protein structures is minimised. The local RMSD is calculated when just the loops are superimposed, and is normally lower. In each case, only the atoms of the target loop are used in the calculation.

1.2.3 Algorithms and Current Accuracy

The accuracies achieved by some loop modelling algorithms are given in Table 1.1. Selected examples are described in more detail below.

FREAD

FREAD is a knowledge-based method that selects possible loop conformations from a database of experimentally determined protein fragments (Deane and Blundell, 2001; Choi and Deane, 2010). Loops are initially selected as potential predictions according to the separation of their anchor residues compared to that of the target and their sequence similarity to the target loop. The fragments are filtered depending on whether their insertion into the protein structure would cause clashes. These fragments (assuming some suitable fragments are found) are then ranked by the RMSD between the anchor residues of the fragment and those of the target loop; the loop with the lowest RMSD is

Table 1.1: Reported accuracies for some loop modelling algorithms. Values given are average global RMSDs, and the values in brackets after each RMSD are the number of loops predicted followed by their length (in residues). The target sets used to evaluate the performance of FREAD, LEAP and LoopIng are the same.

Algorithm	Type	Key RMSD Results			Reference
FREAD	Knowledge-Based	2.07 Å (30 targets of each length between 4-20) 1.15 Å for loops of 8 residues 286 of the 510 targets predicted (coverage = 56%)			Deane and Blundell, 2001; Choi and Deane, 2010
CODA	Combination	0.76 Å (50, 3) 1.00 Å (50, 4) overall coverage = 93.5%	1.50 Å (50, 5) 1.95 Å (50, 6)	2.87 Å (50, 7) 3.09 Å (50, 8)	Deane and Blundell, 2001
LEAP	*Ab initio*	0.39 Å (30, 4) 0.40 Å (30, 5) 0.49 Å (30, 6) 0.69 Å (30, 7) 0.68 Å (30, 8) 0.93 Å (30, 9)	1.44 Å (30, 10) 2.24 Å (30, 11) 3.14 Å (30, 12) 2.91 Å (30, 13) 4.44 Å (30, 14) 4.58 Å (30, 15)	4.90 Å (30, 16) 5.66 Å (30, 17) 6.53 Å (30, 18) 5.87 Å (30, 19) 8.21 Å (30, 20)	Liang *et al.*, 2014
LoopIng	Knowledge-Based	0.61 Å (30, 4) 0.68 Å (30, 5) 1.01 Å (30, 6) 1.26 Å (30, 7) 1.47 Å (30, 8) 1.71 Å (30, 9)	1.90 Å (30, 10) 1.93 Å (30, 11) 2.20 Å (30, 12) 2.39 Å (30, 13) 2.53 Å (30, 14) 3.05 Å (30, 15)	2.82 Å (30, 16) 3.03 Å (30, 17) 3.86 Å (30, 18) 3.89 Å (30, 19) 3.91 Å (30, 20)	Messih *et al.*, 2015
Loopy	*Ab initio*	0.85 Å (161, 5) 0.92 Å (107, 6) 1.23 Å (74, 7)	1.45 Å (61, 8) 2.68 Å (58, 9) 2.21 Å (34, 10)	3.52 Å (37, 11) 3.42 Å (21, 12)	Xiang *et al.*, 2002
LoopBuilder	*Ab initio*	1.31 Å (63, 8) 1.88 Å (56, 9)	1.93 Å (40, 10) 2.50 Å (54, 11)	2.65 Å (40, 12) 3.74 Å (40, 13)	Soto *et al.*, 2007
LoopWeaver	Knowledge-Based	1.27 Å (205, 6) 1.85 Å (171, 7)	2.59 Å (118, 8) 2.91 Å (101, 9)	3.33 Å (60, 10) 3.37 Å (43, 11)	Holtby *et al.*, 2003
ModLoop	*Ab initio*	0.59 Å (40, 4)	1.16 Å (40, 8)	2.61 Å (40, 12)	Fiser and Sali, 2003
PLOP	*Ab initio*	0.24 Å (35, 4) 0.43 Å (117, 5) 0.52 Å (100, 6)	0.61 Å (82, 7) 0.84 Å (66, 8) 1.28 Å (57, 9)	1.22 Å (40, 10) 1.63 Å (18, 11) 2.28 Å (10, 12)	Jacobson *et al.*, 2004
RAPPER	*Ab initio*	1.20 Å (35, 4)	2.90 Å (32, 8)	6.2 Å (34, 12)	DePristo *et al.*, 2003
Rosetta (1)	*Ab initio*	1.96 Å (25, 12)			Stein and Kortemme, 2013
Rosetta (2)	Combination	0.42 Å (40, 4)	0.97 Å (40, 8)	2.23 Å (40, 12)	Rohl *et al.*, 2004
SuperLooper	Knowledge-Based	2.60 Å (14, 11) < 2 Å on average for up to 6 residues; < 3 Å for up to 10	4.00 Å (10, 12)		Hildebrand *et al.*, 2009

assumed to have the most similar structure to the target and is hence returned as the final prediction. If no suitable fragments are found, however, FREAD does not produce a prediction. The computational time required varies with the size of the database being searched, but is normally around 1-2 minutes (on a 2.4 GHz Linux platform).

CODA

CODA (Deane and Blundell, 2001) is a consensus method, that compiles sets of decoys from a knowledge-based algorithm (FREAD) and an *ab initio* one (PETRA). PETRA is somewhat like a knowledge-based method, in that a database of structures is searched, however the protein fragments contained within this database were generated computationally. The database contains all possible fragments of 12 residues that can be made using eight pairs of ϕ/ψ values — these values were chosen so that the conformational space is adequately covered. Like FREAD, the database is searched using Cα separations and fragments are filtered based on potential clashes. After suitable decoys are found using both algorithms, all possible pairs (*i.e.* one from FREAD and one from PETRA) are formed, and the best pair is chosen based on criteria such as their predicted energy and their dihedral angles. Factors such as these are also used to decide which decoy from the best pair to use as the final prediction.

LEAP

LEAP (Loop prediction by Energy-Assisted Protocol) is a multi-step *ab initio* algorithm that uses various energy functions of different complexity to gradually reduce the number of decoys (Liang *et al.*, 2014). Loop prediction begins with the generation of many possible conformations using the CCD algorithm (Canutescu and Dunbrack Jr., 2003); 10,000 to 1,000,000 are made depending on loop length. There are then three subsequent stages that reduce the number of decoys to be considered: a preliminary ranking step using a coarse-grained energy function (used to reduce the number of decoys to 1,000); sidechain modelling and refinement of the loop backbone (leading to a set of 10 structures); and a final all-atom minimisation step. The decoy with the lowest energy is selected as the final prediction. There is no explicit filtering step, but infeasible structures (*e.g.* those that form clashes) will have higher energies and therefore

will not be carried forward to the following stage. In practice, the authors recommend running the procedure ten times for each target, taking the lowest-energy decoy of the ten produced as the final prediction. Each run is reported to require 1-10 hours of computation time (on a single Intel Xeon processor operating at 3.5 GHz), depending on the length of the target (Liang *et al.*, 2014).

LoopIng

LoopIng is a database method that uses a Random Forest algorithm (a machine learning method) to select fragments (Messih *et al.*, 2015). Candidate structures are pre-selected from a database based on the distance between the anchors and sequence similarity. A Random Forest model, trained on known loop structures, is then used to predict the RMSD between the native structure of the target and each of the candidate fragments. Random Forests work through the use of several 'decision trees' — given information about a target loop and previously observed structures, each tree returns the loop of known structure that it predicts to be the most similar. The final prediction is determined via a voting system; the loop most often returned by the set of decision trees is selected. Several input features are used: similarity scores for each fragment residue compared to each target residue; the sequence similarity calculated over the whole loop; the distance between the Cα atoms just before and after the loop; secondary structure of anchor residues; and anchor geometry, based on the three angles defined by Martí-Renom *et al.* (1998). The final prediction is the candidate structure that has the lowest predicted RMSD to the target structure.

Loopy and LoopBuilder

Loopy is an *ab initio* method that takes account of the shape of the potential energy curve (Xiang *et al.*, 2002). It assumes that each loop decoy generated is just one of a set of similar conformations — the number of conformations represented by the one sampled is estimated using statistical mechanics, by looking at whether any other decoys had a similar conformation (the presence of lots of similar conformations indicates that they are located in a broad energy basin). This is converted into a value called the colony energy, which is used to score and rank the decoys. Initially, 2000 decoys are generated

using ϕ/ψ angles chosen from Ramachandran distributions, and the random tweak closure method. Sidechains are added, and the structures undergo minimisation using a simple force field that only contains a van der Waals' term. Ranking occurs in a series of steps. The 1000 best models after the first minimisation stage are retained; the colony energy is calculated for each one and the top 300 are kept. All loop decoys are then compared to one another, and pairs of loops that have an RMSD greater than one tenth of the loop length but less than the loop length are combined into a single decoy by taking half of each loop. Thirty percent of the loop conformations are kept according to their colony energy — this step is then repeated until only one structure remains.

LoopBuilder is another *ab initio* algorithm based on Loopy (Soto *et al.*, 2007). Decoys are generated in the same way, but are scored using DFIRE (Zhang *et al.*, 2004) and minimised using PLOP (Jacobson *et al.*, 2004).

LoopWeaver

The knowledge-based algorithm LoopWeaver introduced a new method of inserting loop fragments into the protein structure that minimises clashes and unrealistic ϕ/ψ angles (Holtby *et al.*, 2013). Decoys are selected from a database according to their anchor RMSD — the top 500 structures are returned, no matter how high their anchor RMSDs might be. The decoy is placed onto the anchors using weighted multi-dimensional scaling — essentially, this expresses the 'stress' of the loop anchors as a function of the anchor residue coordinates for both the fragment and target. This function is minimised to find the optimal anchor coordinates. The final prediction is selected using the statistical potential DFIRE (Zhang *et al.*, 2004).

ModLoop

ModLoop, the *ab initio* loop prediction algorithm within MODELLER, is different to a lot of other *ab initio* methods in that it starts with a closed loop structure, with the atoms positioned uniformly along a line between the two anchor residues (Fiser *et al.*, 2000; Fiser and Sali, 2003). This structure is refined using the CHARMM force field (Brooks *et al.*, 1983), so that the geometry of the loop gradually becomes more realistic. The continuity of the backbone is also maintained throughout. Dihedral angles

are therefore not explicitly sampled like other methods — existing angles are changed to more probable values that minimise the energy function. The process is repeated multiple times, and the final prediction returned is the one with the lowest energy.

PLOP

The Protein Local Optimisation Program, or PLOP, is the *ab initio* loop modelling algorithm within Schrödinger's Prime software (Jacobson *et al.*, 2004). In this algorithm, a 'full' prediction job is made up of many 'standard' jobs. Like many other *ab initio* methods, in a standard job loops are built by choosing random ϕ/ψ angles from Ramachandran distributions. However, instead of building loops by adding all residues onto one of the anchors with subsequent closure, they are built in two segments — half onto the N-anchor and half onto the C-anchor. Many structures are created for each half of the loop. All N-anchor segments are then compared against all C-anchor segments to find pairs that meet in the centre, thereby forming a complete loop structure. Decoys that have unrealistic dihedral angles or clash with the rest of the protein are filtered out, and all remaining loop structures are then clustered (grouped by structural similarity), and the representative structures (those from the centre of each cluster) undergo energy minimisation.

A full prediction job is then made up of a series of standard jobs, with the conformational search space becoming narrower at each stage. The aim is to guide the conformational search to focus on structures with low energy. The final step is the ranking of all the loop structures that were generated from all steps, according to their calculated energy; the loop with the lowest energy is returned as the final prediction.

RAPPER

RAPPER (DePristo *et al.*, 2003) is an *ab initio* loop modelling algorithm. Conformations are generated by adding one residue at a time onto the N-anchor, by choosing random ϕ/ψ pairs from a distribution and positioning the atoms accordingly. Restraints are placed upon the atom positions to enforce loop closure — the Cα of residue i, in a n-residue loop, must be located within a sphere of radius $3.8(n-i+1)+1$ Å, centred on the Cα atom of the C-anchor residue. Conformations are made until 1,000 have

been generated, with no two conformations having an RMSD of less than 0.2 Å to try and ensure that the conformational space has been explored adequately. The decoys are then ranked using a statistical potential called the residue-specific all-atom probability discriminatory function, or RAPDF (Samudrala and Moult, 1998).

Rosetta

Loop modelling in Rosetta is carried out using a kinematic closure protocol, made up of 'KIC moves' (Mandell *et al.*, 2009; Stein and Kortemme, 2013). Prediction begins with the generation of a random loop structure. During a KIC move, three Cα atoms of the loop segment are chosen as 'pivots', leaving the remaining Cα atoms as 'non-pivots'. Dihedral angles of the non-pivots are sampled from Ramachandran distributions, similarly to RAPPER and PLOP. The dihedral angle changes of the pivots required to maintain loop closure are then calculated analytically. The full protocol, which includes sidechain optimisation and backbone energy minimisation, involves carrying out KIC moves iteratively, with different pivot atoms each time. The lowest scoring model, according to the statistical Rosetta scoring function, is reported as the final loop prediction.

'Next-generation KIC' is a new version of this algorithm (Stein and Kortemme, 2013). This improved protocol includes the sampling of ω dihedral angles, neighbour-dependent ϕ/ψ sampling and annealing (the weights of terms in the Rosetta energy function are slowly increased so that energy barriers can be overcome easily).

An algorithm that uses Rosetta's approach to *de novo* protein structure prediction has also been proposed (Rohl *et al.*, 2004). This algorithm uses a Monte-Carlo-based fragment assembly approach, in combination with a minimisation protocol. The energy function used to minimise the structures includes a loop closure term, that forces the continuity of the backbone. A Monte-Carlo 'move' involves changing the structure of a section of the loop to match that of a known protein fragment. The move is accepted or rejected based on the Metropolis criterion: if the move improves the predicted energy of the protein, then the move is accepted, otherwise it is accepted with a probability that is inversely proportional to the energy increase (*i.e.* a large increase in energy leads to a reduced probability of acceptance).

SuperLooper

SuperLooper (Hildebrand *et al.*, 2009) is a knowledge-based method which selects fragments from the Loops In Proteins (LIP) database (Michalsky *et al.*, 2003). This database contains all protein fragments of between 1-15 residues found in the PDB — at the time of publication, this was approximately 10^8 fragments. SuperLooper searches this database in a similar manner to FREAD — candidate structures are selected based on the similarity of the fragment's anchor residues and sequence to the target. The set of decoys returned, however, is forced to be structurally diverse; candidate structures with identical sequences and similar backbone conformations to others are excluded. The final prediction is selected based on sequence similarity and anchor geometry.

Summary

Knowledge-based, *ab initio* and hybrid approaches have all been used for protein loop modelling. Knowledge-based methods include FREAD, LoopIng, LoopWeaver and SuperLooper — these methods differ in how they select possible loop structures from a database, or in how these loop structures are inserted into the target structure. As might be expected, these methods can give extremely accurate results, but may fail when the target loop structure is unlike what has been observed previously. For example, FREAD can give sub-Ångström predictions but occasionally fails to find any suitable fragments at all. There is more diversity in the *ab initio* methods; aspects that differ between algorithms include the method of adding residues to the decoy structure, the loop closure algorithm, and the ranking method. These methods (such as LEAP, Loopy, LoopBuilder, ModLoop, PLOP, RAPPER, and Rosetta) can be accurate for short loops, when the conformational space is smaller, but struggle with long loops. Another limiting factor is the computational time required: LEAP, for example, requires up to 10 hours to predict a single loop. However, *ab initio* algorithms are not limited by our knowledge of loop structures, since possible conformations are generated computationally. Some attempts have been made to create hybrid methods (*e.g.* CODA and the 2004 fragment-assembly version of Rosetta), but these have not been widely used and have been overtaken in accuracy by more recent software.

1.3 Antibodies

The immune system protects an organism against foreign substances, such as bacteria, viruses and other unwanted particles which may cause harm. Upon being infected, the first response is non-specific — strategies are used that are not targeted towards a particular substance. For example, many kinds of bacteria have similar components on their surface that may be recognised by some types of white blood cell, leading to their removal. This is referred to as the innate response.

When the unwanted substance is not recognised by the innate immune system, however, a different approach must be used to target that substance specifically. This is known as the adaptive response, and involves the production of proteins called antibodies.

Loop structure prediction is important in the case of antibodies, since their behaviour mainly depends upon six loops found in their binding site. For the majority of this thesis, we focus on one of these loops (the H3 loop), and describe our investigations into improving the accuracy of structure prediction.

1.3.1 Biological Function

Antibodies, or immunoglobulins, are proteins that bind to foreign objects that find their way into an organism, preventing them from causing harm and marking them for removal. A huge number of different antibodies can be produced, making them capable of binding to a huge range of substances; from proteins on the cell surface of bacteria to non-biological small molecules (Sela-Culang *et al.*, 2013). The substance that an antibody binds to is known as an antigen, and the specific region of the antigen to which the antibody binds is called the epitope. Mature antibodies bind with high affinity and are specific, meaning that they bind to other epitopes only very weakly, or not at all (Saper, 2009).

There are five varieties of human antibody, which differ in structure and role in the immune response. These varieties, known as isotypes, are labelled IgA, IgD, IgE, IgG and IgM; the most abundant type is IgG (Janeway *et al.*, 2001; Schroeder Jr and Cavacini, 2010).

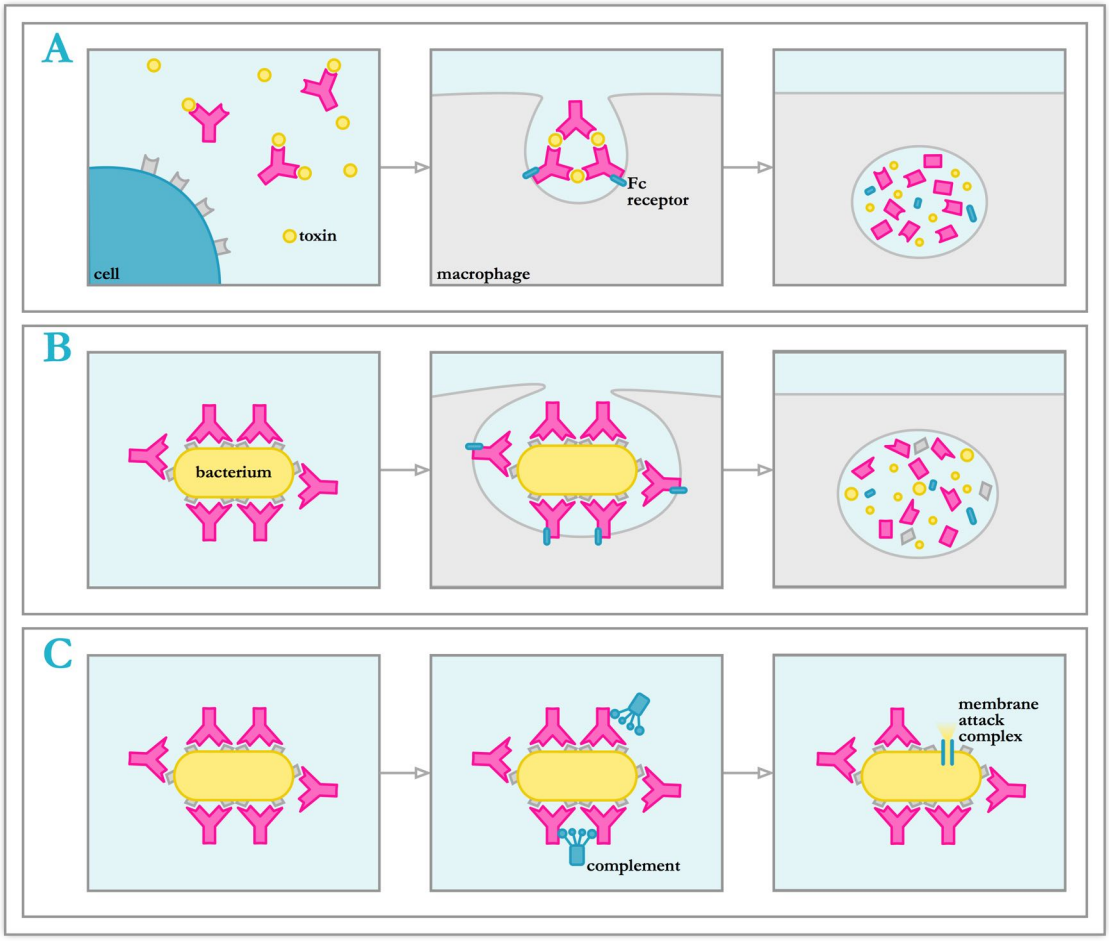

Figure 1.7: Antibody functions. A — neutralisation. Binding of an antigen by an antibody can stop it from harming cells by blocking the interaction. The antibody-antigen complex then binds to receptors on a macrophage and is digested. B — opsonisation. Antibodies act as markers, telling cells of the immune system that it needs to be removed. C — activation of the complement system. Other proteins of the immune system (complement) bind and cause phagocytosis or cell death through lysis. Figure adapted from Janeway *et al.* (2001).

Antibodies function in three main ways: neutralisation, opsonisation, and activation of the complement system (Figure 1.7, Janeway *et al.* 2001). Neutralisation occurs when the antibody binds to an antigen, preventing interactions between it and the organism's cells. For example, some toxins produced by bacteria bind to receptors on cell surfaces, inhibiting their function with potentially harmful consequences. If the toxin becomes bound to an antibody instead, then it can no longer bind to the receptor, leaving the cell free to carry out its purpose.

By binding to an antibody, antigens are 'tagged' for removal in a process called opsonisation (Janeway *et al.*, 2001). Once bound, the antibody interacts with receptors

on the surface of phagocytes (*e.g.* macrophages), a type of cell that can engulf and break down substances in a process called phagocytosis. The antigen is therefore removed from the organism and prevented from causing harm. Bacteria or viruses may have many antigens on their surface — this allows numerous antibodies to bind, forming a coat and allowing the whole pathogen to be phagocytosed efficiently.

The third way in which antibodies function is through the activation of the complement system (Janeway *et al.*, 2001). After antigen binding has occurred, other parts of the immune system are recruited, called complement. This set of proteins forms part of the innate immune system, and can non-specifically mark foreign substances and bring about their removal. Antibodies are able to direct the action of this system towards specific targets. Once recruited, complement can either cause phagocytosis or prompt cell death, via formation and activation of a membrane attack complex that ruptures the cell membrane (lysis) (Jiang *et al.*, 2011).

1.3.2 *In vivo* Generation

The process through which antibodies are created in the body is called clonal selection (Burnet, 1957; Janeway *et al.*, 2001). The procedure is summarised in Figure 1.8. B cells, a type of white blood cell responsible for producing the antibody molecule, are each able to produce only one kind of antibody; *i.e.* all those produced by the same cell will have exactly the same sequence. To begin with, a range of B cells in the bone marrow produce IgM-type antibodies, and display them on their surfaces as receptors. The range of antibodies available at this point is known as the germline repertoire. Providing they do not bind to self-antigens (molecules that are native to the organism and therefore not harmful), the B cells are transported from the bone marrow to the lymph nodes, where they begin producing IgD antibodies. At this point, they may come into contact with an antigen (Gonzalez *et al.*, 2011). Given the diversity of the repertoire, most of the antibodies will not bind, however some will. If binding occurs, the antigen is internalised by the B cell and parts of it are presented on its surface. Helper T cells recognise these fragments and release cytokines, activating the B cell and inducing proliferation. Some of the new B cells start producing soluble antibodies (IgM and

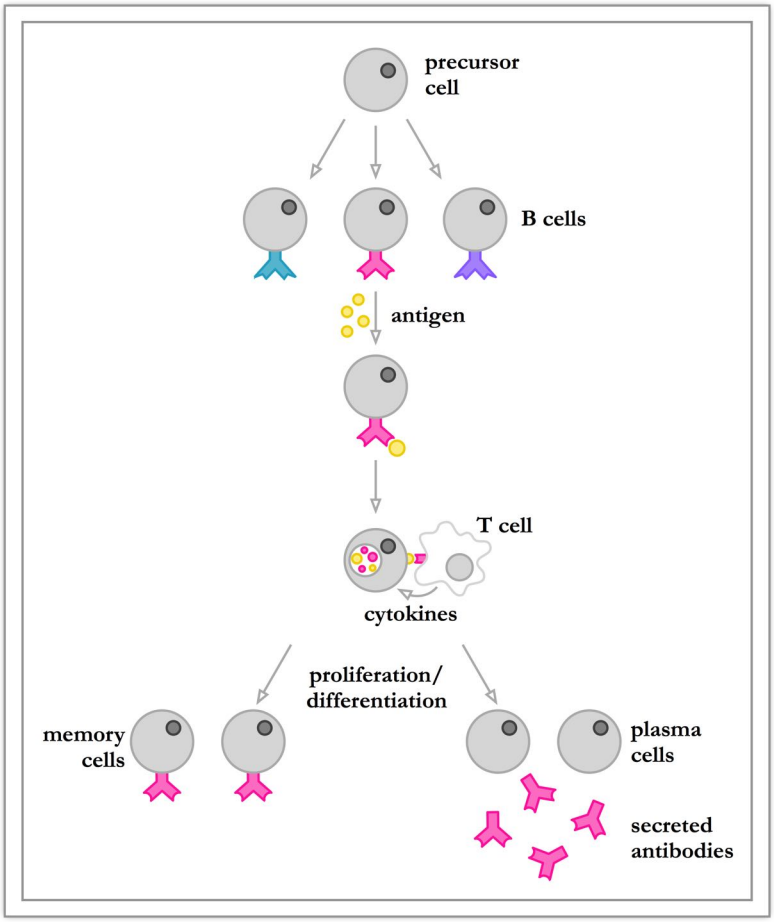

Figure 1.8: Clonal selection. Precursor cells become B-cells, each of which can produce a single type of antibody. On exposure to an antigen, the B cell that produces antibodies specific to that antigen binds to it, digests it and displays parts of it on the cell surface. Helper T cells recognise these fragments, and activate the B cell, causing it to proliferate. Some of the B cells secrete antibodies that go on to initiate the removal of the antigen; others become memory cells, and remain in the body in case the antigen is encountered in the future.

IgG), while some are retained by the immune system and act as a 'memory'; these can be activated if the host encounters the same antigen again. Multiple cycles of clonal selection normally take place, so that the affinity of antigen-binding is increased (this is known as affinity maturation; see also Section 1.3.3). At the end of the process, the final, mature B cells enter the bloodstream and secrete antibodies, which are then able to carry out their main functions (see Section 1.3.1).

1.3.3 Antibody Diversity

For the immune system to work effectively, different antibodies must be available to bind the extremely large range of antigens. This is indeed the case; it is estimated that the human antibody repertoire contains 10^{11} unique sequences (Glanville *et al.*, 2009). However, it cannot be a straightforward case of having one gene to code for every different antibody — estimates differ, but the human genome only contains around 20,000-25,000 genes (International Human Genome Sequencing Consortium, 2004), nowhere near enough to encode for 10^{11} different antibodies along with all other proteins that must be made. The body must therefore use other methods to create diversity in antibody structures. There are four main ways in which this occurs (Figure 1.9): V(D)J recombination, junctional diversification, combinatorial joining of light and heavy chains, and somatic hypermutation.

V(D)J Recombination

Antibodies are composed of two types of protein chain; known as the heavy and light chains. Each of the two types is encoded by multiple gene segments that are spliced together using a process called V(D)J recombination (Tonegawa, 1983, Figure 1.9A). The sequence for the light chain is made up of three sections: the variable segment (V), which encodes the first 95-101 amino acids; the joining segment (J), which codes for up to 13; and the constant region (C) (Janeway *et al.*, 2001). The heavy chain is also encoded from variable, joining and constant segments, but there is a fourth — the diversity segment (D), located between V and J. Before recombination, the constant region of the DNA sequence is located far away from the variable segment (stage 1 of Figure 1.9A). When the heavy chain of an antibody is to be expressed, the D segment and the J segment are bound together by an enzyme complex called V(D)J recombinase (stage 2, Figure 1.9A), bringing the D and C segments closer together (this does not occur for the light chain, as the D segment is not present). The variable segment is then bound to the DJ segment, so that all the segments are now in close proximity (stage 3, Figure 1.9A). The introns between the segments are subsequently removed, producing mRNA that can be translated into the heavy chain (stages 4 and 5, Figure 1.9A).

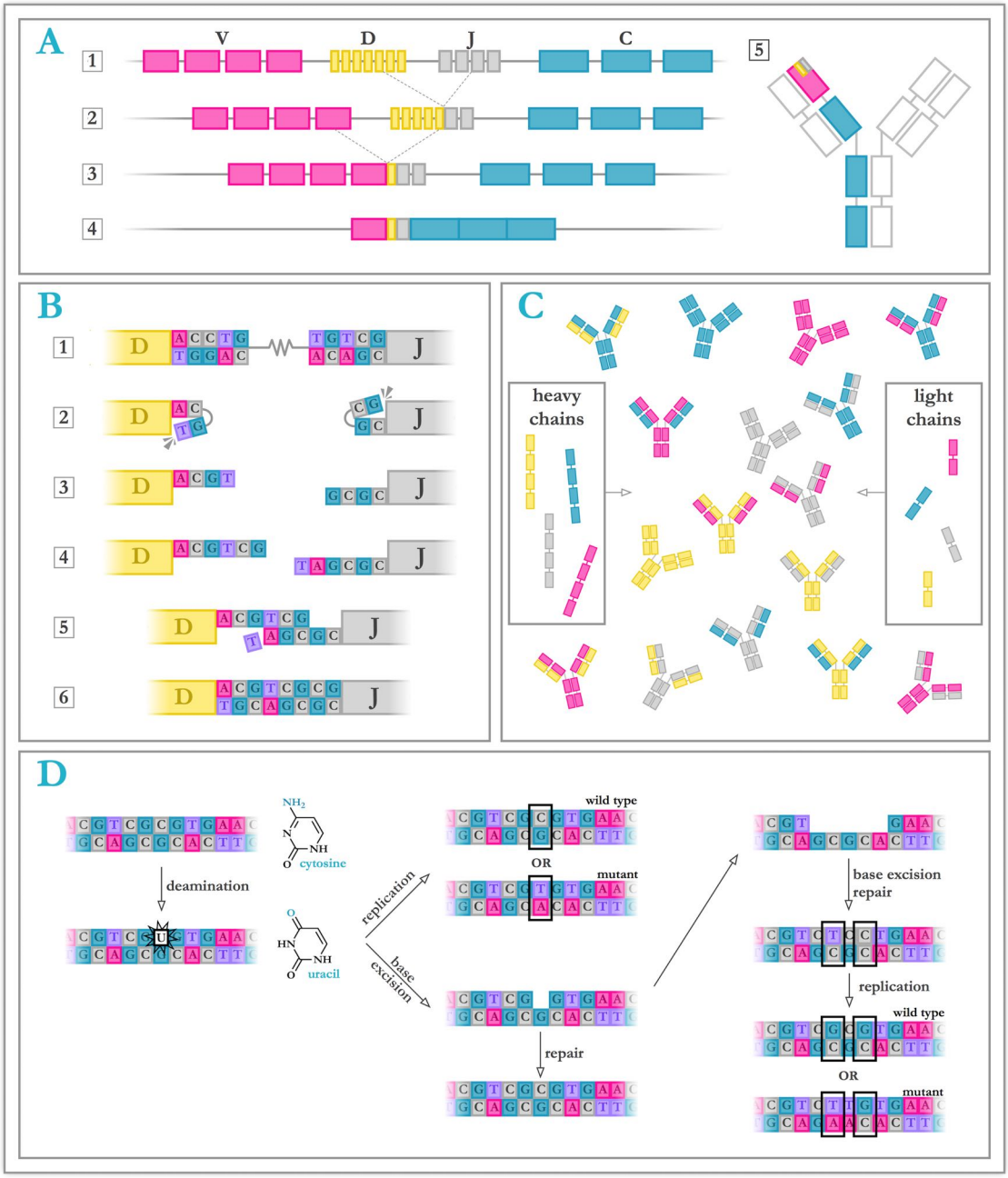

Figure 1.9: Sources of antibody diversity. A — V(D)J recombination. B — Junctional diversification. C — Combinatorial joining of heavy and light chains. D — Somatic hypermutation. See main text for details.

In reality, there is not only one gene for each of the V, D and J segments; there are many, which can be matched up in different combinations to produce a diverse range of antibodies. To take the heavy chain as an example, there are 65 V, 27 D and 6 J segments, giving a total of 10,530 possible combinations (Market and Papavasiliou, 2003). This is an efficient way of creating a large repertoire of antibodies. However, not all the gene

segments that have been discovered are functional — the large number of gene segments available to choose from has weakened the evolutionary pressure on each gene, and so some have accumulated mutations, rendering them non-functional. Such genes are called pseudogenes. Another factor to consider is that not all the V segments are chosen with the same probability, some are used only rarely (Feeney *et al.*, 2000).

Junctional Diversification

There are three loops on each chain of an antibody. The DNA encoding for two of them is found in the V segment, while the other is found at the join between the V and J segments (and the D segment for the heavy chain). During recombination, nucleotides can be added to or removed from the DNA sequence at the segment junctions to further diversify the resulting antibody structures. The stages involved in this process are shown in Figure 1.9B. First of all, the enzyme complex V(D)J recombinase binds to the DNA between two segments and cleaves it, producing a DNA hairpin. Cleavage of this hairpin then generates P-nucleotides, so called because their sequences are palindromic. The enzyme terminal deoxynucleotidyl transferase (TdT) then adds up to 20 nucleotides to each strand, which subsequently pair up. Any non-matching bases are removed by an exonuclease enzyme, any gaps in the base pairing are filled, and finally DNA ligase completely fuses the nucleotides to form the coding joint. Nucleotides can also be deleted from the sequence. The DNA sequence can be changed considerably in this way, further increasing the number of antigens that can be recognised.

Combinatorial Joining of Heavy and Light Chains

Just as diversity is generated by putting together different combinations of gene segments, different pairings of heavy and light chains can be made. Given the number of heavy and light chains that can be created due to V(D)J recombination, excluding the effect of junctional diversification, and assuming that every heavy chain can pair with every light chain, there are approximately 3.5×10^6 possible antibody sequences (Market and Papavasiliou, 2003).

Somatic Hypermutation

During the proliferation of B cells in clonal selection (see Section 1.3.2), the rate of mutation is increased compared to normal cells, in a process called somatic hypermutation (Figure 1.9D). An enzyme called Activation-Induced Deaminase (AID) catalyses the conversion of cytosine to uracil at various points in the DNA, causing mis-matching base pairs. When the DNA is replicated, this mutation leads to a B cell capable of producing a new antibody. Alternatively, the mis-matching uracil can be removed by base excision repair enzymes — in the process, additional surrounding bases can also be removed. DNA polymerases then re-fill the gap, however these enzymes are error-prone and more mis-matches are introduced. Again, this leads to a new antibody sequence after replication. If the new antibody has an increased binding affinity to the antigen, then the B cell that produced it is selected for further proliferation; this process is repeated to gradually increase the effectiveness of the antibody (affinity maturation). Mutations caused by somatic hypermutation occur at a rate of 10^{-3} per base pair per cell division; about a million times faster than spontaneous mutations (Odegard and Schatz, 2006). This means that affinity maturation occurs on much faster timescales than it would otherwise.

1.3.4 Therapeutic Applications and Engineering

The ability of antibodies to bind with high affinity and specificity to their targets means that they are especially good candidates for therapeutic and diagnostic applications. Since the first antibody treatment, muromonab, was approved in 1986 for the prevention of transplant rejection, the market has grown extremely quickly (Leavy, 2010). By 2012, antibody therapies accounted for over a third of the total sales for the biopharmaceutical sector in the US, and they are currently the biggest-selling class of biopharmaceuticals (Aggarwal, 2014).

Although molecules from biological sources tend to be larger, more complex and far more difficult to characterise than traditional small molecule drugs (Crommelin *et al.*, 2003), they are promising as therapeutic agents for several reasons. For example, they are able to interact with targets that small molecule drugs find difficult, such as

protein-protein interfaces. Small molecules can be targeted to specific locations by linking them to proteins (especially useful if the drug is toxic). Attrition rates are also lower: in 2014 it was reported that 14.6% ($n = 1{,}173$) of biopharmaceuticals that enter Phase I trials go on to be approved, nearly twice as many as small molecules (7.5%, $n = 3{,}496$) (Hay *et al.*, 2014). This is likely to be because biopharmaceuticals are more specific and hence less toxic. Antibodies can be used for many disease areas: some currently on the market include infliximab (Remicade) and adalimumab (Humira) for the treatment of rheumatoid arthritis; trastuzumab (Herceptin) and bevacizumab (Avastin) for cancer; and alemtuzumab (Lemtrada) for multiple sclerosis (Kuroda *et al.*, 2012).

Antibody therapeutics normally work by directing the body's immune response towards a particular target (Beck *et al.*, 2010). Alternatively, antibodies can be linked to small molecule drugs and used to deliver them to the correct place — for example, toxic cancer drugs can be targeted towards malignant cells to minimise contact with healthy ones (Scott *et al.*, 2012). Once a suitable target has been identified, antibodies that bind to it can be generated. Most therapeutic antibodies approved to date were produced using hybridoma technology (Köhler and Milstein, 1975). This process involves injecting mice with an antigen and harvesting its B cells (those that generate antibodies targeting that antigen). A problem with this method is that murine antibodies may be recognised as foreign by the human immune system and removed from the body. This can be overcome by minimising the murine content, *i.e.* replacing regions of the antibody's structure with their human equivalents. An alternative to hybridoma technology developed more recently is phage display (McCafferty *et al.*, 1990). Fully human antibodies can be produced in this way — bacteriophages that are engineered to display human antibodies on their surfaces are exposed to a target antigen, and those that bind are extracted and amplified.

1.3.5 Antibody Structure

Standard mammalian antibodies have a characteristic shape reminiscent of the letter Y; they consist of two identical halves, each containing a heavy and a light chain (Figure 1.10A+B). Heavy chains are made up of a series of domains: several constant domains,

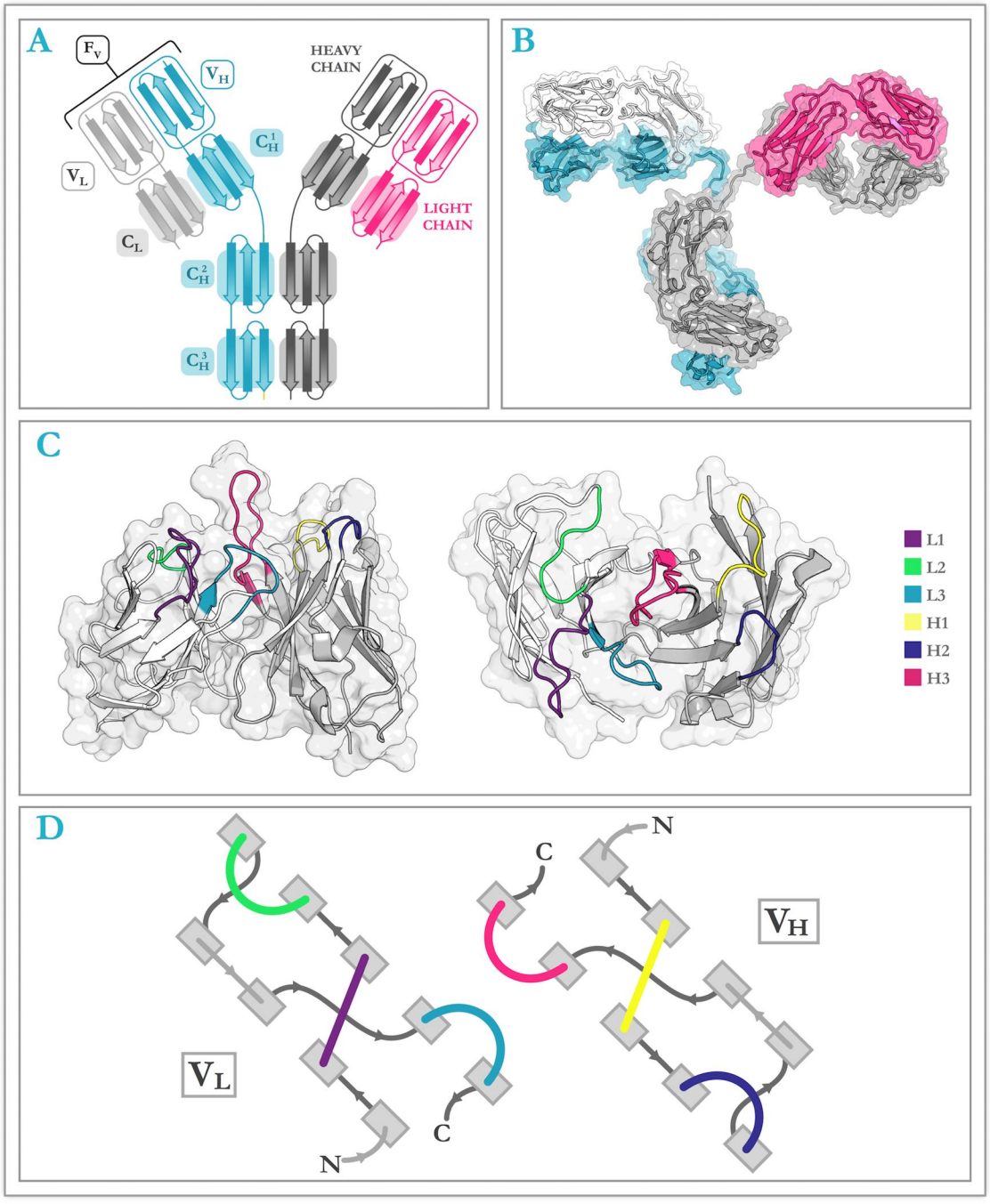

Figure 1.10: Antibody structure. A — An antibody is made up of four chains; two light (white and pink) and two heavy (grey and blue). Each chain is made up of a series of domains — the variable domains of the light and heavy chains together are known as the Fv region. B — An actual antibody structure (PDB code 1IGT). C — The structure of the Fv, viewing the binding site from the side (left) and above (right). The complementarity determining regions (CDRs) are coloured according to the legend. D — The connectivity of the β strands in the VH and VL. The same colours are used for the CDRs as in part C of this figure.

labelled CH1, CH2, CH3 etc., and one variable domain, VH. Light chains have two domains: one constant (CL) and one variable (VL). The DNA encoding the variable and constant domains is found in the V and C gene segments, respectively (see Section 1.3.3). All domains are composed of β-sheets. The five isotypes differ in the structure of their constant domains: for example IgA, IgD and IgG antibodies have three constant domains on the heavy chain, while IgE and IgM have four. The heavy and light chains are held together by both non-covalent and covalent bonds (disulphide bonds between cysteine residues). Each domain consists of approximately 110 amino acids; light and heavy chains are therefore normally about 220 and 440 residues long respectively (for IgA, IgD and IgG).

Higher vertebrates have two types of light chains, κ and λ, which differ in sequence. These two types do not seem to have different functions. The proportion of each type varies from species to species — for instance, 60% of human IgG antibodies contain κ light chains, but mouse antibodies are approximately 95% κ-type (Finlay and Almagro, 2012).

The variable domains of both the heavy and light chain are together known as the Fv region; most naturally occurring antibodies have two (one on each 'arm' of the Y). A typical example of an Fv region is shown in Figure 1.10C, and the connectivity of the β-strands is shown in Figure 1.10D. The fact that antibodies are bivalent (they have two binding sites), means if an antigen is polyvalent, *i.e.* it contains multiple copies of a specific epitope (as is often the case), cross-linking can occur. This forms a large lattice of interconnected antigens and antibodies, which can be efficiently phagocytosed by macrophages (Pommerville, 2012). The flexible hinge between the two Fv regions increases the ease of production of this lattice by allowing the distance between binding sites to vary (Stanfield *et al.*, 2006).

The Fv region features six loops, known as complementarity determining regions or CDRs, that form the majority of the antigen binding site. There are three CDRs on each of the heavy and light chains; those located in the VL domain are labelled L1, L2 and L3, and those found on the VH domain are labelled H1, H2 and H3. These loops are the most variable part of the whole structure, and they govern the majority

of the antigen-binding properties of an antibody. The parts of the variable region that are not part of the CDRs are called the framework.

It is the structure of the antigen-binding site that determines the type of antigen an antibody can bind to. Anti-protein antibodies tend to have flat binding sites; anti-peptide antibodies have grooved sites mainly due to a longer L1 loop (Finlay and Almagro, 2012). Anti-hapten (small molecule) antibodies normally have small binding sites (Lee *et al.*, 2006), with antigen-antibody contacts further inside the interface between the two variable regions (where proteins and peptides generally cannot reach due to their larger size). The H3 loop contributes largely to the topography of the binding site — long H3s can create finger-like protrusions that can interact with epitopes not found directly on the antigen's surface, and short H3s create cavities in the antibody surface, with a specific shape that only allows certain antigens with smaller or protruding epitopes to bind (Finlay and Almagro, 2012).

Canonical CDR Structures

The conformational diversity of five of the six CDRs (L1, L2, L3, H1 and H2) is thought to be limited. For these CDRs, only a few different shapes have been observed, forming a set of discrete conformational classes known as canonical structures. Since its proposal in 1987 (Chothia and Lesk, 1987), the idea has been reinvestigated many times as the number of known antibody structures has increased (*e.g.* Martin and Thornton, 1996; North *et al.*, 2011; Nowak *et al.*, 2016). These studies have led to the identification of particular amino acids at certain positions that are thought to be structure-determining; the canonical class of a CDR of unknown structure can therefore be predicted from its sequence with high accuracy. The least diverse CDR is L2, with around 99% of known structures belonging to the same class (Nowak *et al.*, 2016).

CDR H3

Unlike the other five CDRs, the H3 loop cannot be classified into canonical forms because of its diversity: a huge range of structures have been observed. This is due to how antibody sequences are encoded in the genome, which leads to a huge number of

possible sequences — the DNA encoding the H3 loop is found at the join between the V, D and J gene segments (see Section 1.3.3). H3 loops vary widely in length: most are between 3 and 25 residues but they are occasionally far longer. Bovine antibodies, for example, have H3s that are 50 or even 60 residues in length (Wang *et al.*, 2013). For comparison, the canonical CDRs each have a most 8 different lengths, and are normally far shorter — the longest canonical form is 17 residues long, but there are few examples of these five loops with lengths over 13 (North *et al.*, 2011).

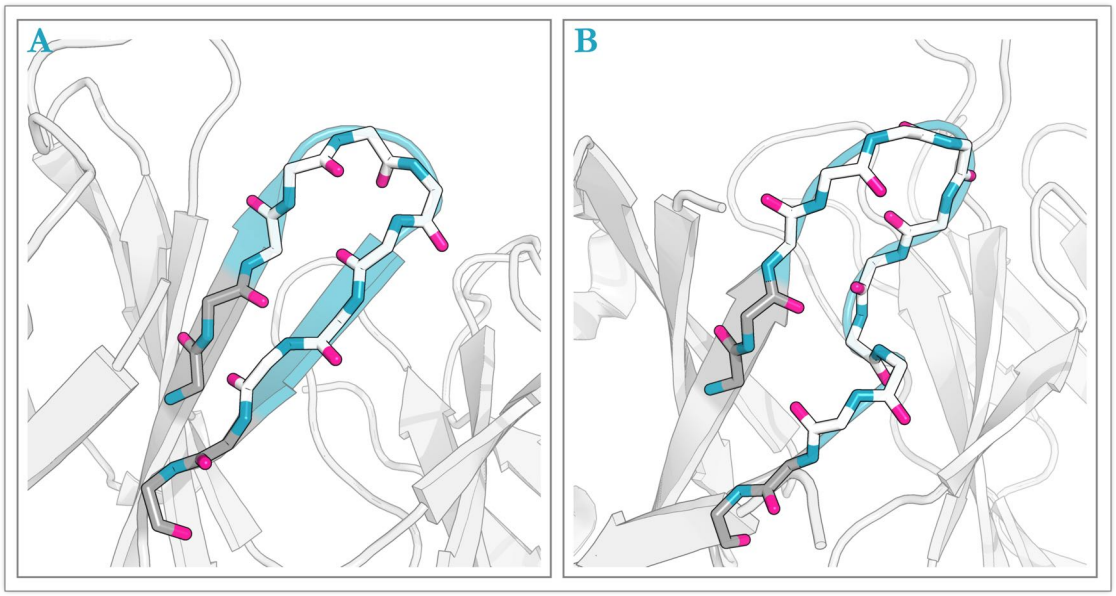

Figure 1.11: The 'torso' region of the H3 loop has been observed in two conformations: extended (A) and kinked (B). Anchor residues are shown in stick representation in grey. The majority of H3 structures are kinked.

The 'torso' of H3 loops (the residues nearest to the anchors) has been observed to adopt one of two conformations, labelled kinked (or bulged) or extended (or non-bulged — see Figure 1.11). The majority of H3 loops are kinked (Morea *et al.*, 1998; North *et al.*, 2011). Proposals have been made about why this is the case, such as the interaction of a basic residue in the N-anchor with an asparagine located within the loop, which have led to the development of rules that aim to predict which conformation will be adopted (Shirai *et al.*, 1996; Morea *et al.*, 1998). However, as more antibody structures have become available, these guidelines have been revisited and found to fail in some cases (North *et al.*, 2011).

It is the H3 loop that is thought to contribute the most to an antibody's antigen-binding properties (Kuroda *et al.*, 2012). It is located in the centre of the binding site, and forms the most contacts with the antigen (Alzari *et al.*, 1988; MacCallum *et al.*, 1996). It has also been shown to have the greatest effect on the energetics of binding (Kunik and Ofran, 2013), and to be the part of the antibody structure that changes the most upon binding (Sela-Culang *et al.*, 2012). Knowledge of H3 structures is therefore extremely useful, enabling predictions to be made about antibody binding properties (*e.g.* Clark *et al.*, 2006; Lippow *et al.*, 2007; Diskin *et al.*, 2011; Kuroda *et al.*, 2012; Krawczyk *et al.*, 2013; Kiyoshi *et al.*, 2014; Lewis *et al.*, 2014; Thakkar *et al.*, 2014).

Numbering of Antibody Sequences

Numbering schemes have been developed that are used to annotate equivalent positions within antibody sequences, and hence determine exactly which residues are part of the CDRs. This allows antibody sequences to be aligned and compared easily. One of the most common of these is the Chothia numbering scheme (Figure 1.12). By aligning the VH and VL regions of several antibody structures, it can be seen which parts of the structure are variable (the CDRs) and which are relatively constant (the framework). Chothia and Lesk (1987) used this information to determine equivalent positions in the structures and define where the CDRs start and end: residues 26-32 (H1), 52-56 (H2), and 95-102 (H3) on the heavy chain, and 24-34 (L1), 50-56 (L2), and 89-97 (L3) on the light chain. Other numbering systems exist, such as IMGT (Lefranc *et al.*, 2003), Kabat (Kabat *et al.*, 1983) and North (North *et al.*, 2011), which differ in their nomenclature and the positions at which they allow insertions and deletions. Since it was determined using known antibody structures, and our focus is on structure, we use the Chothia scheme throughout this thesis.

1.3.6 Antibody Modelling

Antibody structures are extremely useful when developing a novel therapeutic, allowing them to be engineered more rationally than when using methods such as hybridoma technology or phage display. They can be used to increase binding affinity by guiding

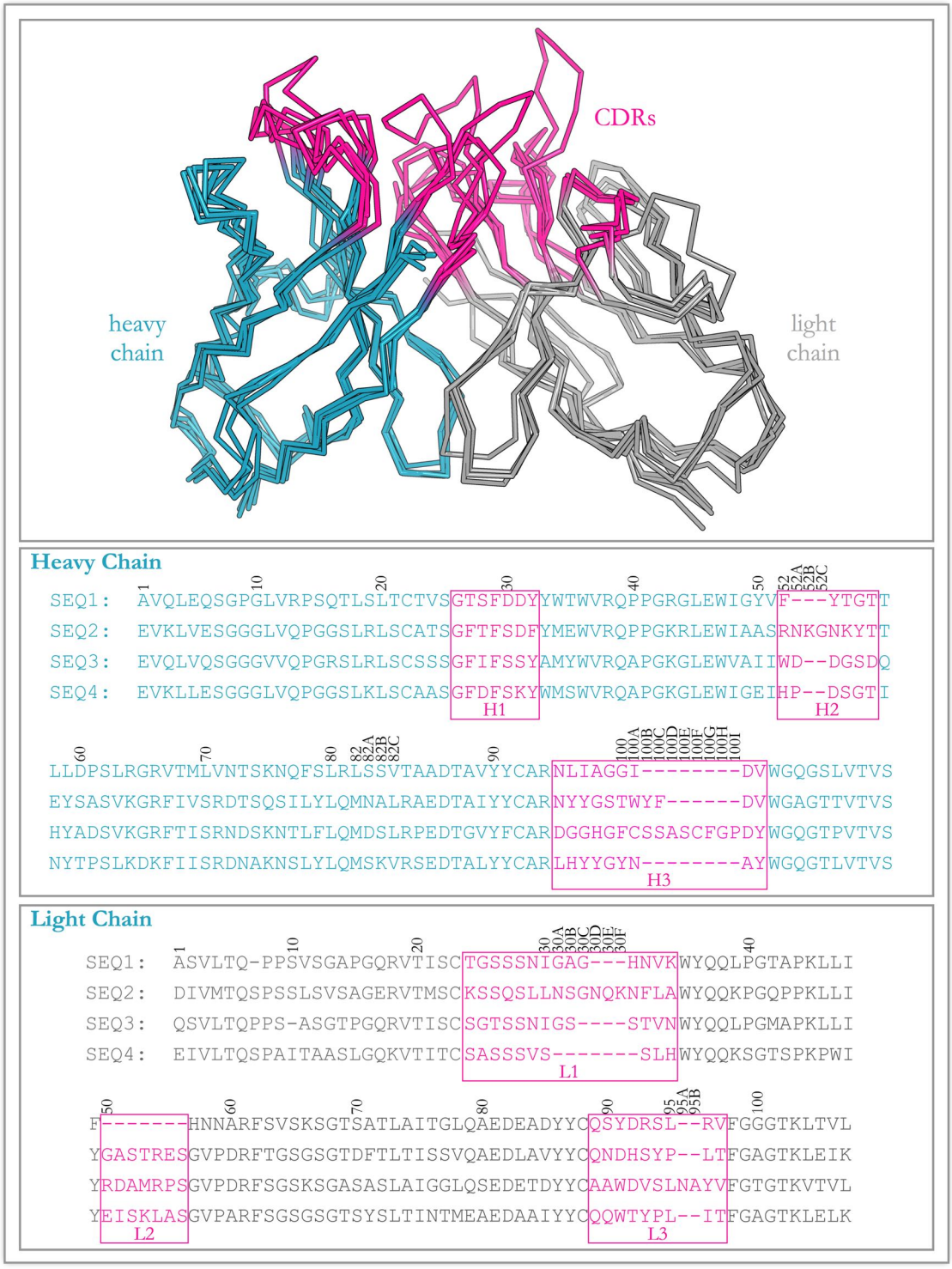

Figure 1.12: Numbering of antibody sequences using the Chothia system. By superimposing several antibody variable domains, it can be seen which parts are structurally conserved (shown in blue and grey for the heavy and light chains, respectively), and which are more diverse (pink). This leads to the CDR definitions as shown.

which residues should be mutated, through the use of computational techniques such as binding affinity prediction (*e.g.* Sirin *et al.*, 2015), epitope and paratope prediction (*e.g.* Ren *et al.*, 2014; Krawczyk *et al.*, 2013), stability measurements (*e.g.* Rouet *et al.*, 2014), and docking (*e.g.* Pedotti *et al.*, 2011). Computational tools have already been used successfully, allowing the binding affinity of mature antibodies to be increased (*e.g.* Clark *et al.*, 2006; Lippow *et al.*, 2007; Diskin *et al.*, 2011; Kiyoshi *et al.*, 2014; Thakkar *et al.*, 2014). Antibodies have even been designed that are bispecific, with a different binding site on each of the two Fv regions (*e.g.* Lewis *et al.*, 2014).

Unfortunately, as explained in Section 1.1.6, experimental structure determination is often time-consuming and expensive. In addition, the flexibility of protein loops often means that the electron density obtained by X-ray crystallography is missing or of poor quality for these regions (Petoukhov *et al.*, 2002). When one also considers that a single residue change in a protein loop may change its structure (Schaefer and Rost, 2012), we see why X-ray crystallography is not currently a routine part of antibody design. The ability to build accurate models of antibody structures, especially their CDR loops, from their sequences is therefore highly desirable.

The current accuracy of antibody structure prediction is monitored through a CASP-style (Moult *et al.*, 1995) blind prediction test called the Antibody Modelling Assessment (or AMA). The first AMA was conducted in 2011 (Almagro *et al.*, 2011) — participants are given the sequences of several unpublished, high-resolution antibody structures and asked to model them. More recently, the results of the second assessment (AMA-II) were published (Teplyakov *et al.*, 2014). There were two rounds in this study: the first entailed modelling the entire Fv region from its sequence; in the second, the accuracy of H3 structure prediction was tested by giving the participants the native structures of the Fv regions with the H3 loop residues missing.

Framework Prediction

Since the framework is highly conserved between antibodies, the framework can be successfully modelled using a homology approach (see Section 1.1.7). After annotating the target antibody sequence using a numbering scheme (Section 1.3.5), alignment of

that sequence to those of known structure allows a template to be selected, based on sequence identity. Separate templates may be chosen for the VH and VL, however If a template is available with high sequence identity to both the target VH and VL, only one is required. Usually, modelling algorithms use a sequence identity cutoff to decide when to use separate templates — when a structure is available for the whole Fv with a sequence identity over a particular value, only one template is used, otherwise separate templates are chosen, selecting the structure with the most similar sequence in each case. Normally, the framework of the variable region can be modelled with an RMSD below 1 Å — in the Antibody Modelling Assessment II, predictions were made with an average accuracy of 0.65 Å and 0.5 Å for the VH and VL respectively (Teplyakov *et al.*, 2014).

One of the challenges of modelling the framework structure is predicting the relative orientation of the VH and VL domains. Significant differences can be observed between different antibody structures (Dunbar *et al.*, 2013); it is important to model the orientation correctly since it can markedly change the shape of the binding site (Abhinandan and Martin, 2010). During modelling, if just one template is selected for the whole Fv, then the orientation can just be copied across from this structure. However, in the case of using separate templates for the VH and VL, other procedures must be used. These include superimposing the VH and VL models onto a third template, iterative optimisation, use of an energy function, and machine learning techniques. The VH/VL orientation has been characterised and several key residues identified in the framework that influence it — by comparing to the equivalent residues in the target sequence, prediction accuracy can be improved (Abhinandan and Martin, 2010; Dunbar *et al.*, 2013; Bujotzek *et al.*, 2015).

Canonical CDR Prediction

Modelling of the canonical CDRs (L1, L2, L3, H1 and H2) tends to rely upon the prediction of which canonical class they belong to. An exception to this is ABodyBuilder (Leem *et al.*, 2016), which uses FREAD to model the CDR loops without classifying them first (Choi and Deane, 2011). Key residues have been identified that influence which canonical form a given loop will belong to; this allows prediction from sequence

(Nowak *et al.*, 2016). If the CDR of the template has the same canonical class as that predicted for the target, then sometimes this structure is retained, reducing errors in the anchor regions and meaning no further modelling is required (*e.g.* Fasnacht *et al.*, 2014). Otherwise, templates for each loop are selected in the same way as for the framework; loop structures that share similar sequences to the target and that belong to the same canonical class are selected and grafted onto the framework. The existence of canonical classes means that prediction accuracy for the five canonical CDRs is often high; in the second Antibody Modelling Assessment, they were modelled with average RMSDs of 1.2 Å, 0.5 Å, 1.6 Å, 1.1 Å, and 1.0 Å for L1, L2, L3, H1 and H2 respectively (Teplyakov *et al.*, 2014).

H3 Prediction

Due to its structural diversity, structure prediction of the H3 loop is challenging. However, it is possibly the most important part of the structure to model correctly, since it is mainly responsible for the antigen-binding properties of an antibody (Kuroda *et al.*, 2012). While some algorithms exist that do not treat it any differently to the other CDRs (*e.g.* ABGEN, Mandal *et al.* 1996), this is not usual and it normally requires a special approach, using a knowledge-based or *ab initio* approach, or some combination of the two. Some algorithms have tried to use the presence of a kinked or extended conformation to guide H3 loop modelling, by using a series of rules to predict which conformation is adopted from sequence. This information can then be used to either pre-filter a database of solved structures in the case of knowledge-based methods, or limit the conformational search of an *ab initio* algorithm.

In the second round of AMA-II, the accuracy of H3 structure prediction was tested by giving the participants the native structures of the Fv regions with the H3 loop residues missing. This second round was added so that any errors introduced into the H3 model caused by inaccuracies in the framework structure could be eliminated, giving an impression of the current accuracy of H3 prediction — we use this approach to evaluate the performance of the algorithms described in this thesis. Each group

was required to submit five predictions for each of ten H3 targets, with loop lengths ranging from 8 to 14 residues.

The group that achieved the best results was Schrödinger, using the commercial Prime software. The loop modelling algorithm is freely available under the name PLOP (Jacobson *et al.*, 2004, see Section 1.2.3). For eight of the ten targets, Prime produced the most accurate model. However, as is the case for all the groups, once all five of the predictions are taken into account instead of only the best, average RMSDs become far worse. There are several possible reasons for this: the set of loop models generated may only contain a couple of good models; or the ranking system used to select good models is inadequate. Alternatively, the five predictions for each target may have been purposefully chosen so that they cover a larger conformational space between them, preventing the submission of five very similar but incorrect models. This indicates that the submitters believe the ranking method used cannot consistently choose the best conformations.

The results obtained during AMA-II, as well as some other H3 prediction studies, are shown in Table 1.2. Reasonable accuracies are currently being achieved for short H3 loops (up to around 9 residues), but predictions become far worse for loops beyond that length. There is an appreciable difference in accuracies achieved modelling H3 loops compared to the other CDRs. For example, the knowledge-based method FREAD has been shown to produce sub-Ångström predictions for the five canonical CDRs (0.81 Å, 0.42 Å, 0.96 Å, 0.98 Å and 0.88 Å for L1-H2), while the average accuracy for H3 loops is 2.25 Å (Choi and Deane, 2011). RosettaAntibody also produces sub-Ångström predictions for the other CDR loops (0.78 Å, 0.54 Å, 0.81 Å, 0.84 Å and 0.93 Å), while the accuracy of H3 prediction ranges between 1.6 Å and 6.0 Å depending on length.

Examples of H3 Prediction Algorithms

The accuracy achieved by some loop modelling algorithms is given in Table 1.2. The algorithms are described in more detail below.

ABGEN

ABGEN (the AntiBody structure GENeration algorithm) is an antibody modelling tool published by Mandal *et al.* (1996). There are two parts to the algorithm: ABalign, which

Table 1.2: Reported accuracies for H3 prediction achieved by some loop modelling algorithms. Values given are average global RMSDs, and the values in brackets after each RMSD are the number of loops predicted followed by their length (in residues). Some target sets are the same, indicated by a * or † symbol. RMSDs quoted for the AMA-II target set (denoted by *) are carbonyl RMSDs, *i.e.* calculated over the C and O atoms of the backbone only.

Algorithm	Type	Key RMSD Results			Reference
ABGEN	Knowledge-Based	2.3 Å (15, 5-17, models) 1.9 Å for up to 10 residues 3.0 Å for over 10 residues			Mandal *et al.*, 1996
Accelrys Tools	Combination	Best of Top 3 = 3.14 Å; Average of Top 3 = 3.88 Å (11, 8-14, models)* Best of Top 5 = 1.86 Å; Average of Top 5 = 2.89 Å (10, 8-14, crystal)*			Fasnacht *et al.*, 2014
CCG (MOE)	Knowledge-Based	Best of Top 3 = 2.86 Å; Average of Top 3 = 3.69 Å (11, 8-14, models)* Best of Top 5 = 2.09 Å; Average of Top 5 = 3.08 Å (10, 8-14, crystal)*			Maier and Labute, 2014
FREAD	Knowledge-Based	2.25 Å (97, 3-19) coverage = 100%			Choi and Deane, 2011
ConFREAD	Knowledge-Based	1.23 Å (97, 3-19) coverage = 70%			Choi and Deane, 2011
H3Loopred	Knowledge-Based	Model environment: 1.3 Å (3, 4-6)† 1.6 Å (22, 7-9)†	1.8 Å (14, 10-11)† 3.3 Å (10, 12-14)†	7.1 Å (4, 17-22)†	Messih *et al.*, 2014
KotaiAntibodyBuilder	Combination	Best of Top 3 = 2.41 Å; Average of Top 3 = 3.02 Å (11, 8-14, models)* Best of Top 5 = 1.25 Å; Average of Top 5 = 2.43 Å (10, 8-14, crystal)*			Shirai *et al.*, 2014
Prime/PLOP	*Ab initio*	Best of Top 3 = 2.74 Å; Average of Top 3 = 3.60 Å (11, 8-14, models)* Best of Top 5 = 1.12 Å; Average of Top 5 = 2.54 Å (10, 8-14, crystal)*			Zhu *et al.*, 2014
		Crystal environment: 0.18 Å (3, 4-6)† 0.70 Å (22, 7-9)†	0.67 Å (14, 10-11)† 2.38 Å (10, 12-14)†	3.63 Å (4, 17-22)†	Zhu and Day, 2013
RosettaAntibody	*Ab initio*	Best of Top 3 = 2.66 Å; Average of Top 3 = 3.11 Å (11, 8-14, models)* Best of Top 5 = 1.97 Å; Average of Top 5 = 3.22 Å (10, 8-14, crystal)*			Weitzner *et al.*, 2014
		Crystal environment: 1.6 Å (3, 4-6)† 1.9 Å (22, 7-9)† Model environment: 1.4 Å (3, 4-6)† 2.2 Å (22, 7-9)†	2.4 Å (14, 10-11)† 3.1 Å (10, 12-14)† 2.9 Å (14, 10-11)† 3.5 Å (10, 12-14)†	6.0 Å (4, 17-22)† 7.6 Å (4, 17-22)†	Sivasubramanian *et al.*, 2009
SmrtMolAntibody	*Ab initio*	Best of Top 3 = 3.02 Å; Average of Top 5 = 3.71 Å (11, 8-14, models)* Best of Top 5 = 2.41 Å; Average of Top 5 = 3.08 Å (10, 8-14, crystal)*			Berrondo *et al.*, 2014
WAM	Combination	≤1.7 Å for 9 out of 11 loops under 10 residues 1.3 - 2.7 Å for loops of 10 residues or more (8, 10-12)			Whitelegg and Rees, 2000

selects a template structure for each part of the structure (*i.e.* framework and CDRs) by sequence similarity; and ABbuild, which is responsible for generating the three-dimensional structure. The CDRs are modelled using a knowledge-based approach, and the H3 loop is not treated any differently — candidate templates are found from known antibody structures and selected based on sequence and length. If no loop exists of the same length, then the closest is selected. The loops are grafted onto the framework structure by superimposing the anchor residues (the two residues on either side of

the loop). Residue mis-matches between the template and target are then dealt with by replacing the sidechains. Clashes are avoided by iteratively changing the sidechain torsion angles. Prediction of the whole antibody structure is reported to take around 5 minutes on an SGI Elan Indigo desktop (Mandal *et al.*, 1996).

Accelrys Tools

Accelrys is a software company that has produced an antibody prediction tool for commercial use. Its performance was evaluated during AMA-II (Fasnacht *et al.*, 2014). Three different methods are used to predict the H3 loop:

1. a purely knowledge-based approach — templates are selected from a database like they are for the other CDRs, except the kinked/extended conformation is also considered.

2. like method 1, with additional re-modelling of the most variable part of the loop using an *ab initio* approach. The section of the loop to be remodelled is chosen by eye.

3. like method 2, but with additional sidechain refinement before the *ab initio* modelling — the sidechains of the H3 loop are mutated to alanine while those of the rest of the structure are refined.

During the second round of AMA-II (H3 modelling onto the native antibody structure), method 2 was used. The final decoy selection is carried out based on clustering — all conformations are grouped by structural similarity, and the clusters are ranked according to the energy of its members (calculated using a physics-based energy function). The lowest energy model from the top-ranked cluster is given as the final prediction. On average, the algorithm takes 30 minutes on a standard desktop machine to produce a prediction (Fasnacht *et al.*, 2014).

CCG (MOE)

The protocol used by the CCG group (part of the commercial MOE software) is a knowledge-based algorithm, used in conjunction with molecular dynamics (Maier and Labute, 2014). Its performance was evaluated during AMA-II. Known H3 structures are scored based on backbone topology, bond lengths and angles, probability of ϕ/ψ angles, crystallographic occupancies, and temperature factors. After clustering, the member of the cluster with the highest score is put into a database. This database is enriched by running MD simulations on these structures. Possible structures are selected from the database depending on anchor RMSD, using a tight cutoff of 0.25 Å, and a final prediction is made using a score that takes into account H3-specific properties, such as surface-accessible surface area, ϕ/ψ angles, and the interaction of the loop to the rest of the Fv. During the AMA-II, the production of each full antibody model took around 30 minutes (Maier and Labute, 2014).

FREAD and ConFREAD

FREAD has been previously described in Section 1.2.3. Research into improving FREAD's ability to predict H3 loops led to a new version with an additional filter that considers the contact profiles of the fragments within the database (Choi and Deane, 2011). Each residue of each fragment in the database is annotated with a number depending on the contacts it forms in its native environment: 0 for no contacts; 1 for external contacts (those to another chain); 2 for internal contacts (formed with other residues on the same chain); and 3 for both internal and external contacts. The actual contact profile of the fragment is then compared to its profile when inserted into the target structure — only fragments with matching pairs are retained. The final prediction was chosen in the same way as in the original FREAD algorithm (see Section 1.2.3). While this led to an increase in prediction accuracy (from 2.25 Å to 1.23 Å), coverage (the proportion of targets for which the algorithm could produce a prediction) was significantly lower (reduced to 70% from 100%).

H3Loopred

H3Loopred (Messih *et al.*, 2014) is similar to LoopIng, which was described in Section 1.2.3. Again, a Random Forest model is developed that uses several features to predict the similarity of a known loop structure to the structure of the target, using a measure called the TM-score (Zhang and Skolnick, 2004a). The features used are a mixture of general and H3-specific properties: loop sequence, the canonical classes and lengths of the other CDRs in the antibody, source organism, germline family, and the similarity scores for each residue and the whole loop. If the structure from the database that is predicted to be the best has a predicted TM-score of less than 0.5, then this loop is returned as the final prediction. Otherwise, the top 50 templates are ranked using a score that considers contacts. The average computation time required is five minutes per target on a single 2.5 GHz CPU with 8 GB of RAM (Messih *et al.*, 2014).

Kotai Antibody Builder

Kotai Antibody Builder is a simplified and automatic version of the software used in AMA-II by the joint Osaka University Astellas (JOA) team (Shirai *et al.*, 2014; Yamashita *et al.*, 2014). In the second round of AMA-II, H3 decoys were generated using a combination of a knowledge-based approach with molecular dynamics simulations. Spanner (the knowledge-based algorithm) selects fragments from a database using sequence similarity, secondary structure similarity, a clash score and the geometry of the anchor residues. Minimisation of these structures is carried out using the OSCAR-loop energy function (Liang *et al.*, 2014), and the top 20 structures are used as initial conformations for a series of MD simulations. Snapshots of the simulations are then grouped into five clusters, with the final set of predictions including one structure from each.

BioLuminate and Prime

BioLuminate and Prime are software packages produced by the Schrödinger company; their performance was evaluated in the AMA-II (Zhu *et al.*, 2014). Prime is the commercial version of the loop modelling algorithm PLOP (see Section 1.2.3). BioLuminate models antibodies using homology; CDRs (including H3) are modelled by selecting

templates from a database. Prime is an *ab initio* loop modelling algorithm. For stage 1 of AMA-II, where H3 predictions were made onto model frameworks, the three submitted models were generated in different ways: a straightforward template selection based on sequence similarity; template selection after clustering known H3 structures, taking the structure with the highest sequence similarity from the largest cluster; and *ab initio* prediction using Prime. In the second round, the *ab initio* approach was used exclusively, but the target loop was extended by one residue on each side to make the terminal residues flexible.

RosettaAntibody

RosettaAntibody models the H3 loop using an *ab initio* approach — the details of this algorithm are described in Section 1.2.3. RosettaAntibody was one of the algorithms used during AMA-II (Weitzner *et al.*, 2014). For the first stage, models were generated using next-generation KIC without using any neighbour dependence during ϕ/ψ sampling. In stage 2, both this approach and 'legacy KIC' were used (the version published by Mandell *et al.* 2009), and for those targets predicted to have a kinked conformation, constraints were added to enforce it.

SmrtMolAntibody

SmrtMolAntibody, the commercial antibody modelling software from Macromoltek, was also tested during AMA-II (Berrondo *et al.*, 2014). An *ab initio* approach is used to model the H3 loop. Firstly, the first and final three residues of the loop are modelled according to their predicted kinked/extended conformation. The remaining residues are then added as dimers, where the ϕ/ψ angles of the two residues have been observed together in nature. After all decoys are generated, the structures are filtered, by checking each trimer for non-physical neighbouring dihedral angles, and finally ranked using a statistical potential. The reported time required to produce a full antibody model is 30 minutes, using a single Intel i7 CPU on a Linux server (Berrondo *et al.*, 2014).

WAM

WAM uses different approaches to model H3 loops depending on their length (Whitelegg and Rees, 2000). If the loop is shorter than eight residues, then a traditional knowledge-based algorithm is used. Specific databases are used depending on whether the loop is predicted to have a kinked or extended conformation. For loops of eight residues or more, the database search is followed by the remodelling of the middle five residues of the loop using an *ab initio* method called CONGEN. CONGEN produces decoys by calculating ϕ/ψ angles that form a closed structure, using the work of Go and Scheraga (1970). The decoys undergo minimisation, and are clustered to remove duplicate conformations. The final prediction is selected from the pool of decoys using a score that considers surface accessibility, the RMSD of the decoy to known kinked H3 structures, and the calculated energy (using a physics-based energy function).

Summary

As for the general loop modelling problem, H3 structure prediction has been attempted using knowledge-based, *ab initio* and hybrid methodologies. Prediction accuracy is generally worse than for general loop types. Nearly all of the algorithms described in this section have been written specifically for the prediction of H3 loops (*e.g.* Accelrys Tools, SmrtMolAntibody, etc.), or are adapted versions of general loop prediction software (*e.g.* FREAD, H3Loopred). The identification of the C-terminal kink present in many H3 structures has meant that more hybrid methods have been used for H3 prediction than for general loop types. An approach often used is to model the 'torso' of the loop (the kinked/extended region) using a knowledge-based algorithm, then complete the rest of the loop using an *ab initio* method (*e.g.* Accelrys Tools, SmrtMolAntibody, WAM). Alternatively, the predicted conformation of the loop torso is used to guide a database search (*e.g.* approach 1 from Accelrys Tools), or constraints are placed on the loop structure to enforce a kinked conformation (*e.g.* RosettaAntibody).

1.4 Thesis Overview

The aim of the work presented in this thesis is to improve the accuracy of H3 loop structure prediction. The remaining chapters of this work are organised as described below.

Chapter 2

Using the knowledge-based method that was developed in our group, FREAD, in this chapter we investigate the effect of various fragment databases on prediction accuracy. We also look at whether results could be improved by changing certain cutoff filters from their default values, and investigate whether decoy selection could be improved through the use of a statistical potential.

Chapter 3

In this chapter, we describe the development of a new *ab initio* loop modelling algorithm, MECHANO, which is specifically designed to model antibody H3 loops. By including H3 data (generated using a novel resampling method), a restricted loop closure algorithm, and the ability of loop building to occur from either anchor residue, the ability of the algorithm to produce near-native conformations is improved.

Chapter 4

In this chapter we describe Sphinx, a novel method that combines knowledge-based and *ab initio* approaches in a more integrated way than previously considered. By using fragments of a different length to the target loop and completing the structure using *ab initio* techniques, Sphinx is able to produce decoy sets enriched with near-native conformations. Unlike other methods, which must use length-matched fragments, Sphinx can use fragments that are shorter than the target — this is useful, since it has been shown that loops can have similar conformations even if they are different lengths (Nowak *et al.*, 2016).

Chapter 5

The last step in a loop modelling algorithm is the selection of a final prediction, through the use of some kind of ranking system. In this chapter, we test a variety of ranking methods (*e.g.* statistical potentials, clustering, and minimisation) and evaluate their ability to select near-native fragments. We incorporate the best procedure into Sphinx, and show that it outperforms some leading H3 prediction software.

Chapter 6

Loop structures are sometimes flexible; *i.e.* they may be able to adopt a number of different conformations. In this chapter, we investigate occurrences of loops with identical sequences having different structures. We show that loops that show evidence of flexibility are more difficult to model accurately than rigid loops, and investigate why this may be the case.

Chapter 7

In this chapter, we form our conclusions and summarise the key findings of this DPhil project. We also discuss potential areas for further research that follow from the work presented here.

Pre-Chapter Image: The membrane protein maltoporin, which has loops in its β-barrel structures that assist in the transport of sugars into the cell.

2

FREAD: Modelling H3 loops using a knowledge-based approach

Contents

2.1 Introduction

Knowledge-based, or database loop prediction algorithms, use structures that have previously been observed to generate target loop conformations. Generally, these methods search databases of protein fragments for conformations likely to be structurally similar to the target. Structures from the database are chosen based on particular features

that may indicate the database fragment is close in conformation to the target. Features that are commonly used include sequence similarity and the geometry of anchor residues. Methods of this type are often very fast, and can be very accurate when the structure of the target loop is close to one that has been observed previously.

In this chapter, we investigate how well the non-H3-specific knowledge-based method FREAD performs when predicting H3 loop structures, and look at possibilities for improvement. We have selected FREAD because it was developed within our research group, meaning we have full control and can make any changes we wish to, and because it has previously been shown to produce excellent results when modelling the loops of general soluble proteins (Deane and Blundell, 2001; Choi and Deane, 2010) — in a benchmarking study, FREAD produced better predictions than the *ab initio* algorithms MODELLER, PLOP and RAPPER, with the majority of predictions having an RMSD of below 2 Å for loops of up to 20 residues in length (Choi and Deane, 2010).

In addition to its superior performance on general loop types, FREAD has also been used successfully to predict the structures of specific types of loops such as those of membrane proteins (Kelm *et al.*, 2013), and antibody CDRs (Choi and Deane, 2011). In particular, Choi and Deane (2011) focussed on H3 loop prediction, and produced two modified versions of the algorithm that improved accuracy; one that placed greater emphasis on sequence similarity (FREAD-S) and another that included contact information (ConFREAD, see Chapter 1). However, these modifications meant that there were more targets for which FREAD failed to make a prediction. Instead, we attempt to improve prediction accuracy by investigating the choice of fragment database and the effect of changing the default cutoffs that FREAD uses to select fragments.

2.2 Method

2.2.1 The FREAD algorithm

As input, FREAD takes a protein structure with a missing loop segment, together with the sequence of the loop. A database is searched, containing the structures of a set of

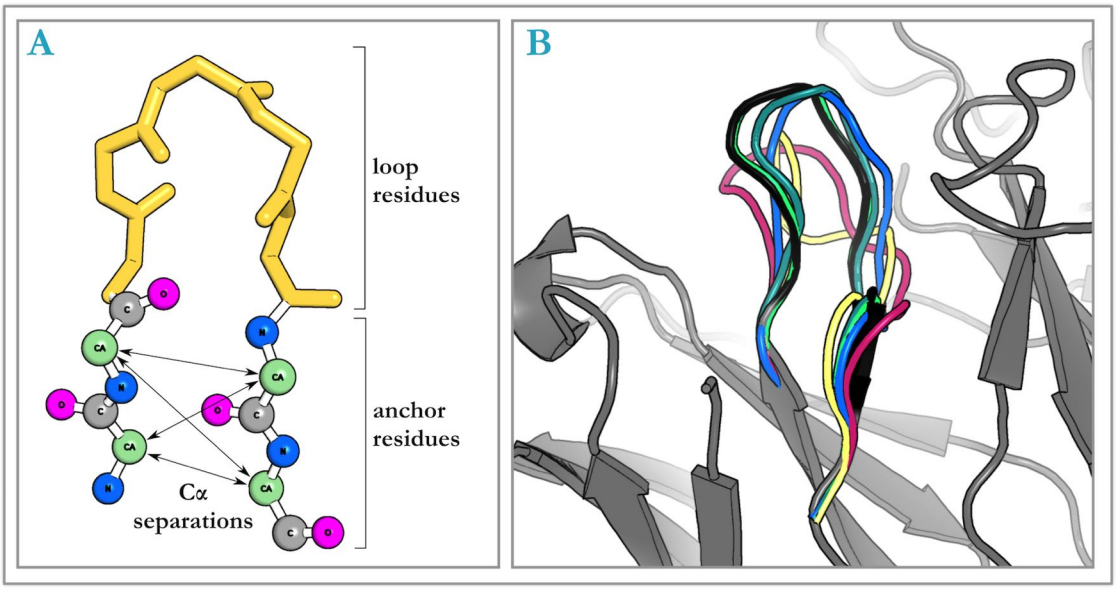

Figure 2.1: A — Diagram of the anchor residues (shown as backbone only in a ball-and-stick representation) and the loop residues (shown in yellow). The anchor residues are the two residues on the N-terminal side (N-anchor) and the two on the C-terminal side (C-anchor) of the target loop. The Cα separations used by FREAD to determine the suitability of a fragment are indicated. B — The top five predictions generated by FREAD for the H3 loop of the antibody 1S3K. The native loop structure is also shown, in black.

proteins from which all possible fragments can be used. Initially, the algorithm extracts the structures of the anchor residues (the two residues on either side of the loop), and the database is quickly screened to find candidate fragments. This computationally fast preliminary assessment filters out potential fragments based on two features:

- Length — the fragment must have the same number of residues as the target loop;
- The Cα separations of the two residues on either side of the loop (the anchor residues — see Figure 2.1A) and the Cα separation of the equivalent residues in each database fragment. If the Cα separations are similar, this is an indication that the fragment is of the correct shape to fit into the gap present in the input structure.

Fragments that meet these conditions are then filtered again, using three further factors that indicate how structurally similar the fragment from the database is to the target loop:

- The similarity between the sequence of the target loop and the fragment, evaluated using an environment-specific substitution score;

- Any clashes when the database fragment is positioned into the target protein;
- The backbone RMSD between the anchor residues of the target and those of the database fragment. Initially in this work, we use the default cutoff value of 1 Å.

The fragments from the database that remain after filtering out all those that are not suitable are incorporated into the input structure via a 'melding' method. This involves taking an average of the coordinates of the fragment anchor residues and those of the target anchor residues, to try and ease any strain that would result from inserting a non-native loop structure. The average is weighted (using the value D in the equation below) so that the anchor residue closest to the loop is more like the fragment's anchors, and the anchor residue closest to the remaining protein more closely resembles the target structure. The melded coordinates are calculated using the following equation:

$$M_i = \frac{1}{3}(DP_i + (3-D)N_i)$$

where M_i is the melded coordinate, D is 1 if the anchor residue in question is adjacent to the loop and 2 if it is one residue away, and P_i and N_I are the coordinates of the atom in the fragment structure and target structure respectively. Decoys are subsequently ranked by the backbone RMSD of their anchors when compared to the anchor residues of the target. The top-ranked loop (with the lowest anchor RMSD) is taken as the final prediction (referred to henceforth as the 'top prediction'). An example set of loop decoys produced by FREAD is shown in Figure 2.1B.

Environment-Specific Substitution Scores

The similarity between two sequences is measured using a substitution score. A substitution score is a measure of how likely it is that one type of amino acid will mutate into another type. It is more probable that residues will mutate into amino acid types with similar properties — for example, a hydrophilic residue such as lysine is more likely to mutate into arginine, another hydrophilic amino acid, than the hydrophobic isoleucine. Given a sequence alignment, the total score for one sequence mutating into the other can be calculated by adding together the scores for each position. This indicates how

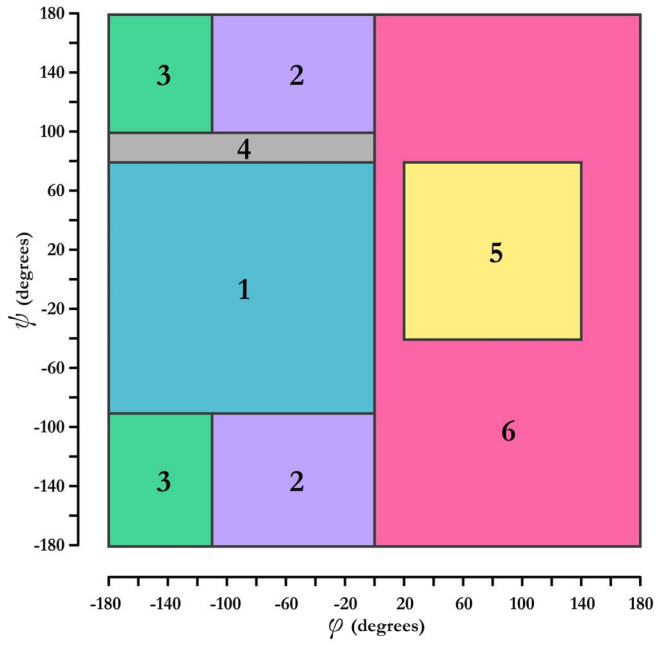

Figure 2.2: Regions of the Ramachandran plot that define the six environments used in the environment-specific substitution scores.

closely related two structures are. Score tables are usually derived from sets of aligned sequences, calculated based on the frequency of substitutions observed and the number expected. The scores are commonly given as log-odds scores.

Environment-specific substitution scores (ESSS), such as those used in FREAD, also take into account the 'environment' of the mutating residue — *i.e.* the probability that a residue with a particular property is substituted by another amino acid type. FREAD defines six environments that depend on the dihedral angles (ϕ and ψ) of the residue, with the Ramachandran plot partitioned into six regions (Figure 2.2). The environment of the second residue does not need to be known, allowing the fragment structure and sequence to be compared to the target sequence, whose structure is obviously unknown. In this way, structural information can be incorporated into the scoring system.

The score for amino acid a in environment E being replaced by amino acid b is given by:

$$s(a, E \rightarrow b) = \log\left(\frac{P(b|a,E)}{q_b}\right)$$

where $P(b|a,E)$ is the probability that amino acid a in environment E is substituted by amino acid b:

$$P(b|a,E) = A^E_{ab} / \sum_c A^E_{ac}$$

where $\sum_c$ is a sum over all 20 amino acid types, and q_b is the background probability of observing amino acid b:

$$q_b = \sum_{a,E} A^E_{ab} / \sum_{a,b,E} A^E_{ab}$$

A^E_{ab} is the number of times amino acid a in environment E is observed being replaced by amino acid b. Initially, we have used the default parameters of FREAD and therefore a minimum ESSS of 25.

2.2.2 Fragment Databases

There are two main aspects of the FREAD algorithm that can be changed when considering its use on a specific protein type: the environment-specific substitution table from which the ESS scores are calculated, and the contents of the database of protein fragments that is searched. In their 2013 work, Kelm *et al.* showed that the results achieved by FREAD are more dependent on the database. When modelling a membrane protein loop, better results are achieved using a database of membrane protein fragments instead of soluble protein fragments, but little difference is observed upon changing the ESSS tables to be membrane protein-specific (Kelm *et al.*, 2013). We decided, therefore, to test a number of different protein fragment databases to investigate their effect on prediction accuracy when used for H3 loops. Four databases were created from the protein structures available at the time this research was carried out (August 2013):

1. All antibodies in the Structural Antibody Database (SAbDab) (Dunbar *et al.*, 2014) (6007 chains);

2. A non-redundant set of antibodies from SAbDab, using a 90% sequence identity cutoff (1056 chains);

3. A non-redundant set of all proteins in the PDB (Berman *et al.*, 2000) excluding antibody structures, with resolution below 3 Å and a 90% sequence identity cutoff (23,210 chains);

4. Antiparallel β-sheet loops — loops between residues defined as β-strands by DSSP (Joosten *et al.*, 2011), from all proteins in the PDB (excluding antibody structures and duplicate chains, leading to a database containing loops from 59,607 chains). The five residues from either side of the loop are also included.

The different databases were chosen to see whether using antibody structures produces better results than general loops, and whether removing redundant structures makes any difference to prediction accuracy. We chose to test the antiparallel β-sheet loop database because H3 loops themselves are loops connecting two antiparallel β-strands. All non-redundant sets were generated using PISCES (Wang and Dunbrack, 2003). With the exception of the one containing antiparallel β-sheets, the databases contained all possible fragments of the proteins (from 3 to 30 residues in length), not just the loop regions. Any fragments with missing residues were discarded.

Since the two databases of antibody fragments may contain the true answer, self-prediction may occur. Therefore any predicted loop from a chain identical in sequence to that of the query chain is ignored. This means that loop decoys from both the bound and unbound forms of the antibody (if both are present) are omitted.

2.2.3 Test set

The complete SAbDab dataset contained 1678 antibody structures as downloaded in August 2013. Of these, 1465 have at least one complete H3 loop (*i.e.* without gaps in the structure). In this project we considered only those structures which have H3 loops with lengths between 3 and 19 residues — a total of 1392 structures, encompassing 83% of the total structures in SAbDab (the length distribution of H3 loops is shown in Figure 2.3). We used FREAD to predict the H3 conformations for half of the examples at each loop length (giving a large number of test cases). There are at least 10 target loops per length; in total the test set contains 691 loops (see Fig 2.3). If a SAbDab entry contained

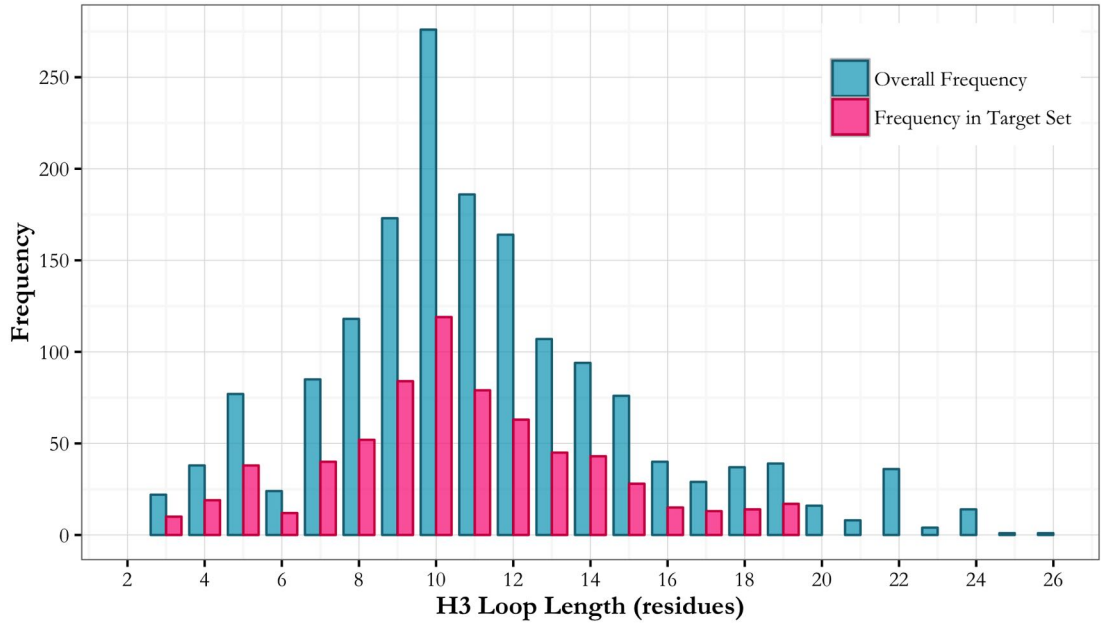

Figure 2.3: The length distribution for antibody H3 loops in the PDB. Most H3 loops are 10 residues long, but the range of observed lengths is large. There are few examples of very short (below 4 residues) and very long (over 20 residues) loops. The number of loops of each length in the target set used to test FREAD is also shown.

multiple copies of the H3 loop (*e.g.* two copies of the VH domain), one was chosen at random. The structures used as the input to FREAD are the Chothia-renumbered versions downloaded from SAbDab, so the H3 loop regions are defined as the residues numbered from 95 to 102 (Chothia and Lesk, 1987).

2.3 Results and Discussion

Throughout this section, the loop decoy that results in the lowest RMSD is referred to as the 'best' prediction, while the one that FREAD predicts to be the most accurate (*i.e.* lowest RMSD) is known as the 'top' prediction. The top and best predictions are not always the same due to imperfections in the ranking system. RMSDs reported are calculated over the backbone atoms of the loop (N, Cα, C and O) after superimposing the anchor residues, unless otherwise stated.

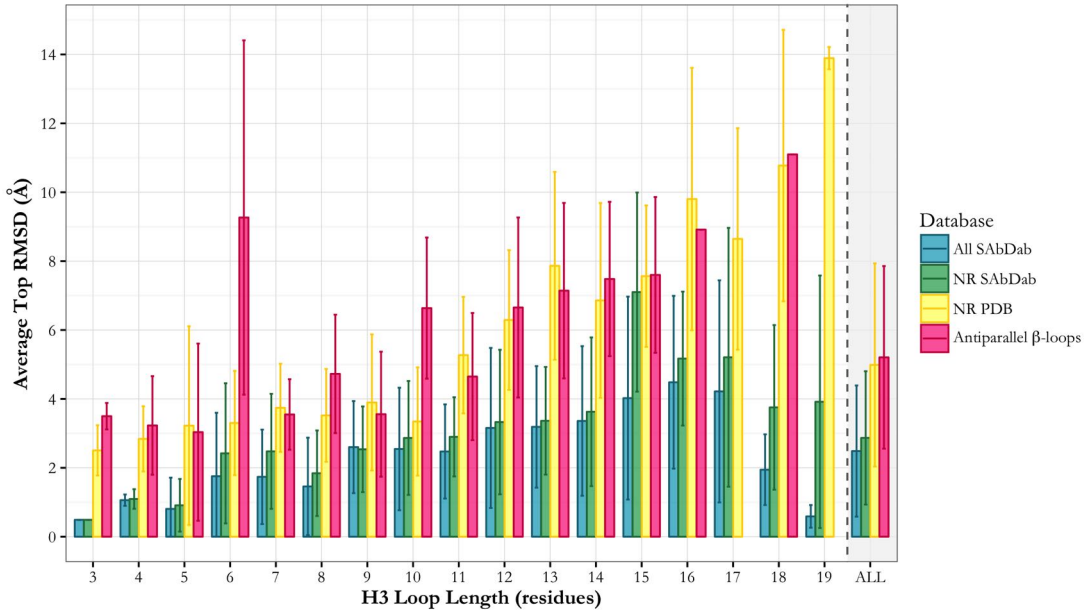

Figure 2.4: The average RMSD of the top predictions for the H3 targets at each loop length, achieved by each of the four databases described in Section 2.2.2. Error bars show the standard deviation. The All SAbDab database generated the best predictions, followed by the NR SAbDab database, NR PDB database and finally the β-sheet database.

2.3.1 Effect of the fragment database

FREAD was used to predict the structures of 691 H3 loops using the four databases described in section 2.2.2. The average RMSDs of the top-ranked predictions for each loop length are shown in Figure 2.4. It can be seen that the database giving the most accurate predictions is the one containing all antibody structures ('All SAbDab'). The average global RMSD is below 2 Å for H3 loops of up to 8 residues. Accuracy is lower at longer lengths, and the maximum is 4.48 Å for loops of 16 residues. The predictions for 19-residue loops are very accurate, more so than would be expected; this could be because the H3 loop structures that are available for this length are not as structurally diverse as for other lengths, making them easier to predict. This is likely, as there are relatively few examples of H3 loops that are 19 residues long (see Figure 2.3).

The database containing a non-redundant set of antibody structures ('NR SAbDab') produces less accurate predictions than the All SAbDab database; this was expected because this database does not contain any different structures from the All SAbDab database, just fewer of them. For 9-residue H3 loops, more accurate results are produced by the NR SAbDab database than the All SAbDab database. This is due to a failure in

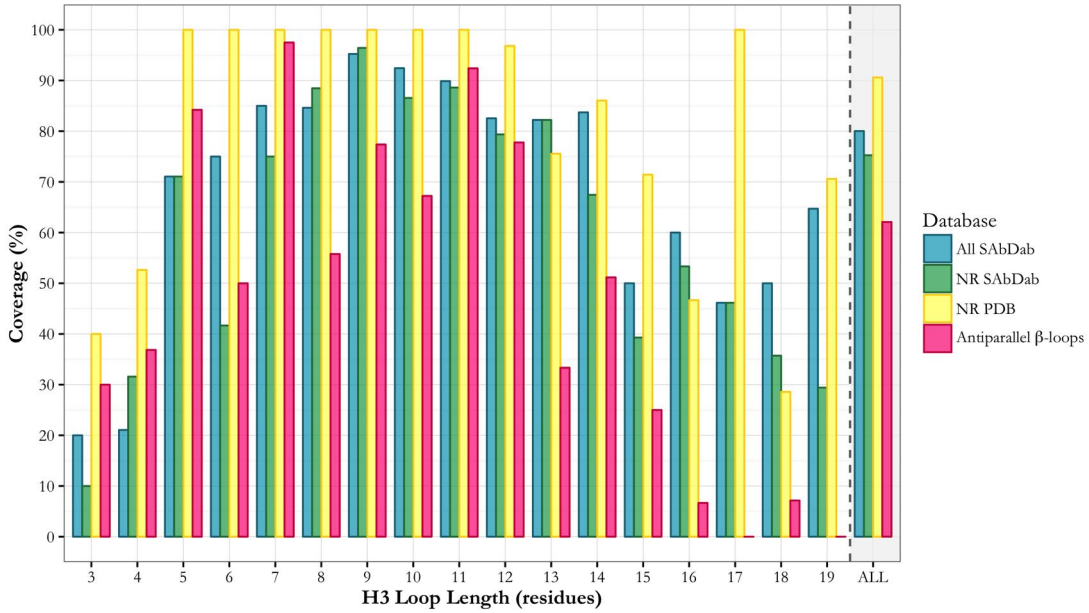

Figure 2.5: The coverage of each of the four tested databases described in Section 2.2.2 (*i.e.* the percentage of times FREAD was able to find a suitable fragment in the database). The best database with regards to coverage is the NR PDB database. The antiparallel β-sheet database is the worst.

the decoy ranking system when using the All SAbDab database — a fragment with high RMSD is predicted to the be the best when using this database, but this fragment is not present in the NR SAbDab database, allowing a near-native structure to be ranked top.

The non-redundant PDB database ('NR PDB') and the antiparallel β-sheet databases perform poorly in comparison to the other databases. None of the average RMSDs from these two databases fall below 2 Å, and they rise as high as 13.9 Å. In fact, as seen in Figure 2.5, for loop lengths of 17 and 19 residues the antiparallel β-sheet database failed to make any predictions at all (coverage = 0%).

'Coverage' is defined as the percentage of cases for which FREAD was able to produce a prediction (*i.e.* how often suitable fragments were found in the database). The best database with respect to coverage is the NR PDB database, which achieves 100% coverage for 8 of the 17 loop lengths tested. This decreases with loop length, presumably because H3 loops are on average longer than general soluble protein loops, and so the number of suitable loop-like fragments in the 'NR PDB' database is lower for longer lengths. The worst coverage is produced by the antiparallel β-sheet database, particularly at longer loop lengths (*i.e.* above 12 residues). After the NR PDB database, the whole SAbDab

database gives the best coverage, followed by the NR SAbDab database. The coverage of the antibody databases appears to correlate with the distribution of H3 loop lengths in known antibody structures (Figure 2.3), with a peak at around 9 residues and lower coverage at the extremes of the length distribution. There are dips in coverage for loops of 6 residues (NR SAbDab database) and 17-19 residues (NR and All SAbDab databases) — reflecting the lack of observed H3 structures of those lengths. This indicates that most predicted fragments are H3 loops themselves, even though the databases contain all possible fragments of the antibodies.

Figure 2.6 shows a comparison between the RMSDs of the top predictions (the highest-ranked decoys) and the best predictions (the decoy giving the lowest global RMSD) for predictions made using the All SAbDab database. If a perfect ranking system were used, these predictions would be the same; however this is not the case. If the ranking system always put the best decoy at the top of the list, accuracies could be improved by up to 1 Å. There is more scope for improvement at longer lengths; there is only a small difference between the top and best predictions for short loops (*i.e.* 3-5 residues) and the RMSD is already usually below 1 Å.

2.3.2 Parameter Investigation

One possible mechanism by which we could improve the results achieved by FREAD is to adjust the parameters controlling which loops from the database are chosen as likely conformations. For example, it is possible that the ESS score and anchor RMSD cutoff values used here are inappropriate when modelling H3 loops. Figure 2.7 shows how the ESS score and anchor RMSD relate to the RMSD of fragments from the database, for the 10-residue loops in the test set. Fragments with high ESS scores tend to have near-native structures, while those with low scores can have a wide range of RMSDs. From the data shown in Figure 2.7, it appears that increasing the ESS score cutoff from 25 to around 40 would improve results, since the number of loops with poor RMSDs would be reduced, making it more likely that the ranking method will choose a near-native structure from the list of predictions. Similarly, the anchor RMSD could be reduced.

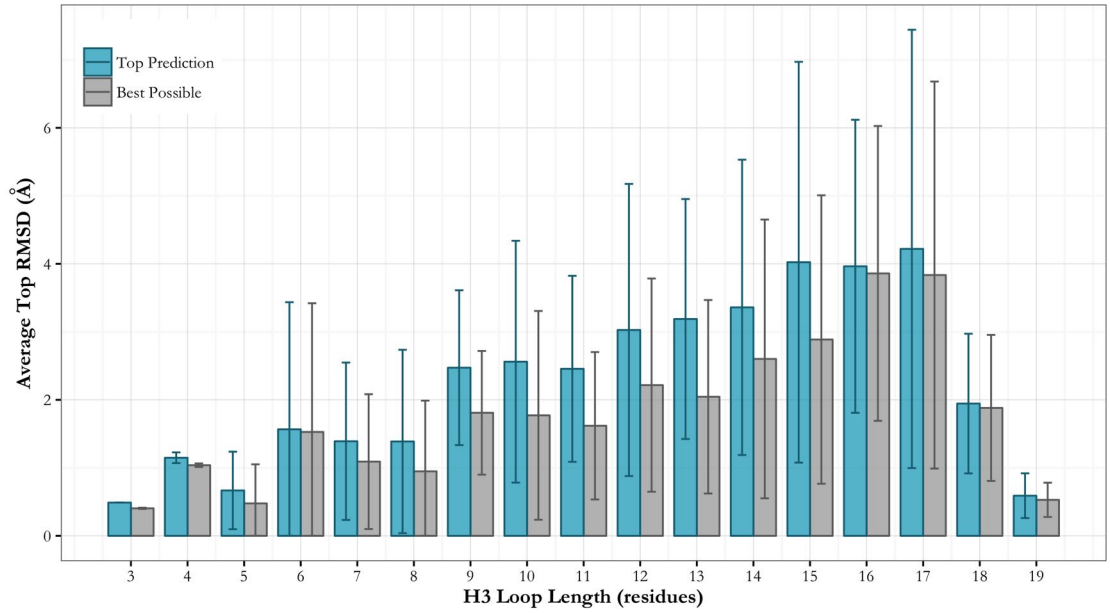

Figure 2.6: The top predictions vs. the best predictions produced by FREAD when using the whole SAbDab database (values shown are the average RMSDs achieved across the test set of H3 loops). 'Top prediction' refers to the decoy predicted to be the best by the FREAD ranking system, and 'best possible' is the fragment from the list of decoys given by FREAD with the lowest global RMSD. Error bars show the standard deviation. The average prediction RMSD could be improved by up to 1 Å if a perfect ranking system could be found.

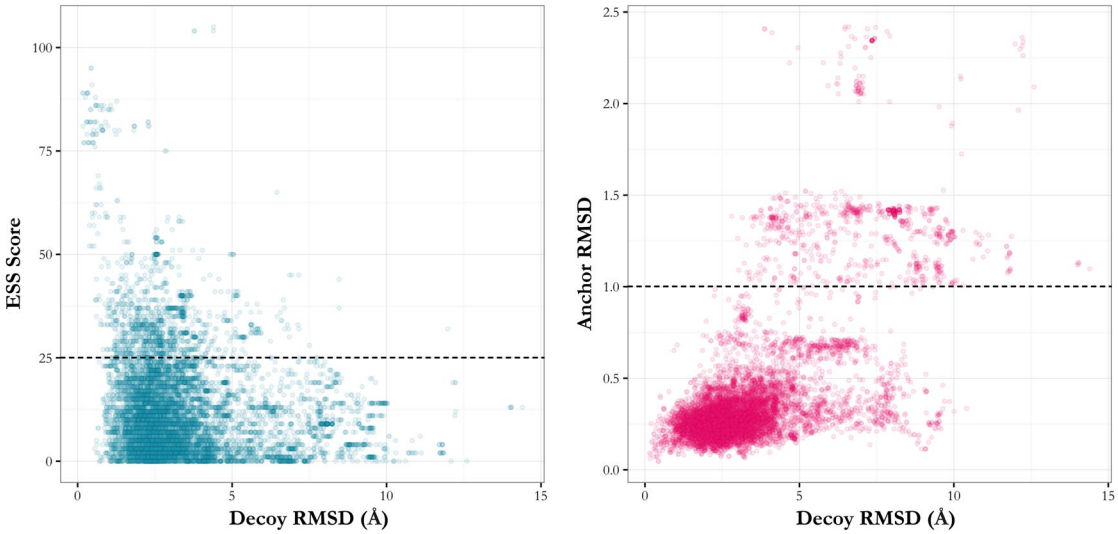

Figure 2.7: The relationship between ESSS/anchor RMSD and decoy RMSD. Results shown are for H3 loops of 10 residues, and were generated using FREAD with the ESSS cutoff removed and the anchor RMSD cutoff raised to 5 Å. Black, horizontal lines represent the current default cutoff parameters. Left plot — environment-specific substitution score (ESSS) against RMSD for all decoys found by FREAD. Fragments with a high ESS score tend to have lower RMSDs than those with low scores. Right plot — anchor RMSD against loop RMSD for all decoys found by FREAD. Fragments with structurally similar anchor residues to the target tend to have lower RMSDs than those with more divergent structures.

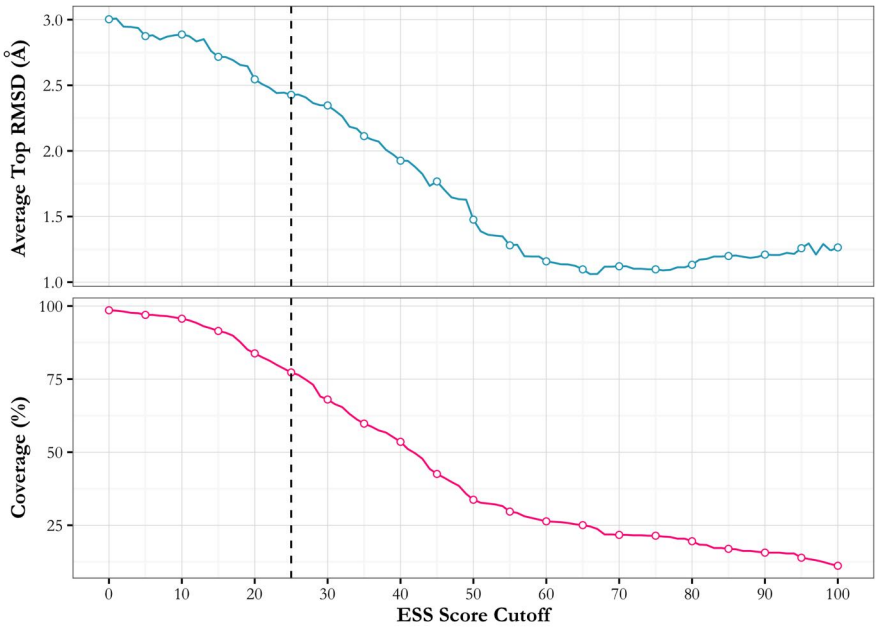

Figure 2.8: ESS score cutoff vs average top RMSD for the set of 691 targets. The default value of 25 is shown as a black vertical line. Increasing the ESSS cutoff and hence making the selection criteria more strict improves prediction RMSD but decreases coverage. A value of 40 would give a good compromise between accuracy and coverage (giving an average top RMSD of 1.93 Å and 54% coverage).

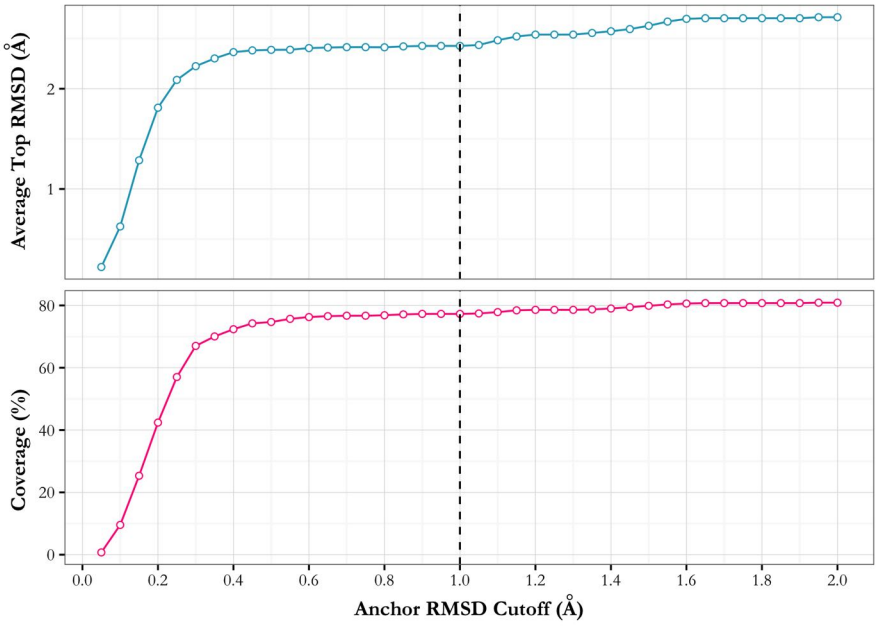

Figure 2.9: Anchor RMSD cutoff vs average top RMSD for the set of 691 targets. The default value of 1 Å is shown as a black vertical line. Decreasing the anchor RMSD cutoff improves prediction RMSD, but only if reduced significantly, to around 0.2 Å. However, this also significantly decreases coverage.

We have investigated how the FREAD parameters can be optimised by independently varying their values and examining how the average RMSD of the top-ranked decoys changes, using the All SAbDab database. The results of this investigation are shown in Figure 2.8 and Figure 2.9, for the ESSS and anchor RMSD cutoffs respectively.

For both ESSS and anchor RMSD, there is a clear relationship between prediction accuracy and coverage. The choice of parameter values is a compromise between prediction accuracy and coverage — making the cutoffs stricter leads to fewer fragments being selected from the database, and if too strict may stop any fragments being chosen at all. On the other hand, tighter restrictions should mean that only high-quality, near-native fragments are selected, reducing the average prediction RMSD. One must decide, therefore, whether it is important to achieve high-accuracy predictions, but with possibly low coverage, or to produce more predictions that are less accurate.

The default ESSS cutoff of 25 produces an average top RMSD of 2.48 Å, with 80% coverage. The highest accuracy results (1.09 Å) were achieved when using a cutoff score of 76; however coverage is extremely low at only 21%. A more reasonable ESSS cutoff would be 40; where the average top RMSD is 1.93 Å and coverage is 54%.

The investigation into the anchor RMSD parameter (Figure 2.9) indicates that the default value (1 Å) is a reasonable choice. Both average top RMSD and coverage stay approximately the same as the cutoff is decreased from 2 Å to around 0.4 Å, then both sharply decrease. There is perhaps room to decrease the default value, but the average top RMSD would not improve by much until the coverage drops substantially. For instance, an anchor RMSD cutoff of 0.3 Å would give an average top RMSD of 2.22 Å and 67% coverage. When combined with the higher ESSS cutoff of 40, average top RMSD becomes 1.67 Å and coverage is 44%.

2.3.3 Ranking Investigation

In addition to changing parameter values, improvements to FREAD's results could be made by ranking the decoys using a different method. We have tested three statistical potentials (calRW (Zhang and Zhang, 2010), dDFIRE (Yang and Zhou, 2008) and SOAP-Loop (Dong *et al.*, 2013)) and compared their ability to rank decoys to that of the

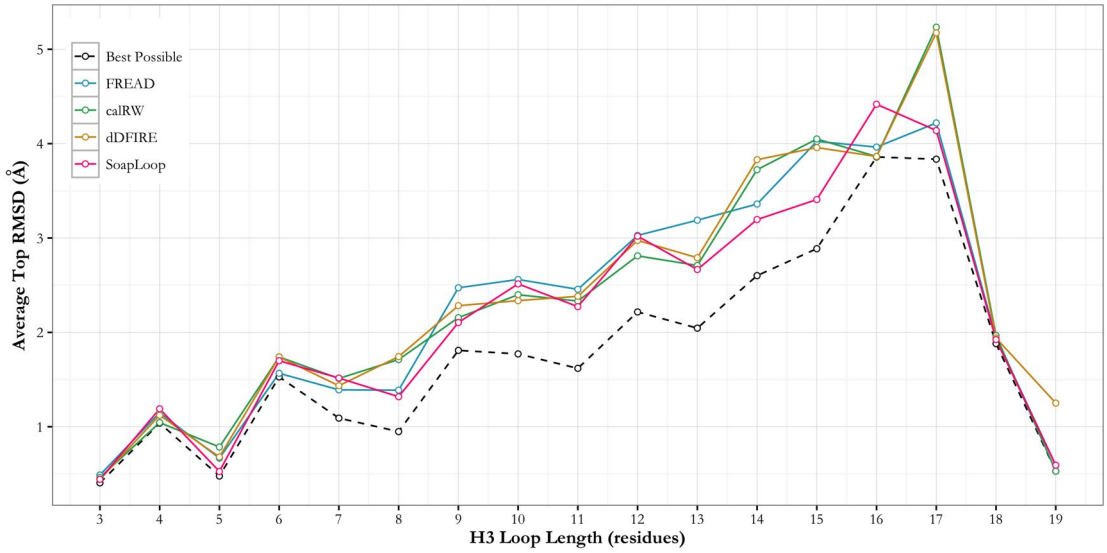

Figure 2.10: The ranking ability of three statistical potentials compared to the FREAD default of using anchor RMSD. The best possible RMSD that can be achieved is shown as a dashed black line. On average, the FREAD method is the worst, however all methods perform similarly.

default method within FREAD (using anchor RMSDs). The outcome of this test is shown in Figure 2.10.

On average, FREAD's method of ranking using anchor RMSD achieves the worst results, producing an average top RMSD of 2.43 Å compared to 2.35 Å, 2.40 Å and 2.28 Å for calRW, dDFIRE and SOAP-Loop respectively. SOAP-Loop therefore produces the best results, however the accuracies achieved by the different methods are very similar.

For short loops, all four ranking methods produce average top RMSDs that are close to the best possible. However, as length increases, the ability of the methods to discern which decoys are closest to the native structure decreases. Exceptions to this are the 18- and 19-residue target loops, where average top RMSDs are very close to the best possible. It is interesting to note that the average best RMSDs for these lengths are lower than would be expected — a ranking method's performance seems to be affected by the quality of the decoys that it is ranking.

2.3.4 Antibody Modelling Assessment II

In 2011, the first Antibody Modelling Assessment (AMA) was conducted (Almagro *et al.*, 2011) — this is a CASP-style blind prediction test (Moult *et al.*, 1995), where the

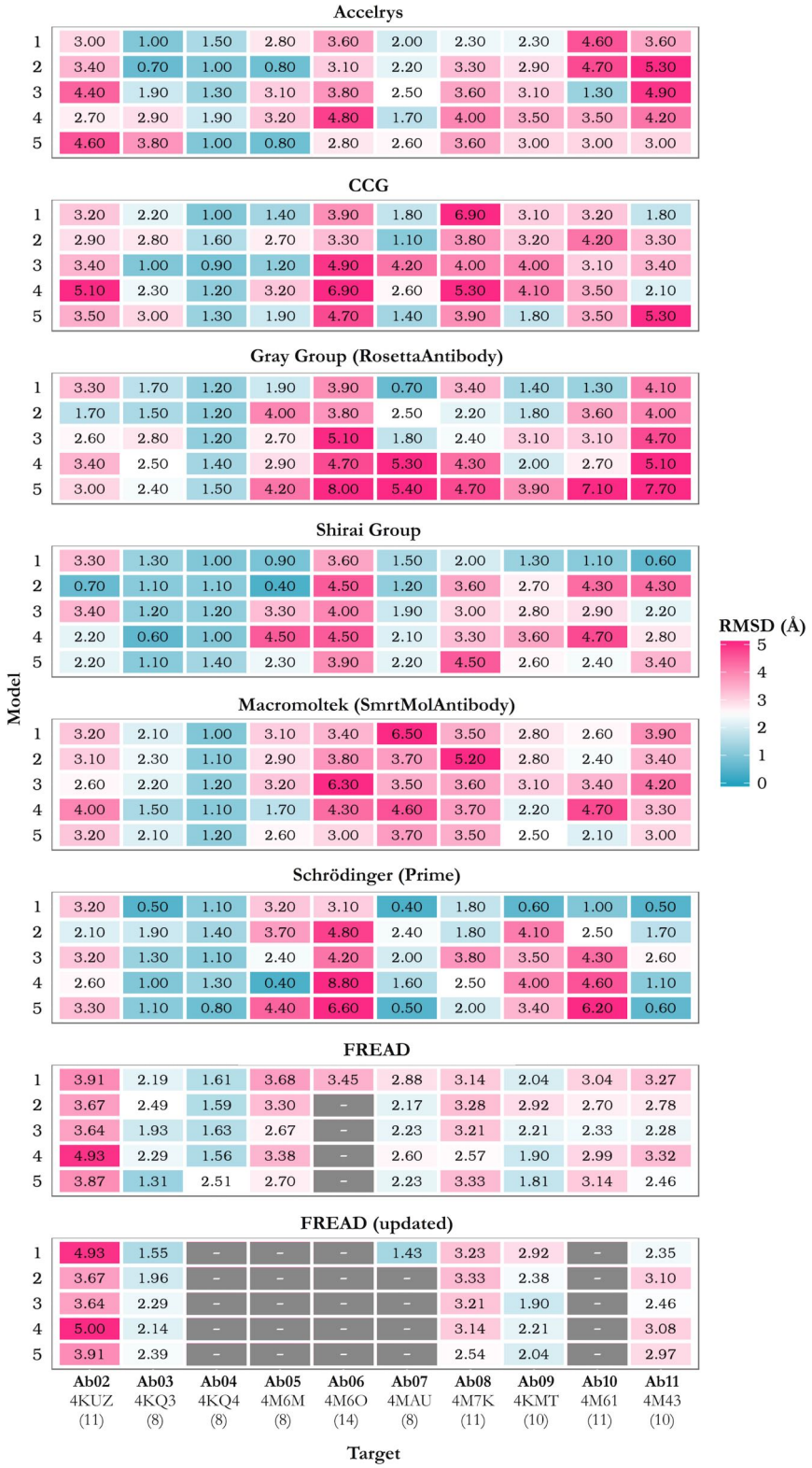

Figure 2.11: Performance of FREAD on the target set from the Antibody Modelling Assessment II. PDB codes and loop lengths are given underneath the target ID. FREAD (default) refers to using the default parameter values and the anchor RMSD ranking method. Grey boxes indicate where FREAD was unable to produce a prediction. FREAD (updated) uses an ESSS cutoff of 40 and ranks decoys using SOAP-Loop, producing similar accuracies but with lower coverage.

participants are given the sequences of several unpublished, high-resolution antibody structures and asked to model them. In 2014, the results of the second assessment (AMA-II) were published (Teplyakov *et al.*, 2014). In the second round of this competition, the current accuracy of H3 prediction structure prediction was benchmarked by giving the participants the native structures of the Fv regions with the H3 loop residues missing. Each group was required to submit five predictions for each of ten H3 targets, with loop lengths ranging from 8 to 14 residues. The results of this competition are reported using carbonyl RMSDs; calculated using only the carbonyl carbon and the oxygen of the loop backbone over residues numbered 95 to 100x (the one before residue 101).

To compare our results to those of other groups, we ran FREAD on the set of AMA-II targets, and its prediction accuracy (along with the accuracy of the other groups) is given in Figure 2.11. To ensure a fair comparison, a date restriction was applied so that fragments from structures deposited in the PDB after March 2013 were ignored. We chose the five top-ranked decoys as our five models. When using the default parameter values and ranking method, the average carbonyl RMSD from all five predictions achieved by FREAD for each target was 2.72 Å, which is better than all other participants except for two (the Shirai group and Schrödinger, whose algorithms both include minimisation steps). The average RMSD of the best of the five predictions per target was 2.37 Å (five groups produced better results than this). 100% coverage was achieved, however in one case (Ab06) only one suitable fragment was found in the database. FREAD produced better decoys on the whole than the only other purely knowledge-based method, CCG.

The second set of FREAD results shown in Figure 2.11 give the performance when increasing the ESSS cutoff to 40 and using SOAP-Loop to rank the decoys. Coverage is decreased; FREAD was only able to provide predictions for six of the ten targets. The average RMSD of the best predictions is decreased slightly to 2.23 Å, however the average RMSD of all five decoys for each target is increased to 2.84 Å.

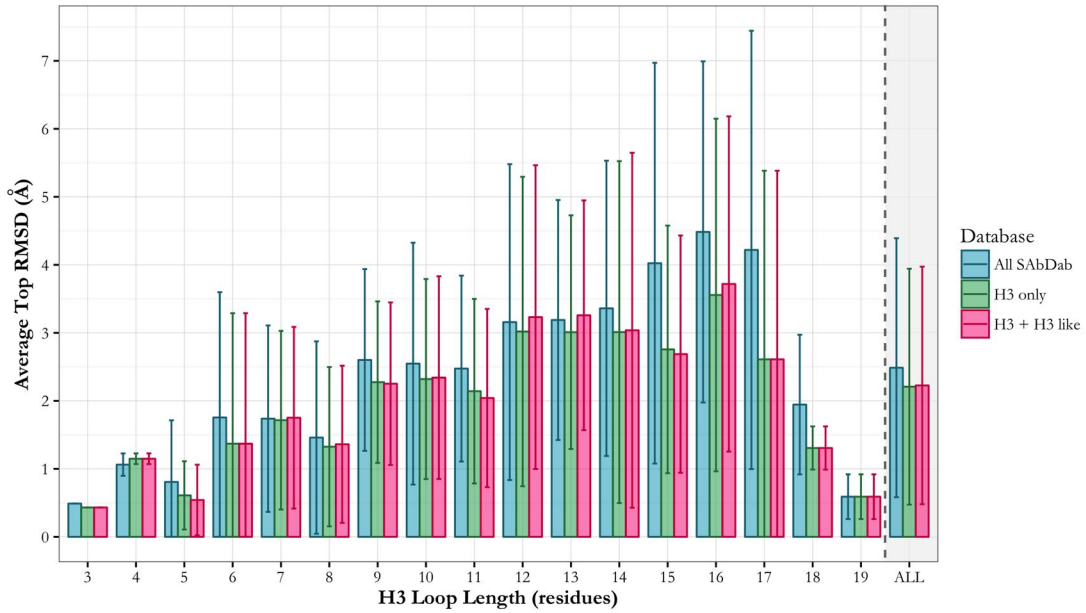

Figure 2.12: Performance of three different fragment databases on the 691 loop target set — the All SAbDab database, a database containing H3 loop regions only, and another with all H3 loops as well as the H3-like fragments found by Weitzner *et al.* Error bars show the standard deviation. Using the H3-only database improves prediction accuracy slightly, but the addition of H3-like fragments does not offer any benefit.

2.3.5 H3-Like Fragments

In 2015, Weitzner *et al.* conducted a search of the entire PDB for loops that have similar conformations to the H3 loops of antibodies. Specifically, they looked to find regions of protein structures that have the 'kinked' conformation adopted by the majority of H3 loops (see Chapter 1). They found 1,030 such loops from 632 protein families, and recommend that they be used in a knowledge-based approach to H3 structure prediction, enriching the existing database with more possible fragments. In particular, they suggest that this may improve long H3 structure prediction, since there are more H3-like non-antibody template loops than there are actual H3 loops from antibody structures.

To investigate this, we created two more databases; one containing only the H3 loop regions ('H3-only'), and another combining H3 loops and the H3-like loops found by Weitzner *et al.* (2015) ('H3+H3 like'). The results achieved using these databases, as well as the All SAbDab database, are shown in Figure 2.12. Compared to the All SAbDab database, the H3-only database improves the average top RMSD (from 2.48 Å to 2.21 Å), indicating that most near-native predictions are made using other H3 structures.

However, coverage decreases from 80% to 74%, suggesting that fragments from other regions of antibodies can still be useful as decoys on some occasions.

The addition of H3-like structures does not dramatically affect results — the average top RMSD is 2.23 Å (compared to 2.21 Å for the H3-only database) and coverage remains the same. This implies that the addition of H3-like structures into the database does not offer much benefit — in fact, of the 691 loops in the test set, H3-like fragments were only selected from the database for 15 (2%). However, while the mean RMSD of the decoys derived from H3-like structures was 2.63 Å, individual RMSDs were as low as 1.16 Å.

2.4 Conclusion

The FREAD database containing all structures from SAbDab produced the best results — loops of up to 8 residues in length are, on average, predicted with global RMSDs below 2 Å. However, even when using this database, RMSDs for mid-range loop lengths are not as good as expected when one considers previous FREAD performance (Choi and Deane, 2010; Kelm *et al.*, 2013). The antiparallel β-sheet FREAD database produced worse results than expected — even though H3 loops themselves are loops connecting β-strands, the average top RMSD achieved using this database was 5.21 Å. This implies that antibody H3 loops are somehow structurally different from other protein loops. Even the addition of fragments thought to be H3-like in structure does not improve results.

FREAD also achieved good results for the set of targets from the second Antibody Modelling Assessment. When considering the best model submitted for each target, FREAD is ranked sixth out of the seven competitors. While the results indicate that FREAD is more consistent at producing near-native structures when considering all models submitted, this may be a consequence of how the other participants selected their five top decoys. For example, the five predictions for each target may have been purposefully chosen so that they cover a larger conformational space between them, preventing the submission of five very similar but incorrect models.

We investigated the effect of the parameter values used by FREAD to see whether results can be improved further. By making cutoffs more strict, the average top RMSD can be improved, but since fewer fragments are selected from the database, coverage decreases. The decision of which values to use therefore depends on the how the resulting model is to be used, and how accurate it is required to be.

FREAD's performance can be improved by changing the ranking system. A decrease in RMSD of up to 1 Å could be made if loops were always ranked correctly. We tested three statistical potentials and compared their ranking accuracy to that of the FREAD default method, and found that they are better (if only slightly). Since statistical potentials are normally very quick to score decoys, it is feasible that one of them could be incorporated into the FREAD algorithm.

For the large test set, increasing the ESSS cutoff to 40 and using SOAP-Loop to rank the decoys improved prediction accuracy. However, when these changes were made and applied to the AMA-II target set, we do not see the same improvement. However, this is a small and particularly difficult set (only one of the loops has an exact sequence match in the PDB), and therefore it is difficult to draw conclusions from this.

Overall, FREAD produces good results, however coverage is often low, since there are some loop targets for which the database contains no suitable structures. A way of overcoming this issue would be to use an *ab initio* method instead, which would give 100% coverage consistently since there is no dependence on pre-observed loop structures. In the next chapter we describe the development of a new *ab initio* method, MECHANO, designed specifically for modelling H3 loops.

Pre-Chapter Image: A zinc finger, which features the coordination of a zinc atom by loop residues, and has a role in RNA binding.

3

MECHANO: Modelling H3 loops using an *ab initio* approach

Contents

3.1 Introduction

Like knowledge-based loop modelling algorithms, *ab initio* methods consist of three main steps: sampling, filtering and ranking. However, instead of sampling from a set of pre-observed loop structures, *ab initio* methods build possible conformations (decoys) computationally. As a result, they are not limited by available experimental data and are, in theory, able to predict novel structures. Also, unlike knowledge-based methods,

ab initio methods should always give a prediction; *i.e.* coverage is 100%.

Decoys are generated by exploring the conformational space. However, the conformational space is extremely large and performing an exhaustive search (*i.e.* examining every possible structure) is impossible. Therefore, techniques must be applied to direct the search towards the native structure, for example by randomly selecting dihedral angles from Ramachandran distributions or performing clash checks. In this chapter we describe MECHANO, a new *ab initio* algorithm that we have developed specifically for generating candidate structures (decoys) of antibody H3 loops, and show how adding these techniques can improve the conformational search.

3.2 Method

A flowchart outlining the MECHANO algorithm is shown in Figure 3.1, and illustrations of the main steps in loop building are shown in Figure 3.2. The inputs to the algorithm are an antibody structure (in PDB format), the sequence and location of the loop to be modelled, and the number of loop conformations (decoys) to be generated. Each decoy generated by MECHANO is independent. Like many *ab initio* methods, the main steps of loop building in MECHANO are loop initialisation (where residues are added sequentially to one of the two loop anchors) and loop closure. If the loop closure algorithm fails then the decoy is rejected and loop building starts again. Loop generation continues until the required number of decoys has been made.

3.2.1 Basic MECHANO Algorithm

Loop Initialisation

MECHANO uses two anchor residues; one at each end of the loop. Loops are built by adding backbone atoms sequentially to one of the anchor residues (this can be either the N- or the C-anchor). When building from the N-anchor, atoms are added in the order N, Cα, C, O; while the addition of atoms from the C-anchor occurs in the order

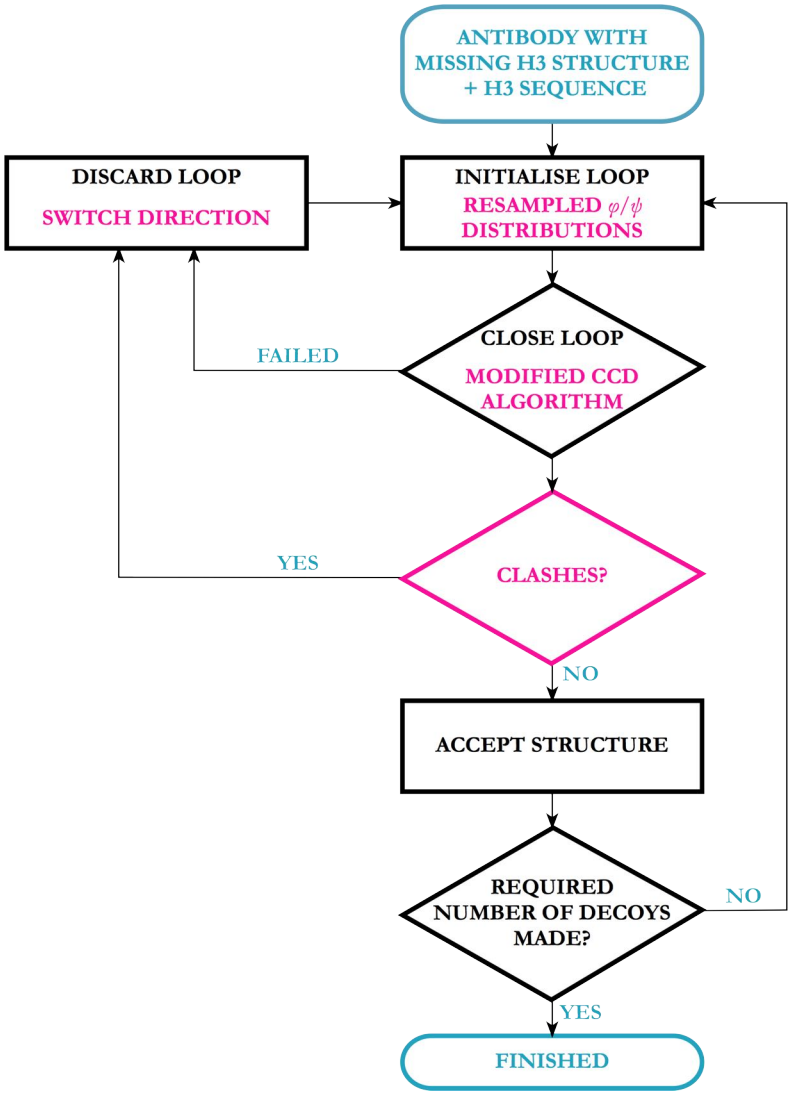

Figure 3.1: Flow diagram displaying the main stages of the MECHANO algorithm. Steps of the basic algorithm are shown in black, while the additions we have made are highlighted in pink. These features are described in detail in Section 3.2.2.

C, O, Cα, N (since the carbonyl carbon atom has to be present for the addition of the oxygen). For each atom to be added, three values are required, giving its position in relation to the atoms already in the structure: a bond length, bond angle and a dihedral angle. For example, for the addition of the N_i atom (the backbone nitrogen of residue i), the values needed are the C_{i-1}—N_i bond length, the $C\alpha_{i-1}$—C_{i-1}—N_i bond angle, and the N_{i-1}—$C\alpha_{i-1}$—C_{i-1}—N_i dihedral. The positions of the new atoms are calculated from these values using the algorithm described in Parsons *et al.* (2005).

Each residue, therefore, having four backbone atoms, requires four bond lengths, four bond angles and four dihedral angles. Bond lengths and bond angles are sampled

randomly from Gaussian distributions, using the mean values and standard deviations found in Engh and Huber's 1991 paper. The dihedral angles for each residue, however, are more complicated. The peptide bond is planar, and the majority of residues are found in the trans conformation (an ω angle of roughly 180°). The ϕ and ψ angles, on the other hand, are interdependent, and are limited to values that do not cause steric clashes. The dihedral angle required to add a carbonyl oxygen atom is related to the ψ angle, since the carbonyl carbon has trigonal planar geometry, and hence the oxygen and carbon

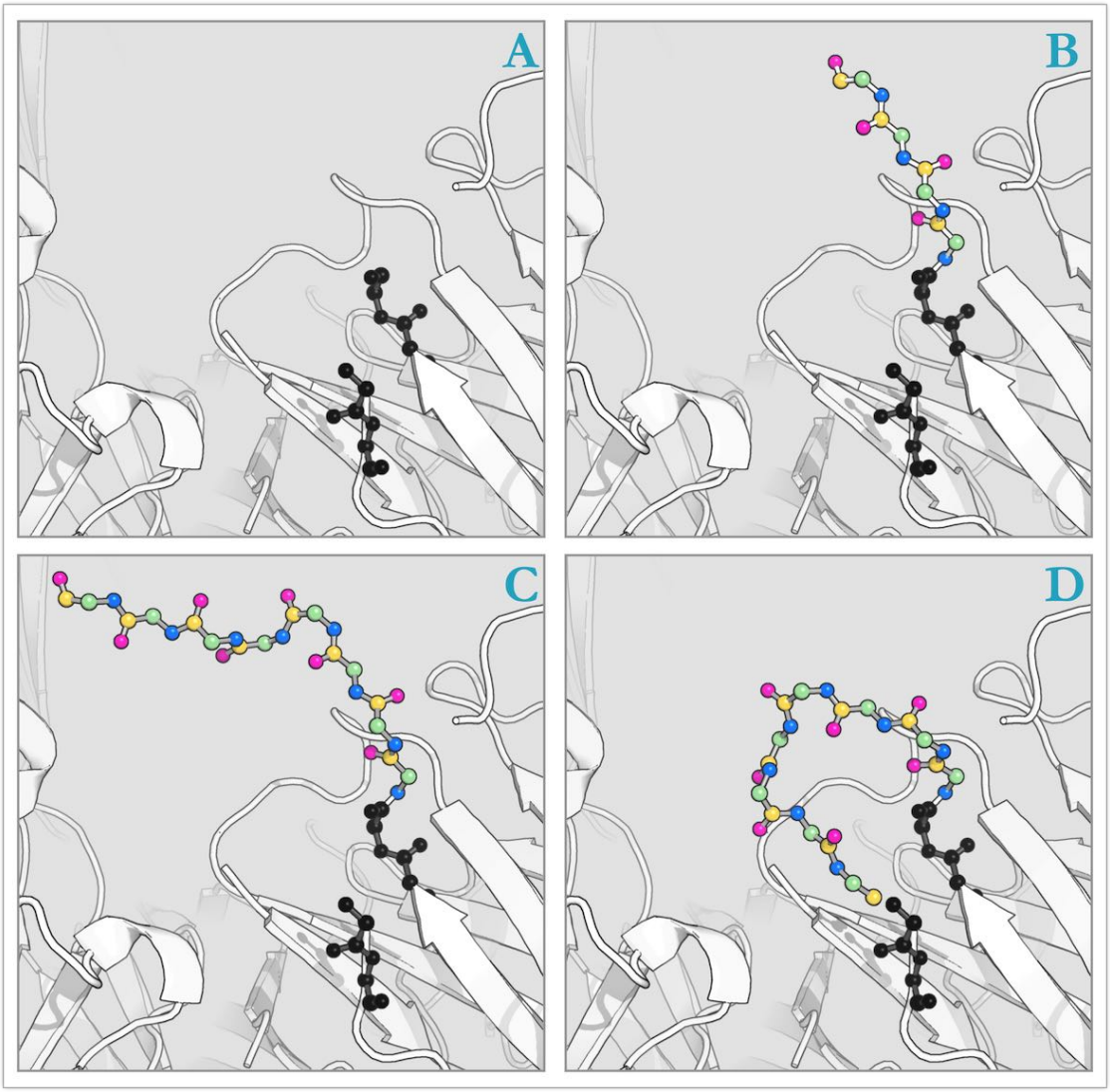

Figure 3.2: Diagram showing the different stages of loop building in MECHANO. MECHANO is given a gap in a protein structure, in which to model a loop (A). The coordinates of the anchor residues are extracted (shown in black). Loop building begins by adding backbone atoms sequentially to one of the two anchors (B) until all residues are complete (C). The loop is then closed using the CCD algorithm (D).

atoms of one residue, and the nitrogen atom of the next, are located in the same plane.

MECHANO, like many *ab initio* methods, has its own libraries of observed ϕ/ψ combinations, from which angles are randomly chosen. MECHANO uses 20 of these ϕ/ψ libraries, one for each amino acid type. The angles are grouped into 1° bins (*i.e.* the Ramachandran plot is split into $1° \times 1°$ sections). Other methods use larger bins and often group amino acid types together, leading to fewer libraries: for example, MODELLER has libraries for all 20 amino acids using 5° bins; and both PLOP and RAPPER have three libraries (proline, glycine, all other amino acid types), with 5° bins and 3° bins respectively.

For each residue, one of these bins is selected at random from the relevant ϕ/ψ library, with a probability proportional to the number of ϕ/ψ observations in that bin. MECHANO then chooses random values that fall within the chosen bin (by adding a random number between 0 and 1 to the lower bound of the bin) to yield the final ϕ/ψ angles that will be used in loop building.

The basic MECHANO algorithm uses dihedral data from general protein loop types. To create this data, we calculated the ϕ/ψ angles of all loop regions of the PDB entries that met the following criteria: non-redundant sequences (maximum sequence identity of 60%), resolution below 2Å, and R factor below 0.3. Loop regions were selected using DSSP (Joosten *et al.*, 2011): we considered a loop region to be three or more consecutive residues that are not designated as part of a sheet (code 'E' in DSSP) or a helix (codes 'H', 'G' and 'I').

Loop Closure

Since the termini of the loop must be connected to the anchor residues, forming a continuous backbone, the structure generated in the loop building step must be 'closed'. For example, if the loop was built from the N-anchor, then the loop structure must be altered so that the other end of the loop attaches to the C-anchor. Loop closure is normally achieved by changing the ϕ/ψ angles of the structure, and methods for doing this can be classified into three categories: analytical methods, where the change in ϕ/ψ angles required is calculated directly; iterative methods, in which the loop structure is

changed gradually over a series of steps to achieve closure; and build-up methods, which attempt to produce a closed structure by guiding the building procedure.

A popular and algorithmically robust loop closure algorithm is cyclic coordinate descent, or CCD (Canutescu and Dunbrack Jr., 2003). CCD is both an analytical and an iterative method. To the end of the 'open' loop structure are added the atoms of the relevant anchor residue, with the conformation present in the target structure (for example, if a loop was built starting from the N-anchor, the atoms of the C-anchor residue are appended to the C-terminal end of the loop). These atoms are therefore present twice: the 'fixed' anchor residue (the true structure) and the 'mobile' anchor residue (the one added to the loop structure), and in theory can be structurally aligned perfectly. Starting from the end of the loop that is attached to the rest of the protein (*i.e.* the end at which loop building began), the dihedral angles are changed sequentially to try and minimise the RMSD between the fixed and mobile anchor residues, aiming to achieve a perfect structural alignment of the two and hence forcing the loop into a 'closed' conformation. The angle change that would minimise the fixed-mobile distance is calculated analytically. The process continues iteratively until the distance is within a cut-off value (*e.g.* 0.2 Å), at which point the loop is considered to be closed.

Implementation

MECHANO is written in the programming language C, and has approximately 3,000 lines of code. It is run as an executable, and as input it requires: the path to the target protein structure (in PDB format), the chain identifier of the chain within the target file that contains the target loop, the sequence of the loop, the length of the loop, the start and end residues, and the number of decoys to generate.

3.2.2 Additions to the MECHANO algorithm

We have added several features to the basic MECHANO algorithm to enable more efficient searching of conformational space. We have:

- incorporated antibody-specific data in both loop initialisation and loop closure;

- developed a modified version of the cyclic coordinate descent (CCD) loop closure algorithm;
- included a filtering step that checks for clashes between the loop and the rest of the protein;
- introduced 'switching', where the loop can be built from either of the two anchors (the N- or the C-anchor, where the N-anchor is closest to the N-terminus of the protein).

These features are described in more detail in the following sections.

Inclusion of resampled antibody-specific data

Since there are currently only around 2500 antibody structures available, there is not enough dihedral data to have residue-specific libraries of ψ/ϕ angles when just considering H3 loop data. The data for MECHANO's libraries was therefore generated from both H3 data and general protein loop data using a resampling method. This required two sets of dihedral data: ϕ/ψ data from H3 loops, and ϕ/ψ data from general protein loops. The H3 data was generated by calculating the ϕ/ψ angles of all the H3 loops in SAbDab (*i.e.* residues 95-102 using Chothia numbering). The data from general protein loops is the same as that described in Section 3.2.1.

Separate distributions were made for each amino acid type. Random ϕ/ψ angles were chosen from the general loop data and added to the H3 distributions to make them less sparse. The probability that a randomly selected point is added to the H3 distribution is dependent on the proximity of the random angles to those already present in the H3 data:

$$P(\text{addition}) = \begin{cases} 0 & \text{if } d > \sqrt{8} \\ 1 & \text{if } d = 0 \\ \frac{1}{\sqrt{16\pi}} \exp\left(-\frac{d^2}{16}\right) & \text{if } 0 < d \leqslant \sqrt{8} \end{cases}$$

where d is the distance to the nearest pre-existing point. In the third case (d is greater than 0 but less than $\sqrt{8}$) the probability is calculated using a Gaussian function centred on the closest pre-existing point (*i.e.* its maximum occurs when d is 0) and a variance of 8. Once a point has been added to the distribution, it can then be considered as a

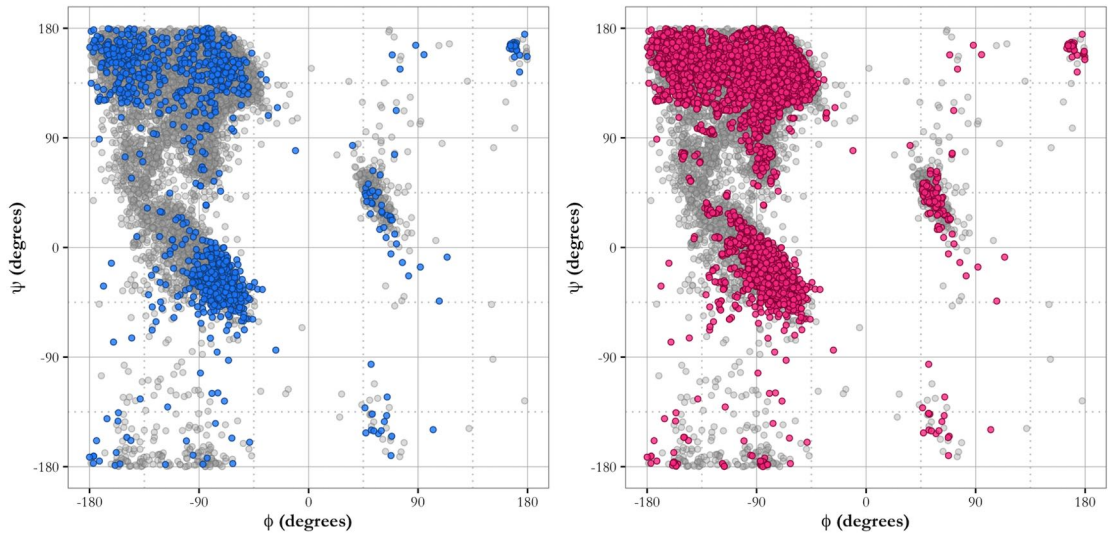

Figure 3.3: Ramachandran resampling, shown for alanine residues. H3 ϕ/ψ dihedral angles (blue, left plot) are different to general protein ϕ/ψ dihedral angles (grey, both plots), but there is a limited amount of H3 data. By using our resampling method, we can generate a Ramachandran distribution (pink, right plot) that is less sparse than the original H3 data, but maintains its unique nature.

nearest neighbour for the following resampling iterations. A total of 20,000 iterations were carried out to give the final resampled distributions (an example of which is shown in Figure 3.3). By adding points that are closer to those already present with a higher probability, the overall characteristics of the H3 data are maintained, enabling us to use this data for loop building in MECHANO. This approach has not been used before in other loop modelling algorithms.

Modified CCD algorithm

For MECHANO, we have written a modified version of the CCD algorithm (see Section 3.2.1), as suggested by Canutescu and Dunbrack Jr. (2003), where constraints are put on the angle changes so that only the allowed regions of the Ramachandran plot are occupied. Using the resampled data from Section 3.2.2, we calculated probabilities for all possible ϕ/ψ combinations. For every residue type, the resampled data was split into 5° bins, and the probability of angles from that bin being observed is given by:

$$P(i) = \frac{N_i}{\sum_j N_j}$$

where N_i is the number of points in bin i. In the same way as the normal CCD algorithm, residues are considered sequentially, except ϕ/ψ are now considered as pairs. The changes in the angles required to minimise the fixed-mobile anchor distance are calculated, along with the probabilities of both the current ϕ/ψ angles and the proposed ones. The angle change is accepted with a probability given by:

$$P(\text{acceptance}) = \begin{cases} 1, & \text{if } P(\text{proposed}) \geq P(\text{current}) \\ \frac{P(\text{proposed})}{P(\text{current})}, & \text{if } P(\text{proposed}) < P(\text{current}) \end{cases}$$

In this way, the algorithm is restricted from changing the dihedral angles to disallowed values; therefore all decoys generated by MECHANO will be physically feasible in ϕ/ψ space.

Clash Check

An important step in decoy generation is the steric clash check. This is used to filter out any conformations that are physically impossible due to the close proximity of their atoms. The clash check in MECHANO uses a hard-sphere approximation to detect clashes between backbone atoms. Each atom type has a defined van der Waals radius, and the addition of the radii for two atoms gives the minimum distance that must exist between them. The values used by MECHANO for these radii are derived from crystal structures (Tsai *et al.*, 1999). If any atoms in the loop decoy are calculated to be too close to either the atoms in the loop itself or the rest of the protein structure, then the decoy is discarded. The conformational space that the decoys produced by MECHANO represent is therefore constrained to the physically acceptable regions.

Switching

It may be preferential for a loop to be built from one anchor instead of the other — for example, it may be easier to close the loop if it is built from a particular anchor (due to buried residues etc.) — this may lead to the decoys built in one direction being more accurate than the other. Switching (Chys and Chacon, 2013) allows the preferred direction to dominate. Loop generation begins by building from the N-anchor, but if the loop closure algorithm fails or a clash is detected, then the direction of loop

Table 3.1: A summary of the six different versions of MECHANO.

#	Version	Summary
1	Basic	The version of MECHANO described in Section 3.2.1.
2	Switching	The Basic version, incorporating the ability to 'switch' the direction of loop building (see Section 3.2.2).
3	Resampled Data	The Basic version, but using the resampled H3 dihedral data instead of data from only general loop types (see Section 3.2.2).
4	Clash Check	The Basic version, incorporating a clash check to filter out physically unfeasible loop structures (see Section 3.2.2).
5	Modified CCD	The Basic version, using the modified CCD algorithm instead of the normal version (see Section 3.2.2).
6	All Additions	A version that incorporates all of the changes made in versions 2-5 (as described in Section 3.2.2)

building changes to the C-anchor. If a failure occurs again, then building switches back to the N-anchor, and so on. In this way, the final set of decoys contains more that were built in the preferential direction.

3.2.3 Testing the MECHANO algorithm

The performance of different versions of the MECHANO algorithm was evaluated on a test set of 40 H3 loops (see Appendix, Table A.3 for more details). Five H3 loops were selected for each of lengths 4, 6, 8, 10, 12, 14, 16 and 18 residues. 5000 decoys were generated for each H3 loop. We developed six versions of MECHANO (see Table 3.1): the basic version as described in Section 3.2.1; four versions containing each of the added features described in Section 3.2.2; and one version containing all of these additions.

3.2.4 Comparison to other loop modelling software

To gauge how well MECHANO performs when modelling H3 loops, several other freely available loop modelling methods were tested, using the same test set as that described in Section 3.2.3 to enable direct comparison. The software used (and any relevant settings) were:

- FREAD (Deane and Blundell, 2001; Choi and Deane, 2010) — as described in Chapter 2;
- Loopy (Xiang *et al.*, 2002; Soto *et al.*, 2007) — 5000 initial conformations generated, 1 final prediction as output;

- Loopy + PLOP (Jacobson *et al.*, 2004; Zhu *et al.*, 2006) — the top prediction for each target from Loopy was subjected to energy minimisation using the Protein Loop Optimisation Program (PLOP);
- RAPPER (DePristo *et al.*, 2003; de Bakker *et al.*, 2003) — 5 runs generating 1000 decoys each were performed for each loop target. Decoys were ranked using the RAPDF score.

MECHANO does not have an in-built decoy ranking system. In order to compare MECHANO's performance to that of the other loop modelling software described above, which do include ranking steps, the decoys generated using MECHANO were subjected to sidechain prediction using SCWRL4 (Krivov *et al.*, 2009), and subsequent ranking using the statistical potential dDFIRE (Zhang *et al.*, 2004).

3.3 Results and Discussion

3.3.1 Additions to the basic MECHANO algorithm

The results achieved using different versions of MECHANO on the 40-loop H3 test set are shown in Figure 3.4. In addition, the RMSD distributions of the decoys generated for loops of ten residues are shown in Figure 3.5, and the computation times are shown in Table 3.2. We have tested six versions: a 'basic' version, which does not include any of the additions discussed in Section 3.2; versions that include switching, clash checks, resampled H3 data, and the modified CCD algorithm; and one version including all four of these. Since MECHANO includes no ranking step, the results reported here are the average for each loop length of the 'best' RMSDs — *i.e.* the RMSD of the most accurate decoy MECHANO generated. This makes it possible to see how effectively MECHANO searches the conformational space, without introducing errors through ranking. The results are indicative of how well MECHANO would perform if the ranking method worked perfectly. All the additions reduce the average best RMSD (except switching for the 16- and 18-residue loops and the resampled data for

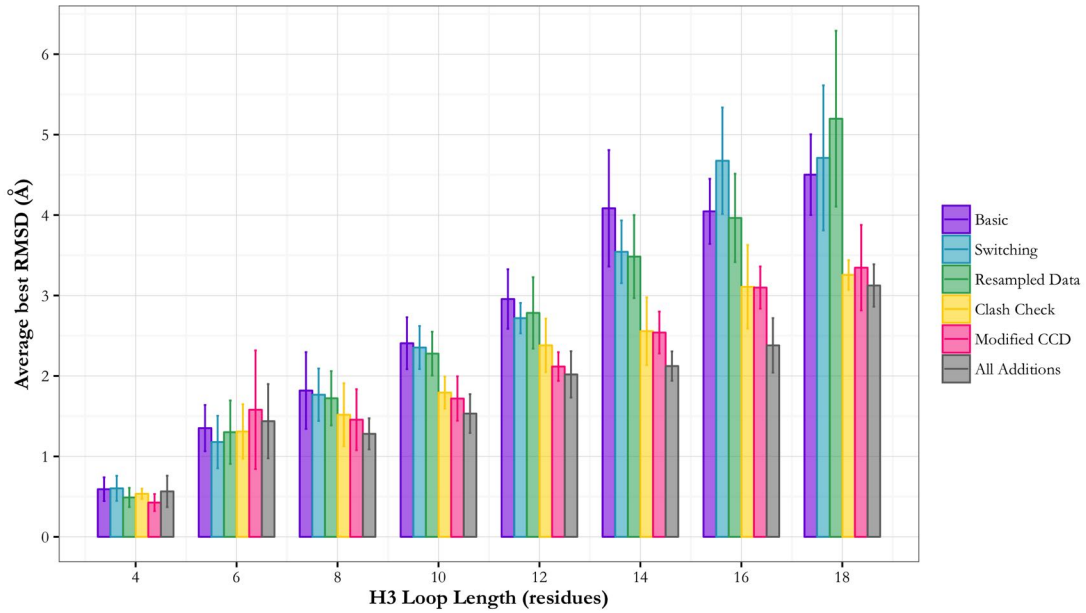

Figure 3.4: Accuracies achieved using the six different versions of MECHANO on the test set of 40 H3 target loops (five per loop length). 'Average best RMSD' refers to the mean of the lowest RMSDs produced for each target at a particular loop length, and the error bars show the standard variation. The version of MECHANO including all four additions produces the best decoys.

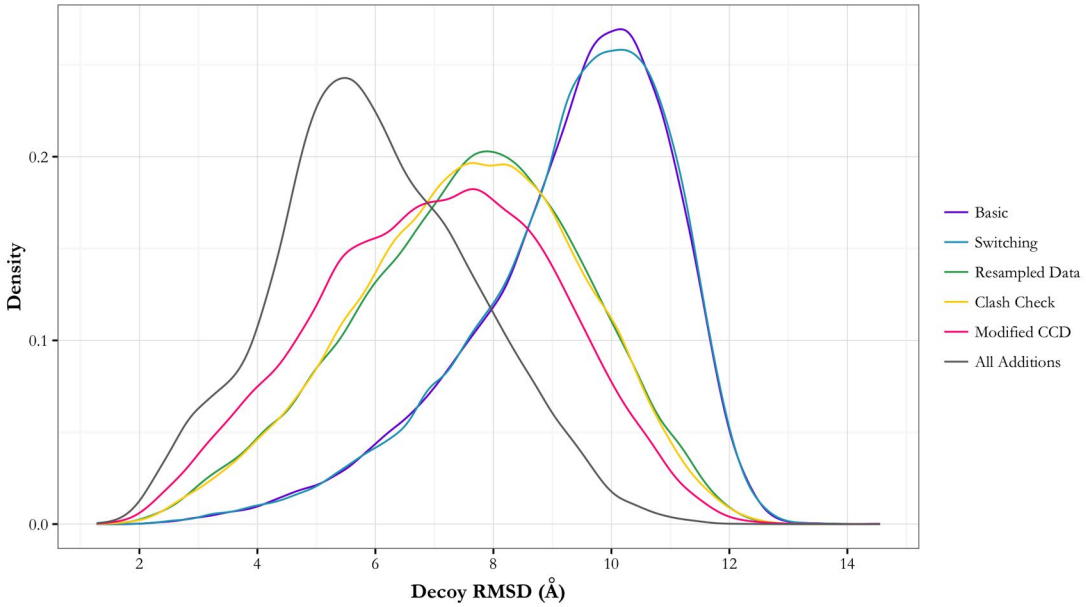

Figure 3.5: RMSD distributions of the 5000 decoys generated for each target loop using the six different versions of MECHANO, for the 10-residue target loops in the 40-loop test set. Increased density to the left of the plot shows an enrichment of good conformations in the decoy set. More near-native decoys are generated by the version of MECHANO that includes all four additions.

Table 3.2: Average computation times for the six versions of MECHANO, when run on one core of a Linux server with 128 GB RAM, and a 2.1 GHz processor.

MECHANO Version	Time (min)
Basic	2.28
Switching	2.05
Resampled Data	2.17
Clash Check	67.94
Modified CCD	15.54
All	97.51

the 18-residue loops), and also increase the number of low-RMSD conformations in the decoy set, making it more likely that a ranking method would select an accurate model. The aspect of the algorithm that has the largest effect is the modified CCD algorithm, reducing the best RMSDs by over 1 Å in some cases, and enabling MECHANO to generate more low-RMSD decoys (Figure 3.5). The inclusion of all additions produces the best results, with loops of up to 12 residues being predicted with an RMSD of or below 2 Å. However, these results are worse than the best predictions achieved by FREAD (Figure 2.4), and there is a clear dependency upon loop length.

We have also calculated the ϕ/ψ angle distributions for the residues of the loop decoys. The Ramachandran plots for the ϕ/ψ angles of the 10-residue decoys are displayed in Figure 3.6, for decoys generated both with and without the modified CCD closure method. It is clear that the modified CCD loop closure algorithm does restrict the ϕ/ψ angles to the allowed regions of the Ramachandran plot (Figure 1.4), producing more physically realistic loop decoys. This is reflected in the best RMSD results and the RMSD distributions.

3.3.2 Software Comparison

We have compared MECHANO to several freely available methods, as described in Section 3.2.4; the accuracies achieved by each method are shown in Figure 3.7. The results shown on this graph are the top predictions, as selected using the dDFIRE ranking method. The accuracy of the best decoys generated for three of the methods are shown

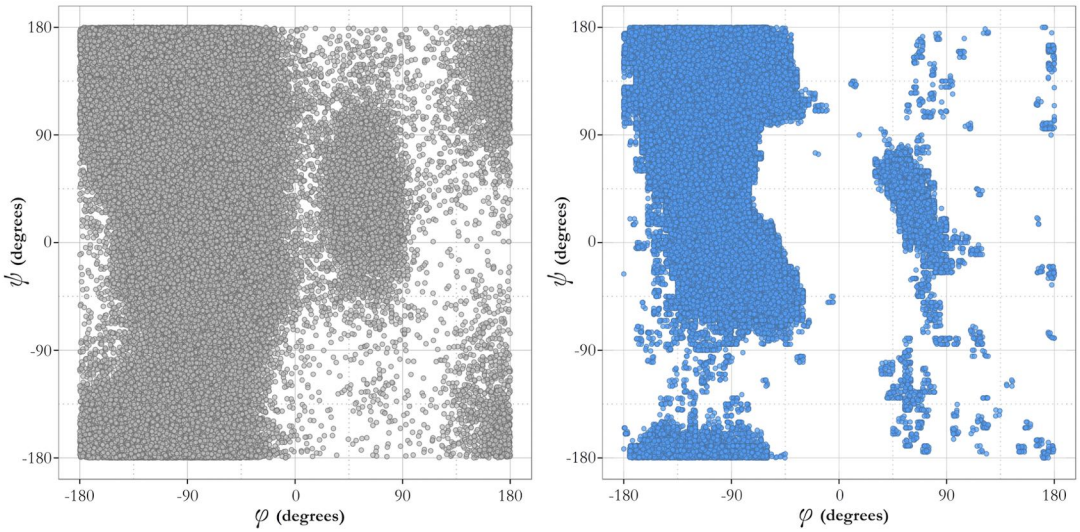

Figure 3.6: Ramachandran plots showing the ϕ/ψ angle distributions of decoys generated by MECHANO for the five 10-residue target loops. The plot on the left shows the distribution when using normal CCD and the one on the right shows the result when using the modified, restrained version of the CCD algorithm. By using the restrained version, the dihedral angles of the decoys only occupy the allowed regions of the Ramachandran plot.

in Figure 3.8 (Loopy only outputs one structure which its internal ranking has predicted to be the closest to the real structure, and so the actual best conformation produced by the method cannot be determined). dDFIRE does not rank the loop decoys correctly — on average, the predicted structure is 1.75 Å less accurate than the best possible. If the ranking system worked perfectly, MECHANO would produce decoys of a similar accuracy to FREAD (with ranking), the only knowledge-based method tested.

For loops of up to 10 residues, MECHANO produces slightly less accurate predictions than all of the other methods except RAPPER — however this can be attributed to failures in ranking the decoys, since the MECHANO 'best' predictions are actually more accurate than the best of the other methods (Figure 3.8). For longer loops, however, even with incorrect ranking choices, MECHANO performs better than all the other *ab initio* methods that we tested.

3.3.3 Antibody Modelling Assessment II

As in the case of FREAD (Section 2.3.4), we ran MECHANO on the H3 targets from the recent Antibody Modelling Assessment (Teplyakov *et al.*, 2014), to compare our

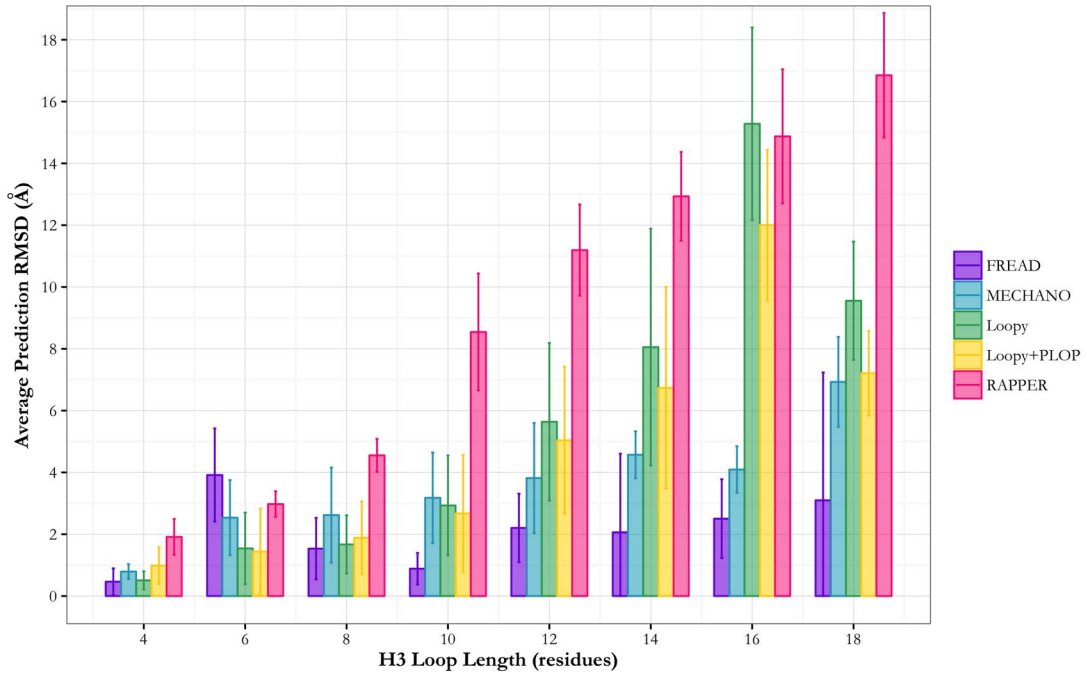

Figure 3.7: The accuracy of predictions made by MECHANO for the 40 target loops, compared to other loop modelling methods (see Section 3.2.4). The MECHANO results here refer to the top predictions as ranked by dDFIRE. MECHANO generally produces better predictions than the other *ab initio* algorithms tested, especially for long loop lengths.

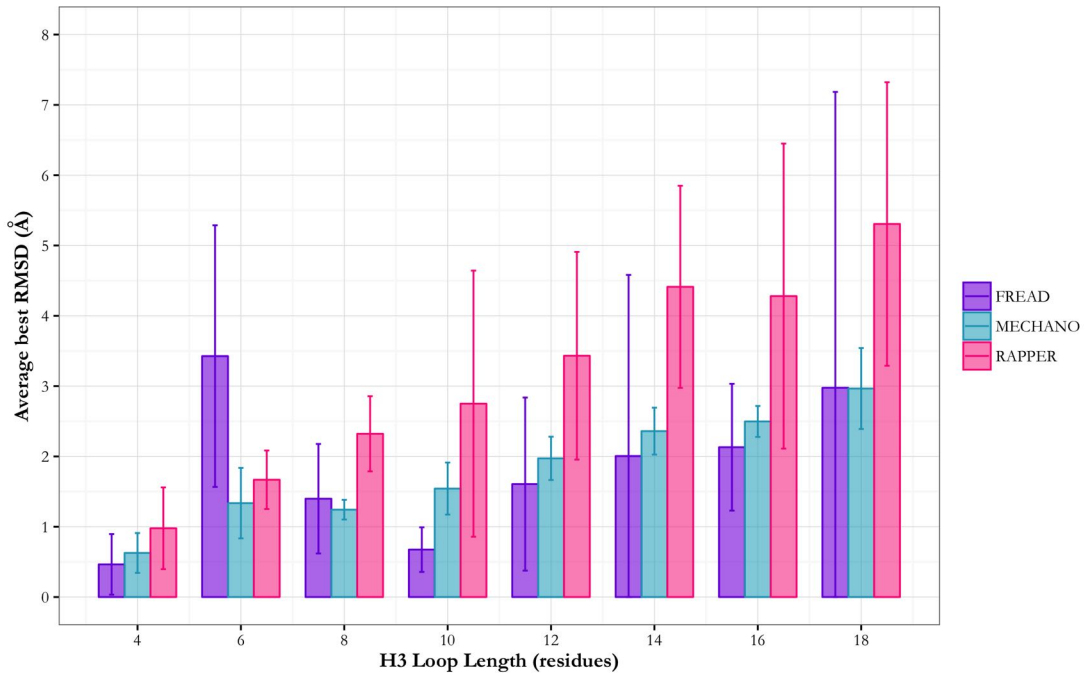

Figure 3.8: The accuracy of the best decoys made by MECHANO for the 40 target loops, compared to those of the other loop modelling methods (see Section 3.2.4). Results shown are the averages of the best decoys generated for the target loops at each length.

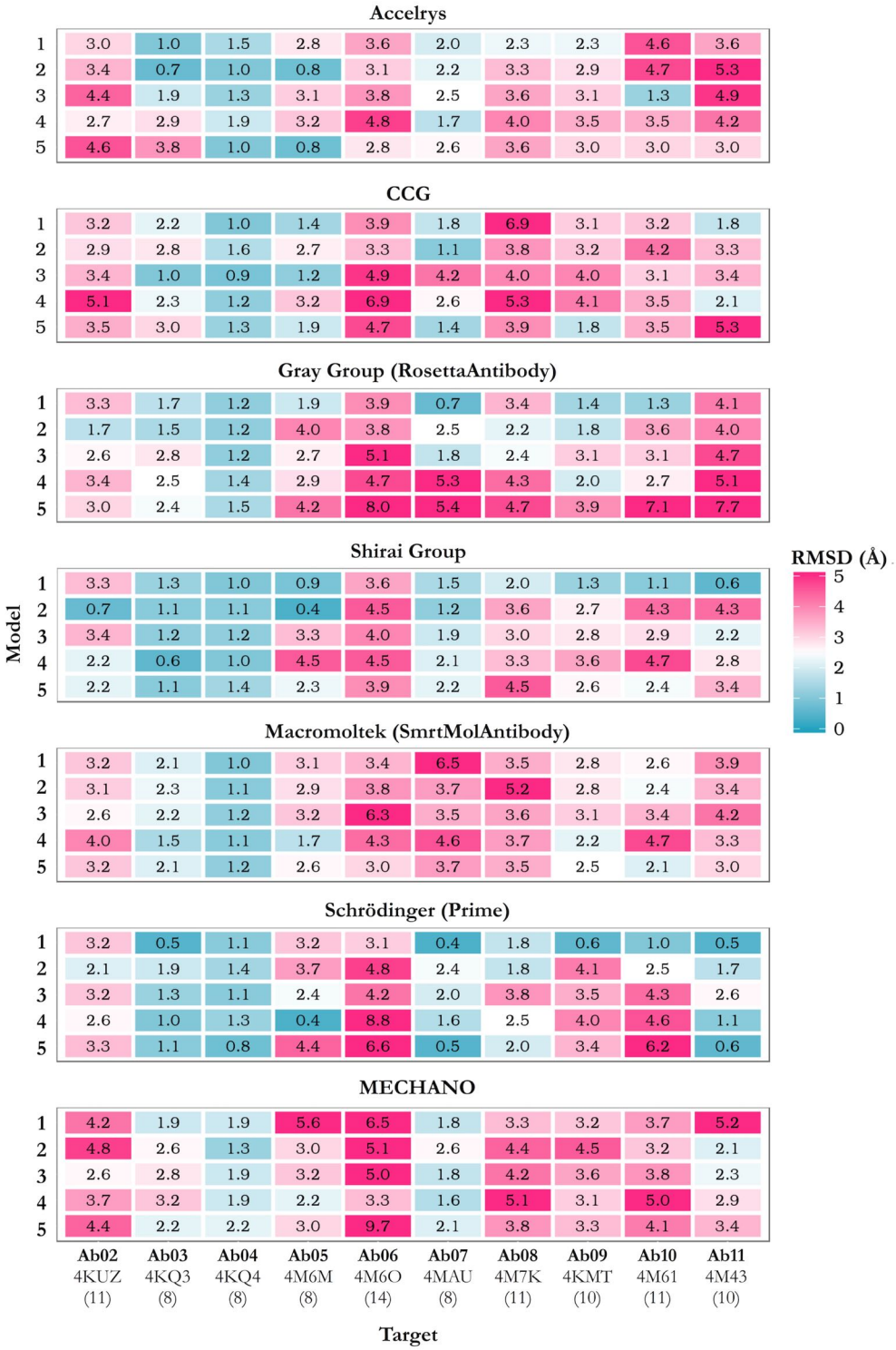

Figure 3.9: Performance of MECHANO on the target set from the Antibody Modelling Assessment II, compared to the actual participants. PDB codes and loop lengths are given underneath the target ID. RMSDs shown are carbonyl RMSDs, as reported in the paper by Teplyakov *et al.* (2014).

results to those of other groups. The results of this competition are reported using carbonyl RMSDs; these are calculated using only the carbonyl carbon and the oxygen of the loop backbone over residues numbered 95 to 100*x* (the one before residue 101). We used MECHANO to generate 5000 decoys per target, added sidechains using SCWRL (Krivov *et al.*, 2009) and ranked the decoys with dDFIRE (Zhang *et al.*, 2004). Taking into account the five top-ranked predictions for each target, the average RMSD achieved by MECHANO was 3.45 Å. When only considering the best of the top five predictions, this falls to 2.47 Å. This is worse than all other participants of the assessment. However, if the ranking system was able to select the best prediction each time, the average RMSD becomes 1.54 Å, which is better than FREAD (2.03 Å) and four of the actual participants.

3.4 Conclusions

We have developed a novel method for the prediction of H3 structures. Both the overall results and the RMSD distributions are improved by using clash checks, resampled H3 dihedral data, 'switching', and a modified CCD algorithm that restrains ϕ/ψ angles to realistic values. These additions increase the efficiency of the conformational search within MECHANO, directing it towards near-native conformations.

We have tested other loop modelling algorithms on the same test set as MECHANO for comparison, and have also compared MECHANO to leading H3-specific algorithms by using the test set from the AMA-II. MECHANO performs reasonably well, especially compared to non-specific algorithms, but is let down by the ranking method. More research is needed to find or develop an appropriate ranking method that can consistently choose the best decoys.

Since *ab initio* methods do not use pre-observed structures to generate decoys, they are not restricted to producing conformations that have already been seen, and can therefore make decoys with novel conformations. This is especially important for H3 loop prediction, since their variability makes them structurally diverse. It is therefore

likely that a new antibody sequence has a novel structure. However, if the target loop structure is similar in any way to another one for which the structure has been solved, *ab initio* methods are unable to use this information. In the next chapter, we describe a novel hybrid loop modelling algorithm, Sphinx, which can combine knowledge-based and *ab initio* methodologies — structural data from protein fragments is used in conjunction with *ab initio* techniques, to take advantage of each traditional approach and thereby improve H3 loop prediction.

Pre-Chapter Image: HIV-1 protease, whose active site features several loops, bound to the antiretroviral drug ritonavir.

4

Sphinx: A hybrid method for loop decoy generation

Contents

4.1 Introduction

The majority of the loop modelling algorithms currently available can be categorised as knowledge-based or *ab initio*, depending on how decoys are generated — knowledge-

based methods use fragments of known protein loop structures, while *ab initio* methods generate new conformations from scratch computationally. There are several drawbacks to both approaches. The number of experimentally-determined structures available is small compared to the vast number of possible conformations (Lee *et al.*, 2009). This means that, since knowledge-based methods are limited to using previously-observed structures for their predictions, if a target loop has a novel structure it is impossible to achieve a high accuracy prediction in this way. This is often the case for long loops, which have a larger number of possible structures than those of just a few residues, with very little of this conformational space covered in the PDB (Totrov, 2012). In some cases, knowledge-based methods fail to predict at all (Choi and Deane, 2010).

Conversely, *ab initio* methods build loop decoys from scratch, and are therefore free to explore areas of conformational space that have not been observed experimentally. However, to explore this space, a large number of possible structures must be made, which makes such algorithms computationally expensive. In addition, since the number of possibilities for long loops is larger, accuracy is often length-dependent, with long loops predicted poorly. If a near-native structure is available in the PDB, algorithms using this approach cannot use the information it contains. The decision about which approach to use, therefore, is often an important one.

An alternative is to use a hybrid method — a combination of knowledge-based and *ab initio* approaches. So far, two ways of combining them have been explored: using the two approaches separately and merging the two decoy sets before ranking (*e.g.* CODA, see Chapter 1 for details); or dividing a given loop into two sections and modelling one with each approach. This latter technique has been applied to the problem of H3 prediction by Martin *et al.* (1989) and Whitelegg and Rees (2000), as well as during AMA-II by the Accelrys group (Fasnacht *et al.*, 2014); the 'stem' regions of the loop (the residues nearest to the anchors) are chosen from a database and the middle section is modelled *ab initio*. In all of these examples, however, the knowledge-based and *ab initio* algorithms are kept quite separate, and in the latter case human input is required to decide where to divide the loop. An alternative approach using Rosetta was described by Rohl *et al.* (2004) — this used a Monte Carlo-based fragment assembly method,

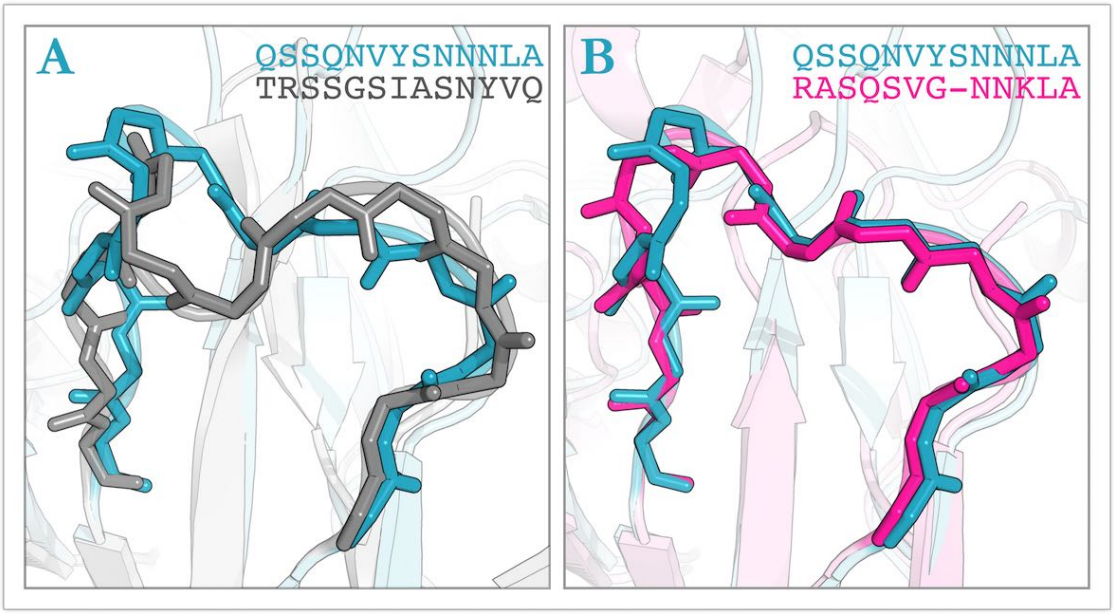

Figure 4.1: Some loops may have a different number of residues, but similar structures. The L1 loop of antibody structure 4JO2 (blue) is 13 residues in length. Its closest structural neighbour with the same number of residues is the L1 loop of structure 3BDX, with an RMSD of 2.8 Å (grey). A structure with lower RMSD can be found by looking at structures of a different length — the closest found for this example is the L1 loop of structure 3LHP (pink), producing an RMSD of 1.3 Å. This shorter loop is also more similar in sequence. Example taken from Nowak *et al.* (2016).

in conjunction with a minimisation protocol. The fragments used by this algorithm are very short (three residues for loops under 15 residues in length) and therefore any structural similarity is only considered for short segments of the target loop, not all of it.

Recent research into the canonical conformations of antibody CDRs has shown that loops of different lengths can adopt similar structures (Nowak *et al.*, 2016). For example, in the case of the L1 loop of the antibody 4JO2, the closest structure amongst L1 loops of the same length (13 residues) has an RMSD of 2.8 Å. However, there is a loop of 12 residues that has a more similar conformation, with an RMSD of 1.4 Å. These structures are shown in Figure 4.1. This result indicates that loops with a different number of residues could still provide useful information when modelling a particular target loop. However, no existing methods are able to use this knowledge, since they either do not use previously known structures, must only use length-matched fragments, or build conformations by assembling very short fragments.

Here we introduce a new loop modelling method, Sphinx, that can use this extra

source of structural information by integrating aspects of both knowledge-based and *ab initio* approaches in a novel way. Sphinx begins the loop modelling process by identifying potential fragments that are a different length to the loop to be predicted. By combining the structural information from such fragments with *ab initio* techniques, the length is adjusted and decoys of the correct length are generated. In this way, we are able to use the additional information present in different length loops. Sphinx is not limited to predicting what has already been observed, like a traditional knowledge-based method, nor does it ignore the available structural information, as would be the case using a purely *ab initio* approach. Our algorithm gives a prediction in every case (unlike some knowledge-based methods) and generates decoy sets that are enriched with near-native conformations.

4.2 The Sphinx Algorithm

The Sphinx algorithm is a combination of FREAD (Deane and Blundell, 2001; Choi and Deane, 2010), a knowledge-based method developed within our research group (see Chapter 2) and our own novel *ab initio* algorithm, MECHANO (described in Chapter 3).

A flowchart of the Sphinx decoy generation procedure is shown in Figure 4.2. The inputs are a protein structure or model (in PDB format), and the location and sequence of the loop to be modelled (*i.e.* chain identifier, and numbers of the start and end residues). The algorithm includes four main steps: database search, loop building, loop closure and clash checking. Here and throughout this thesis, we define the H3 loop region according to Chothia (Chothia and Lesk, 1987). In the Chothia numbering scheme the H3 is found on the heavy chain between residues 95 and 102 inclusive.

4.2.1 Database Search

The first step of the Sphinx algorithm is the search of a fragment database. This search is based on the FREAD algorithm (described in Chapter 2), which selects possible

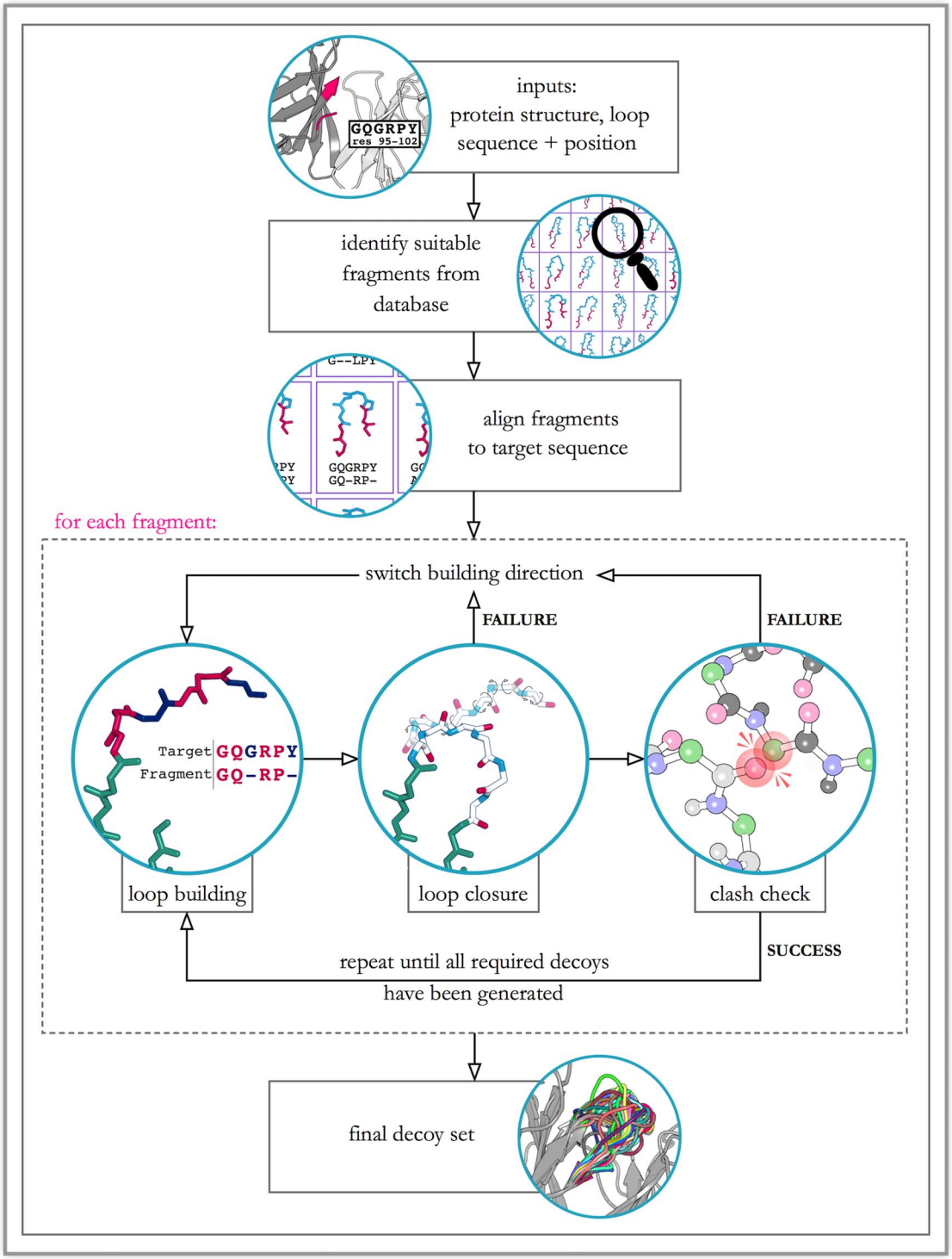

Figure 4.2: A flowchart showing the decoy generation process used by Sphinx.

fragments based on anchor geometry and sequence similarity.

The database initially used in this study was the 'All SAbDab' database that was created for the work reported in Chapter 2. To create this database, we downloaded the structures of all the antibodies in the Structural Antibody Database, SAbDab (Dunbar *et al.*, 2014), which is a collection of all antibody structures that have been deposited in the PDB. These structures were split into all possible fragments of 3 to 30 residues in length. This led to a database containing fragments of 1,804 different antibody PDB structures, with 1,117 and 956 unique heavy and light chain sequences, respectively.

Fragments from the database are selected as potential starting points for prediction depending on three criteria:

1. length — the idea behind Sphinx is that there is potentially useful structural information contained within loops of a different length to the target, since loops with different numbers of residues have been shown to adopt similar conformations. We therefore exclude fragments that are the same length as the target, since these could be found using a normal knowledge-based approach.

2. the anchor geometry of the fragment must match that of the target. As is the case in the FREAD algorithm, the distances between the Cα atoms of the anchor residues (the two residues on each side of the loop) are used to select fragments with similar anchors that will therefore fit into the gap in the structure.

3. sequence similarity — the sequences of fragments that meet the first two criteria are aligned to the target sequence using the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). These alignments are scored using an environment-specific substitution (ESS) score (a score which takes into account the dihedral angles of the fragment residues), as implemented in the original version of FREAD (Deane and Blundell, 2001; Hill *et al.*, 2011). We select fragments with a score of over 25; this is the default cutoff used by FREAD.

Those fragments that are suitable according to these criteria are then passed to the loop building stage of the algorithm.

Sequence Alignment using the Needleman-Wunsch Algorithm

The sequences of the fragments within the database are compared to the target sequence using alignments, where residues are matched depending on their similarity. Regions of sequence similarity may correspond to sections of the protein that are alike in structure. This means that the alignment can act as a guide for loop building, indicating where structural information from a fragment should be used.

In Sphinx, we use the global alignment algorithm developed by Needleman and Wunsch (1970). This is an example of dynamic programming, where a larger problem (the alignment of a whole sequence) is broken down into smaller ones, the solutions to which are used to determine the overall answer. An outline of the algorithm is given in Figure 4.3.

4.2.2 Decoy Set Generation

The structure and sequence alignment information of each selected fragment is then used to build a set of loop decoys. Each decoy is built, one residue at a time, onto one of the anchor residues, using the algorithm described by Parsons *et al.* (2005).

For target residues that are paired with a fragment residue in the sequence alignment, the structural data (bond lengths, bond angles and dihedral angles) used to build the residue are calculated from the corresponding fragment residue. For target residues that are not aligned to a fragment residue, the structural data is randomly selected from a set of purpose-built distributions, like the *ab initio* algorithm MECHANO (see Chapter 3). Bond lengths and angles are drawn at random from the Gaussian distributions parameterised by Engh and Huber (1991), and dihedral (ϕ/ψ) angles are selected from the resampled Ramachandran distributions described in Chapter 3, which combine H3 and general data. As shown in Chapter 3, these distributions produce more near-native decoys than general protein loop distributions. Decoys built using fragments that are shorter than the target loop will always require at least one residue to be built using these distributions; this is not the case for those built using longer fragments.

Continuity of the loop backbone is then enforced using the modified version of CCD that we developed for MECHANO (see Section 3.2.2). This algorithm closes the

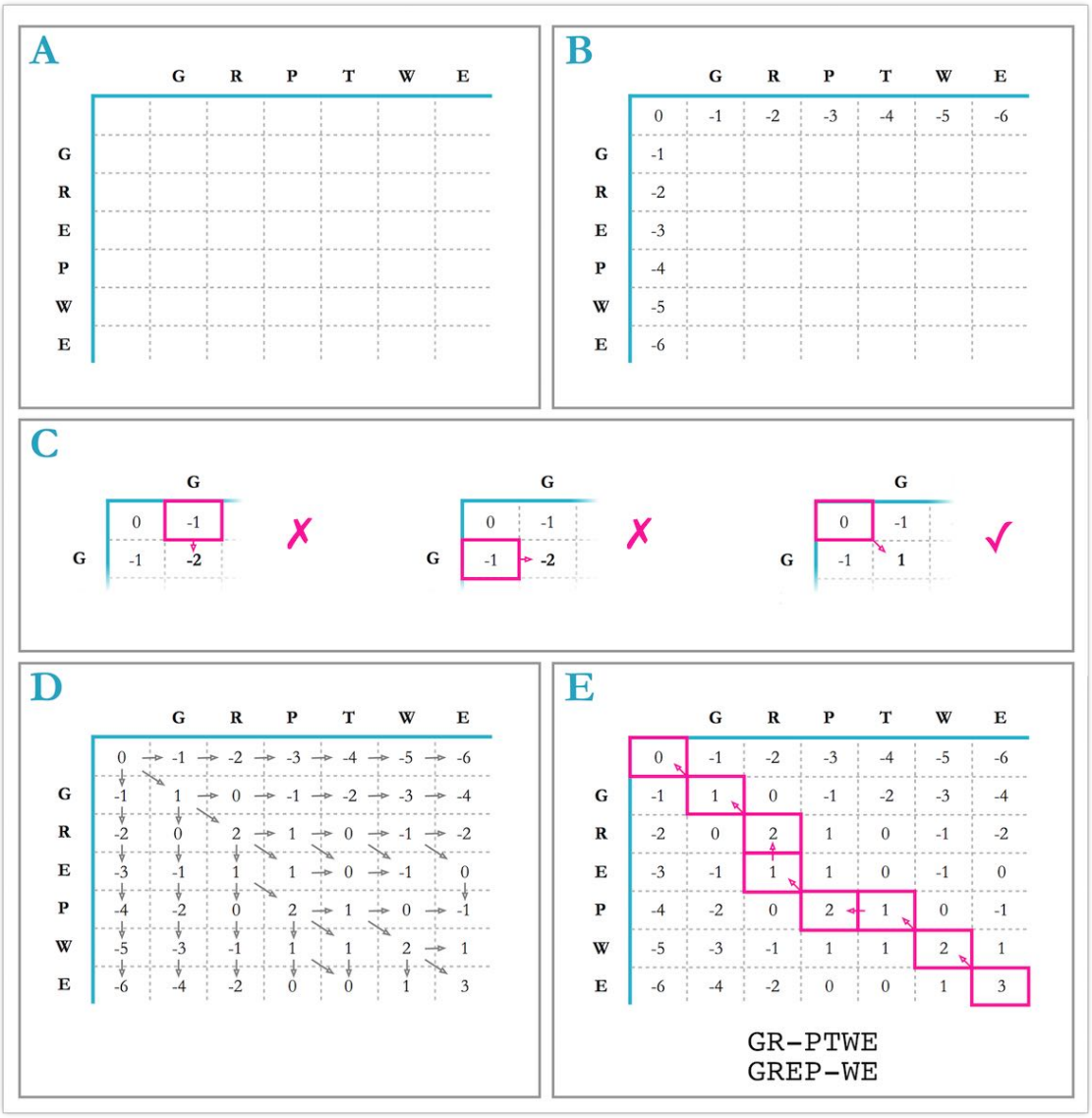

Figure 4.3: An outline of the Needleman-Wunsch alignment algorithm. This example aligns the sequences GRPTWE and GREPWE, using a match score of 1, a mismatch score of -1, and a gap score of -1. A — a grid is initialised, with rows corresponding to the residues of one sequence and columns to residues of the other. B — starting with a value of zero in the top-left cell, the first row and column are filled in. Moving left or down relates to a gap in the sequence, and is therefore given a score of -1. C — for all other cells, there are three possible scores, depending on the three neighbouring cells that are already filled in. The scores leading from the cell above and to the left correspond to a gap; in both cases the score is -2. The score from the diagonal relates to aligning the two residues indicated by the row and column — in this instance the residue types match, and the resulting score is 1. The highest score is selected. D — the rest of the grid is filled in, keeping track of which neighbouring cell the scores resulted from. E — the maximum alignment score is in the bottom-right cell. By tracing back through the grid, the optimal alignment can be found.

loop by iteratively changing the ϕ/ψ dihedral angles, but is restricted from changing these angles to disallowed values (according to the Ramachandran plot; see Figure 1.4). As we demonstrated for MECHANO, this approach forces every conformation to be physically allowed in terms of ϕ/ψ space, leading to higher-quality decoy sets (Figures 3.4 and 3.5).

Following loop closure, the resulting decoy structure undergoes a clash check. Using the values for atomic radii established by Tsai *et al.* (1999) and a hard-sphere approximation, if any atoms in the loop decoy are calculated to be too close to either atoms in the loop itself or the rest of the protein structure, the decoy is discarded. Like MECHANO, Sphinx implements 'switching' (Chys and Chacon, 2013); if loop building fails for some reason (*i.e.* because of clashes or failure to close the loop successfully), the next loop decoy is built from the opposite anchor residue to the one previously used.

Once the required number of decoys has been made from a fragment (or the number of failures reaches a cutoff value), the algorithm moves onto the next fragment until all selected fragments have been used. For fragments that led to one or more residues being built using an *ab initio* approach (*i.e.* those shorter than the target), we initially decided to make 150 decoys, since there is a random element to the building procedure. Otherwise, it is only necessary to make one decoy from a fragment, since there is no stochasticity involved — repeating the procedure would therefore simply lead to the same structure.

4.2.3 Implementation

Sphinx is written in the programming languages Python (for the database search aspects of the algorithm) and C (for the actual decoy generation steps), and has over 13,000 lines of code. As input it requires: the path to the desired protein fragment database, the path to the target protein structure (in PDB format), the chain identifier of the chain within the target file that contains the target loop, the sequence of the loop, the length of the loop, the start and end residues, and the location of the directory in which to place all results files.

4.3 Target Sets

During the development of Sphinx, we used four target sets: Preliminary Sets 1 and 2, a training set, and a test set. Preliminary Set 1 contained five H3 loops, each of ten residues in length, which were randomly selected from the set of all H3 loops in SAbDab with 50% sequence identity or less. Preliminary Set 2 contains the five loops from Preliminary Set 1, with an additional four loops of length 14 and five of length 16. These loops were chosen in the same way as Preliminary Set 1, but were additionally selected to contain some loops that FREAD failed to predict. These preliminary sets are of a small size since the initial version of Sphinx was computationally very expensive.

For the training and test sets, a set of structures was extracted from SAbDab that were determined using X-ray crystallography, and with a maximum sequence identity of 95%. These were then grouped by H3 loop length. Where possible, we also implemented an average B-factor cutoff across the H3 loop backbone atoms of 40 (this was not possible for 17-residue loops due to the limited number of examples). For the training set, which was used to decide upon various parameters used by Sphinx, we randomly selected ten loops for each even loop length from 6 to 18 residues. For the test set, which we used to evaluate the performance of Sphinx, we randomly chose ten loops for lengths 6-13, 15 and 17. The training set therefore contained 70 loops, and the test set contained 90.

As a further test, we ran Sphinx on the set of 53 targets used to benchmark the performance of RosettaAntibody, building loop decoys onto both crystal and model structures (Sivasubramanian *et al.*, 2009). The target 2AI0 was removed from the original set since the structure is incomplete and was previously incorrectly designated as a 6-residue loop. Complete lists of targets are again given in the Appendices (Tables A.4 to A.8).

4.4 Preliminary Results

As a preliminary test, we used Sphinx to generate decoys for the five ten-residue H3 loops of Preliminary Set 1. We also ran Sphinx's constituent algorithms, FREAD and MECHANO, on these targets for comparison. In the case of both Sphinx and FREAD, self-prediction was prevented by ignoring fragments from chains with identical sequences to the chain on which the target loop is located. This means that fragments from both the unbound and bound forms of the antibody, if present, are not considered during the loop building process. This is important, since in a real modelling situation these structures would not be available.

Figure 4.4 shows the best RMSDs that were achieved for each fragment length. The average best RMSD across all fragment lengths achieved for these five targets was 0.77 Å, which is similar to the accuracy of FREAD (0.74 Å) and much better than MECHANO (1.54 Å). For every target, Sphinx produced a better decoy than generated by MECHANO, and in three cases out of five, the best Sphinx decoy was closer to the native than the best from FREAD. This includes one target for which FREAD failed to find a suitable fragment in the database (2XZC) — Sphinx was able to produce a decoy with sub-Ångström accuracy (an RMSD of 0.9 Å).

It can be seen from Figure 4.4 that the best decoys are made using fragments that are shorter than the target loop. All decoys in this case have at least one residue built from distributions. On average, the RMSD decreases as the length of the fragment increases — the best loop decoys were made (on average) from fragments in the database with nine residues, leaving one residue to be modelled from Ramachandran distributions using MECHANO. Fragments that are longer than the target loop, on the other hand, do not produce such near-native structures — on average, the best decoy made in this way is worse than the best decoy generated by MECHANO.

Since the best decoys were made from fragments shorter than the target loop, we decided that all fragments that are longer than the target should be ignored during the prediction process. We ran Sphinx on more H3 targets (Preliminary Set 2 — the five loops from before, plus four of length 14 and five of length 16), this time only

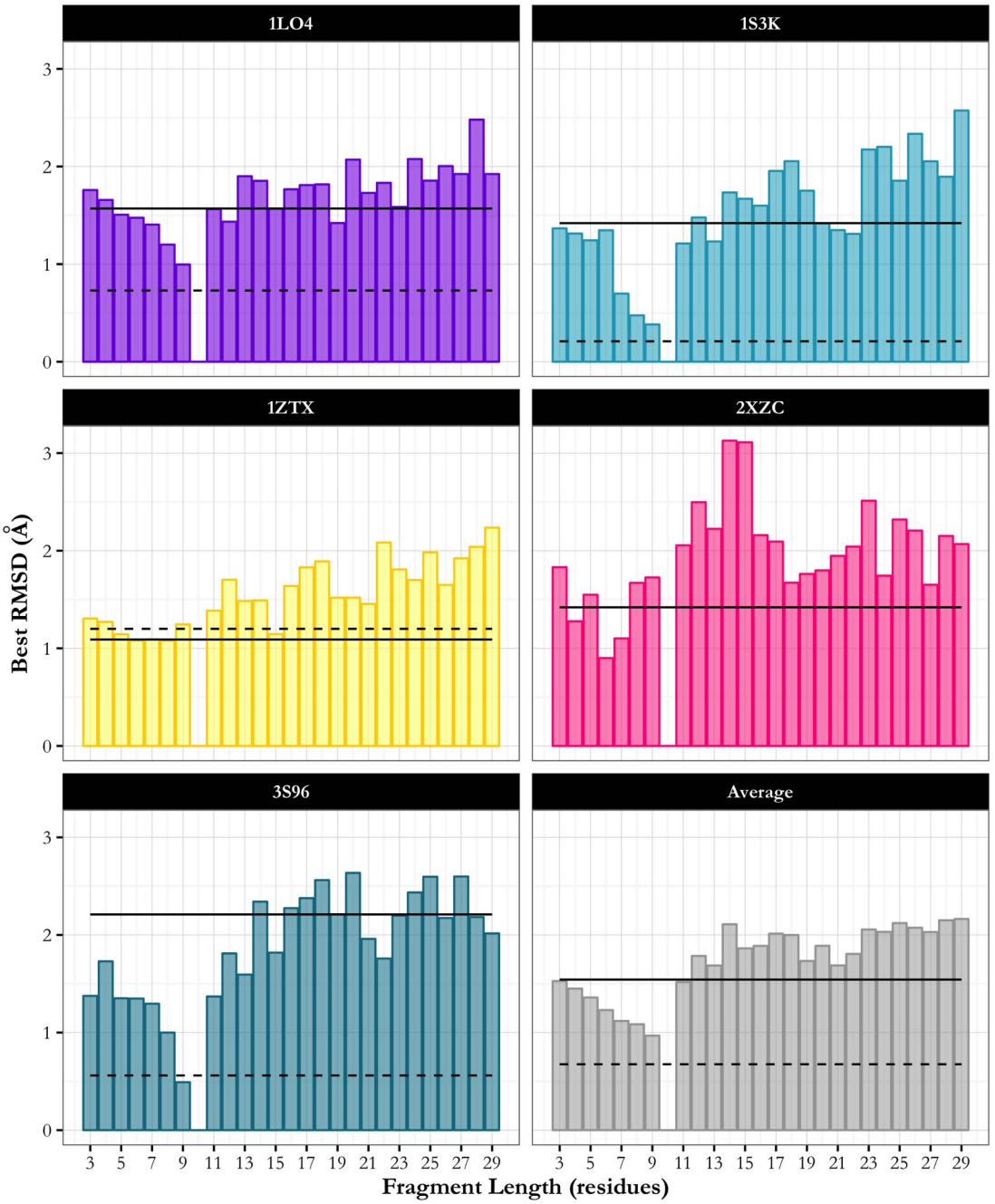

Figure 4.4: Results of using the Sphinx hybrid algorithm to predict the structures of five H3 target loops of length 10 (Preliminary Set 1). RMSDs reported are those of the best decoy generated from all fragments, for each fragment length. The RMSD of the best decoy generated by FREAD and MECHANO are indicated by the dashed and solid lines, respectively.

selecting fragments from the database that are between one and five residues shorter than the target. Some of these targets are loops for which FREAD was unable to make a prediction. The RMSDs of the best decoys generated are shown in Table 4.1. Sphinx produces excellent results; for every target of length 14 or 16, the best decoy Sphinx generated was closer to the native structure than any decoy produced by using FREAD or MECHANO alone. Good results are achieved even where FREAD failed; in fact, loop decoys of under 2 Å are produced for every target.

4.5 Reducing the Number of Decoys

The aim of the Sphinx algorithm is to generate a small decoy ensemble that is enriched with near-native conformations — this should make subsequent decoy selection steps easier, since the number of decoys to score will be small, and a high number of low-RMSD structures leads to a higher chance of one being selected as the final prediction. However, since Sphinx is not limited to using loops of the same length as the target, and 150 decoys have been generated per target, the number of decoys generated so far is very large — over a million in some cases. There are several aspects of the algorithm that we investigated in order to reduce the size of the decoy set, whilst maintaining its quality:

- the choice of database;
- the number of decoys generated from each fragment;
- the number of fragments to be used in loop building;
- which length fragments should be used;
- the addition of a fast scoring function that removes poor decoys from the set.

Table 4.1: Accuracies achieved using the Sphinx hybrid algorithm to predict H3 loops (five of length 10, four of length 14 and five of length 16). Results shown are the RMSDs of the best decoys produced for each target loop, for each fragment length (up to five residues shorter than the target). Also shown are the results achieved when using FREAD and MECHANO independently. The RMSD of the best decoy produced by the hybrid method is underlined, while the best RMSD of all the methods is highlighted in pink.

	Fragment Length	Best RMSD (Å)					
		1LO4	**1S3K**	**1ZTX**	**2XZC**	**3S96**	**Average**
Target Length 10	5	1.51	1.24	1.14	1.55	1.35	1.36
	6	1.48	1.35	1.08	0.90	1.35	1.23
	7	1.40	0.70	1.09	1.10	1.30	1.12
	8	1.20	0.48	1.08	1.67	1.00	1.09
	9	1.00	0.38	1.25	1.73	0.49	0.97
	Sphinx (all lengths)	1.00	0.38	**1.08**	**0.90**	**0.49**	0.77
	FREAD	**0.73**	**0.21**	1.20	-	0.56	**0.74**
	MECHANO	1.57	1.42	1.09	1.42	2.21	1.54
			1DQL	**1YEE**	**2DQU**	**4DAG**	**Average**
Target Length 14	9		2.30	1.46	1.70	1.28	1.69
	10		2.03	0.93	1.84	1.39	1.55
	11		1.97	0.72	1.62	1.40	1.43
	12		2.21	0.62	1.82	1.87	1.63
	13		2.92	0.46	1.85	2.29	1.88
	Sphinx (all lengths)		**1.97**	**0.46**	**1.62**	**1.28**	**1.33**
	FREAD		6.46	0.78	2.90	-	3.38
	MECHANO		2.92	2.48	2.29	1.93	2.41
		2HFG	**3EYF**	**3SO3**	**4HHA**	**4JDV**	**Average**
Target Length 16	11	1.77	1.75	1.93	1.18	2.02	1.73
	12	1.85	1.89	1.94	1.66	1.98	1.86
	13	2.04	1.98	2.05	1.93	1.96	1.99
	14	1.52	1.72	2.41	1.12	2.03	1.76
	15	2.66	1.99	2.98	1.03	2.04	2.14
	Sphinx (all lengths)	**1.52**	**1.72**	**1.93**	**1.03**	**1.96**	**1.63**
	FREAD	-	-	-	1.23	3.03	2.13
	MECHANO	2.60	2.27	2.81	2.23	2.58	2.50

Table 4.2: A comparison of the performance of Sphinx when using the 'All SAbDab' and 'H3 only' databases (labelled 1 and 2 respectively), to predict the H3 loops in Preliminary Set 2. Results shown are the RMSDs of the best decoy produced for each fragment length (up to five residues shorter than the target), the number of fragments used, the number of decoys generated, and the execution time.

Target Length 10

		Best RMSD (Å)											
Target		1LO4		1S3K		1ZTX		2XZC		3S96		Average	
Database		1	2	1	2	1	2	1	2	1	2	1	2
Fragment Length	5	1.51	1.66	1.24	1.40	1.14	1.24	1.55	1.33	1.35	1.44	1.36	1.41
	6	1.48	1.05	1.35	1.09	1.08	1.06	0.90	1.41	1.35	1.26	1.23	1.17
	7	1.40	0.73	0.70	0.62	1.09	1.08	1.10	1.00	1.30	1.27	1.12	0.94
	8	1.20	0.82	0.48	0.61	1.08	0.85	1.67	1.16	1.00	0.75	1.09	0.84
	9	1.00	0.99	0.38	0.46	1.25	0.97	1.73	0.95	0.49	0.56	0.97	0.79
	ALL	1.00	0.73	0.38	0.46	1.08	0.85	0.90	0.95	0.49	0.56	0.77	0.71
	#Fragments	948	881	2,344	1,468	3,605	2,106	150	16	1,763	1,734	1,762	1,241
	#Decoys	88,795	78,993	225,335	206,739	357,586	206,739	14,266	1,541	168,751	166,870	170,947	132,176
	Time (hours)	14.2	13.6	32.8	19.2	29.0	20.2	1.1	0.2	24.0	18.1	20.2	14.3

Target Length 14

		Best RMSD (Å)									
Target		1DQL		1YEE		2DQU		4DAG		Average	
Database		1	2	1	2	1	2	1	2	1	2
Fragment Length	9	2.30	2.06	1.46	1.20	1.70	1.74	1.28	1.54	1.69	1.64
	10	2.03	1.77	0.93	0.93	1.84	1.67	1.39	1.42	1.55	1.45
	11	1.97	1.98	0.72	0.68	1.62	1.76	1.40	1.76	1.43	1.55
	12	2.21	2.32	0.62	0.64	1.82	1.96	1.87	2.08	1.63	1.75
	13	2.92	2.04	0.45	0.46	1.85	2.03	2.29	1.96	1.88	1.62
	ALL	1.97	1.77	0.45	0.46	1.62	1.67	1.28	1.42	1.33	1.33
	#Fragments	773	764	2,736	2,157	11,534	4,471	14,256	665	7,325	2,014
	#Decoys	73,354	71,396	268,532	207,296	1,149,786	441,329	1,282,590	64,581	693,566	196,151
	Time (hours)	17.0	18.7	38.3	42.8	99.3	54.0	91.3	10.6	61.5	31.6

Target Length 16

		Best RMSD (Å)											
Target		2HFG		3EYF		3SO3		4HHA		4JDV		Average	
Database		1	2	1	2	1	2	1	2	1	2	1	2
Fragment Length	11	1.77	1.69	1.75	1.83	1.93	2.01	1.18	1.58	2.02	2.04	1.73	1.83
	12	1.85	1.73	1.89	2.00	1.94	1.90	1.66	1.79	1.98	2.10	1.86	1.90
	13	2.04	1.84	1.98	1.52	2.05	2.10	1.93	1.96	1.96	2.08	1.99	1.90
	14	1.52	1.58	1.72	2.17	2.41	2.38	1.12	1.12	2.03	2.01	1.76	1.85
	15	2.66	1.56	1.99	2.19	2.98	2.98	1.03	0.98	2.04	2.14	2.14	1.97
	ALL	1.52	1.56	1.72	1.52	1.93	1.90	1.03	0.98	1.96	2.01	1.63	1.59
	#Fragments	2,441	1,740	3,209	1,313	2,095	1,049	7,315	1,782	3,700	2,993	3,752	1,775
	#Decoys	133,669	83,935	215,199	56,698	182,298	84,460	444,122	54,269	328,847	279,068	260,827	111,686
	Time (hours)	139.22	96.92	195.88	77.65	82.53	45.88	397.62	104.88	135.35	89.87	190.12	83.04

4.5.1 Choice of Database

Firstly, we investigated whether a smaller database could be used, whilst maintaining the high accuracy of prediction. Reducing the number of fragments available to Sphinx should mean that fewer fragments are selected for the loop building stage, leading to a smaller decoy set. We found during our investigations into the FREAD algorithm that a database containing only fragments of H3 loops could be used successfully, improving the average prediction RMSD from 2.48 Å to 2.21 Å for a large set of 691 targets (see Chapter 2). We therefore decided to use this database for Sphinx, with the hope of achieving a similar outcome. To create this database, the structures of all the H3 loops in SAbDab (Dunbar *et al.*, 2014), plus five residues on each side, were extracted and split into all possible fragments of 3 to 30 residues in length. At the time this work was undertaken (April 2015), this led to a database containing all possible fragments of 3043 H3 structures, from 1848 different PDB entries. We re-ran Sphinx on Preliminary Set 2 using this fragment database; the results of this are given in Table 4.2.

While the results for individual target vary, overall the prediction accuracy remains approximately the same — 0.71 Å, 1.33 Å and 1.59 Å for lengths 10, 14 and 16 respectively for the H3-only database compared to 0.77 Å, 1.33 Å and 1.63 Å for the All SAbDab database. In addition, the number of fragments selected, and subsequently the size of the decoy set, is reduced considerably by using the smaller database. On average, the number of decoys is decreased from 352,366 to 143,137, and in some cases can be reduced to 10% of its former value (*e.g.* 4DAG).

A further consequence of decreasing the number of fragments that enter the loop building procedure is that the execution time is reduced — on average this is reduced by more than half, from 93 hours to 44.

4.5.2 Decoys Per Fragment

To decide on the optimal number of decoys to make from each fragment, we ran Sphinx on our 70-loop training set, making 150 decoys from each fragment that passed an anchor geometry filter and have an alignment score of over 25 (see Methods for details). On average, there were 6000 fragments per target. For all target lengths, as the number of

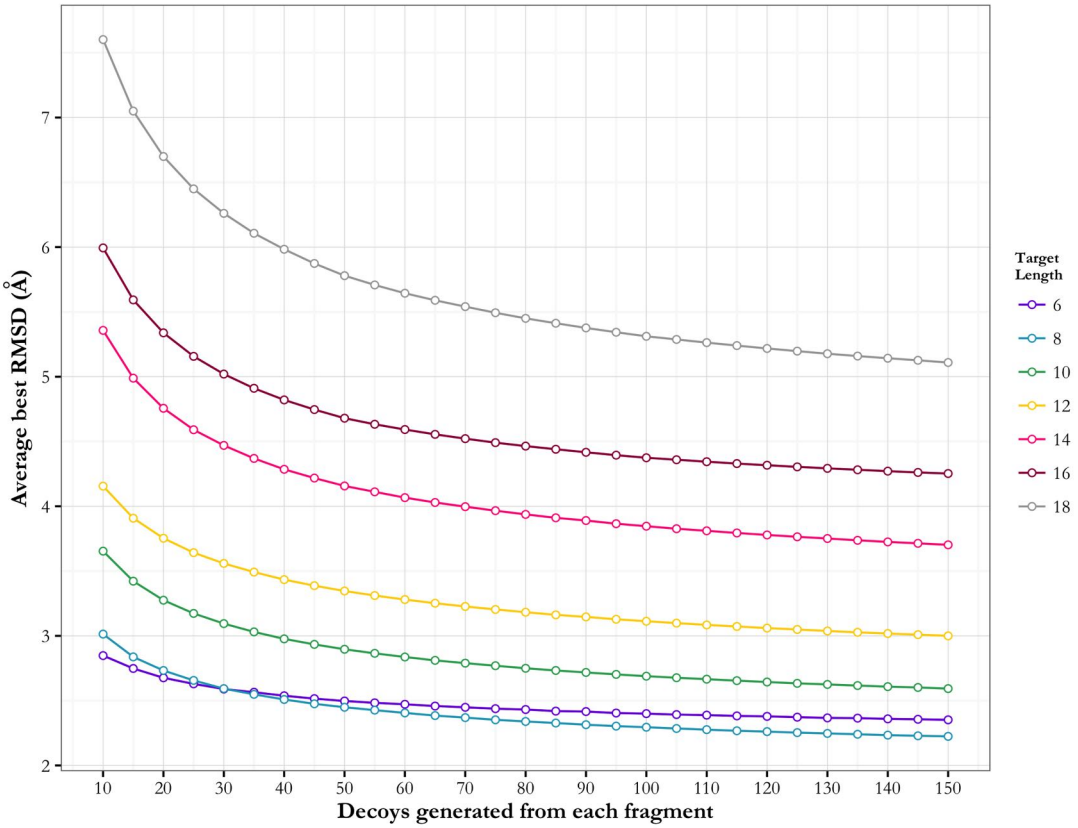

Figure 4.5: Dependence of the average best RMSD on the number of decoys generated from each fragment. Result shown is the RMSD of the best decoy produced by a fragment, averaged over all fragments (those that pass the anchor geometry and alignment score filters). As the number of decoys made from a fragment increases, the average RMSD of the best decoy generated decreases. Minimal improvement occurs after 100 decoys are made.

decoys made per fragment increases, there is an initial marked improvement in the best RMSD, after which the rate of improvement decreases (see Figure 4.5). We decided to generate 100 decoys per fragment, since for all target lengths examined there is minimal improvement in the best RMSD beyond this point.

4.5.3 Number of Fragments and Minimum Fragment Length

We next investigated how many and what length fragments should be selected for use in loop building. Fig. 4.6 shows how the RMSD of the best decoy generated by Sphinx varies as the number of fragments selected and the minimum fragment length are changed. These results were generated using the data from the previous test, selecting 100 decoys from those made from each fragment. For a given minimum fragment length, increasing

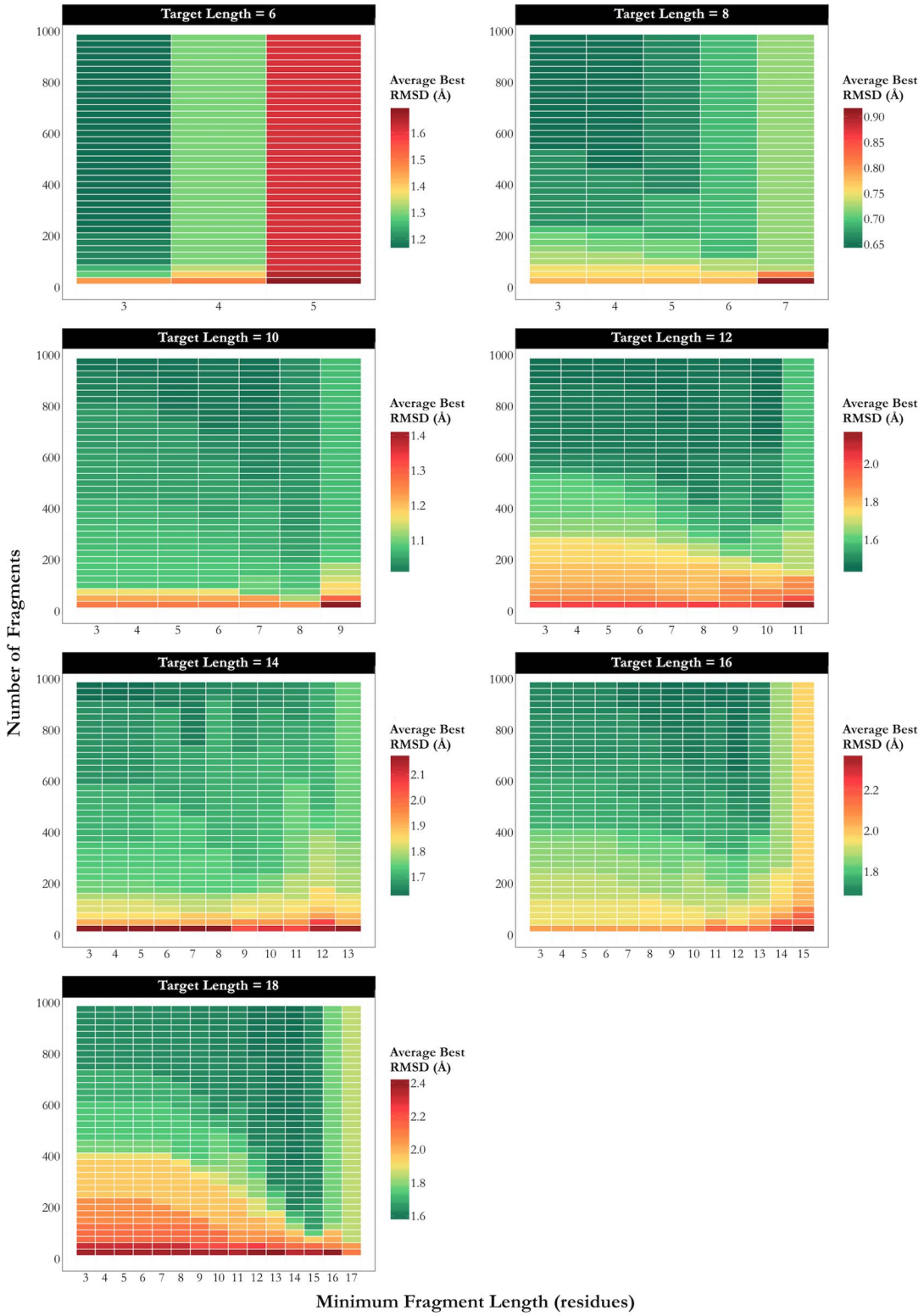

Figure 4.6: Dependence of the average best RMSD on the number of fragments used and the minimum fragment length, for each target length present in the training set. Increasing the number of decoys always improves the performance of the Sphinx algorithm, while the optimum minimum fragment length is 3 or 4 residues shorter than the target, depending on loop length.

the number of fragments always improves the average best RMSD, however for long loop lengths more fragments must be used to gain results similar to those achieved for short loops. The results for fragments of length $n-1$ (or $n-2$ for 16- and 18-residue targets), where n is the target length, are not as good as shorter fragments. This is possibly because, in these cases, only one or two residues are modelled *ab initio*, leading to reduced freedom of movement during decoy generation. Decoys are therefore unable to move too far away from the structure of their fragments. By examining these data, we decided upon the following set of parameters that achieve good results whilst minimising the number of decoys that need to be made and considered in subsequent steps:

$$\text{minimum target length} = \begin{cases} n-3 & \text{if } n \leqslant 12 \\ n-4 & \text{if } n > 12 \end{cases}$$

$$\text{number of fragments} = \begin{cases} 25(n-2) & \text{if } n \text{ is even} \\ 25(n-1) & \text{if } n \text{ is odd} \end{cases}$$

These parameters have been used for the remainder of this work.

4.5.4 Reduced Set Selection

To reduce the number of decoys further, we have developed our own H3-specific, knowledge-based energy function that quickly scores each decoy. The aim is to distinguish between high- and low-RMSD conformations, so that the poor structures can be removed from the set. Knowledge-based energy functions score decoys by comparing their features to experimentally-determined structures, based on the assumption that the observed distributions of particular features reflect energetics: *i.e.*, a common characteristic is assumed to be energetically favourable. The function we have developed for Sphinx is based upon the RAPDF (residue-specific all-atom probability discriminatory function) proposed by Samudrala and Moult (1998). The score for each decoy is determined by calculating the pairwise distances between the atoms of

the decoy and the rest of the protein, and comparing these distances to those seen in experimentally-determined antibody structures:

$$S = -\sum_{ij} \ln\left(\frac{P(d_{ab}^{ij}|C)}{P(d_{ab}^{ij})}\right)$$

where S is the score for the decoy, and d_{ab}^{ij} is the distance between atoms i and j, of types a and b. Distances are discretised into bins; the probability of observing a distance in a particular bin d between atom types a and b in native conformation C is given by:

$$P(d_{ab}|C) = \frac{N(d_{ab})}{\sum_d N(d_{ab})}$$

where $N(d_{ab})$ is the number of observations between atom types a and b in bin d. The denominator here is therefore the total number of observations of atoms a and b. Also required is the probability of observing atom types a and b in bin d, in any structure:

$$P(d_{ab}) = \frac{\sum_{ab} N(d_{ab})}{\sum_d \sum_{ab} N(d_{ab})}$$

i.e. the total number of observations in bin d, regardless of atom type, divided by the total number of observations over all bins.

Unlike the original RAPDF, which includes all 167 possible atom types, our version considers only the atoms of the backbone (N, Cα, C and O) and Cβ atoms, for each residue type (except glycine, which has no Cβ atom). This means that the time-consuming process of sidechain addition does not have to be carried out. We only consider the distances involving the atoms of H3 loop, since the remaining structure does not change and the calculated distances will therefore be the same for every decoy. The observation data required for this function was obtained from a non-redundant set of antibody structures (with a maximum sequence identity of 95%), considering only those pairwise distances involving H3 loop atoms.

The original RAPDF used 18 bins; 1 bin for distances between 0-3 Å, followed by seventeen 1 Å bins up to a maximum of 20 Å. We have investigated the optimum number of bins by varying the maximum distance included in the score calculation; the results of this investigation are shown in Figure 4.7. We ranked the decoys according to

their scores, and monitored the RMSD of the best decoy left in the set as increasingly large selections are taken. From these results, we decided to use six bins (maximum length of 8 Å, one bin for distances between 0-3 Å, followed by five 1 Å bins), since the average best RMSD in the 'reduced set' tends to be lower.

We have also developed a scoring function that does not use H3-specific data for comparison. The data for this function was calculated from a non-redundant set of protein structures (with a maximum sequence identity of 40% and a resolution cutoff of 2 Å), considering only those pairwise distances involving loop atoms. Loop regions were defined as regions connecting secondary structures of at least three residues in length, using DSSP (Joosten *et al.*, 2011). The same test was carried out as described for the H3 version, and again the optimum number of bins was found to be six.

We then investigated how many decoys should be selected to make up the reduced set. This must be a compromise between choosing enough decoys so that the best remain, whilst minimising the overall number that have to be dealt with in any subsequent ranking steps. Again, we monitored the RMSD of the best decoy in the set as the number of decoys selected is increased (Figure 4.8). From this data, the optimum number of decoys for all target lengths is 500, giving the user a manageable number, whilst ensuring that the final decoy set still contains and is enriched with near-native structures.

4.6 Testing Sphinx

We ran Sphinx on the test set (see Methods) using the parameters established in the previous section. The results achieved by Sphinx, as well as the results achieved by the algorithms on which it is based (FREAD and MECHANO), are shown in Fig. 4.9. The MECHANO algorithm can be thought of as running Sphinx without any fragment information — *i.e.* all bond lengths, angles and dihedral angles are chosen randomly from the relevant distributions — comparisons to its performance therefore indicate the added value of using fragment information. To ensure a fair comparison, MECHANO was

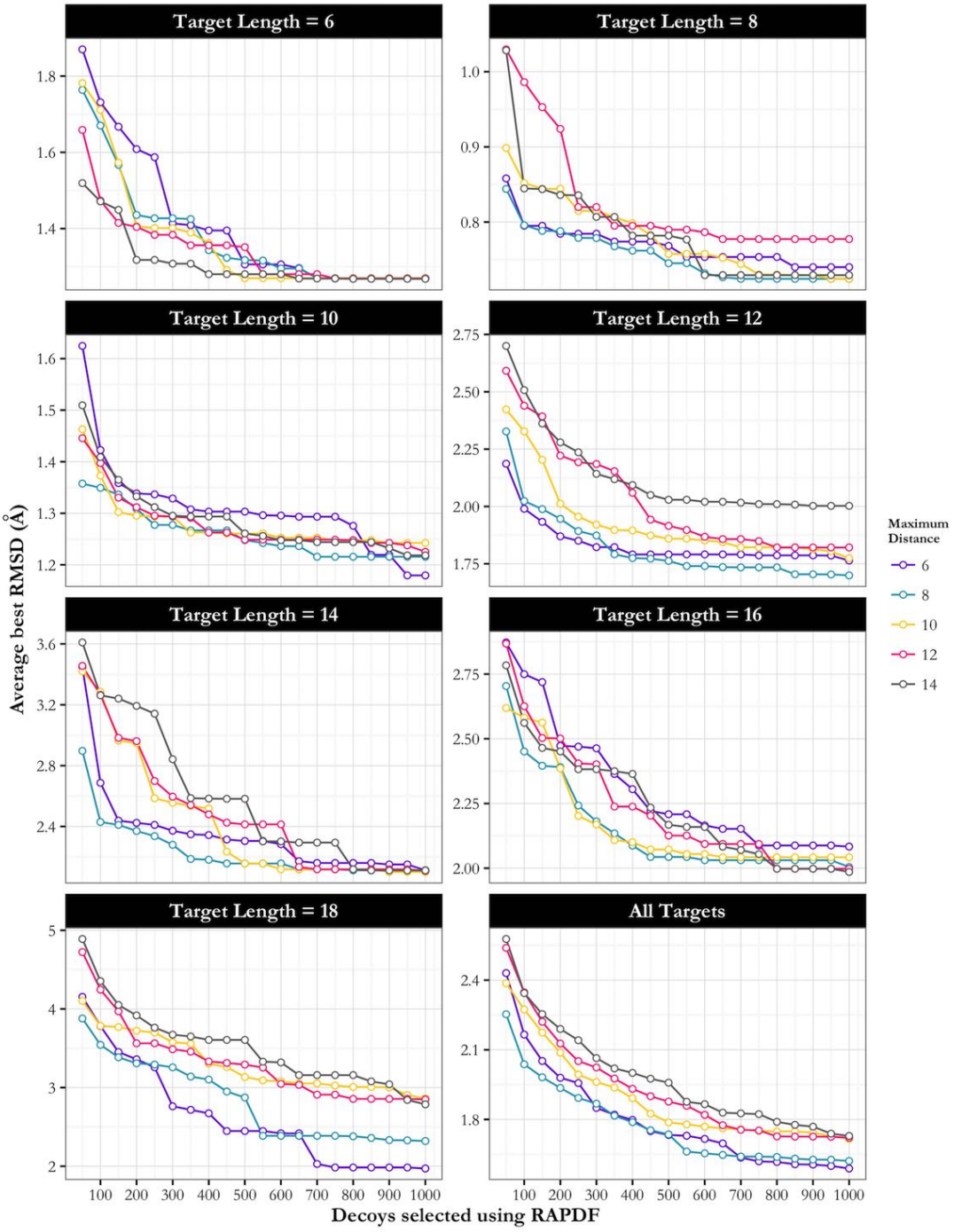

Figure 4.7: The effect of the maximum interatomic distance included in the score calculation on the average RMSD of the best decoy selected, for increasing selection size. A maximum distance of 8Å produces the most consistently accurate results.

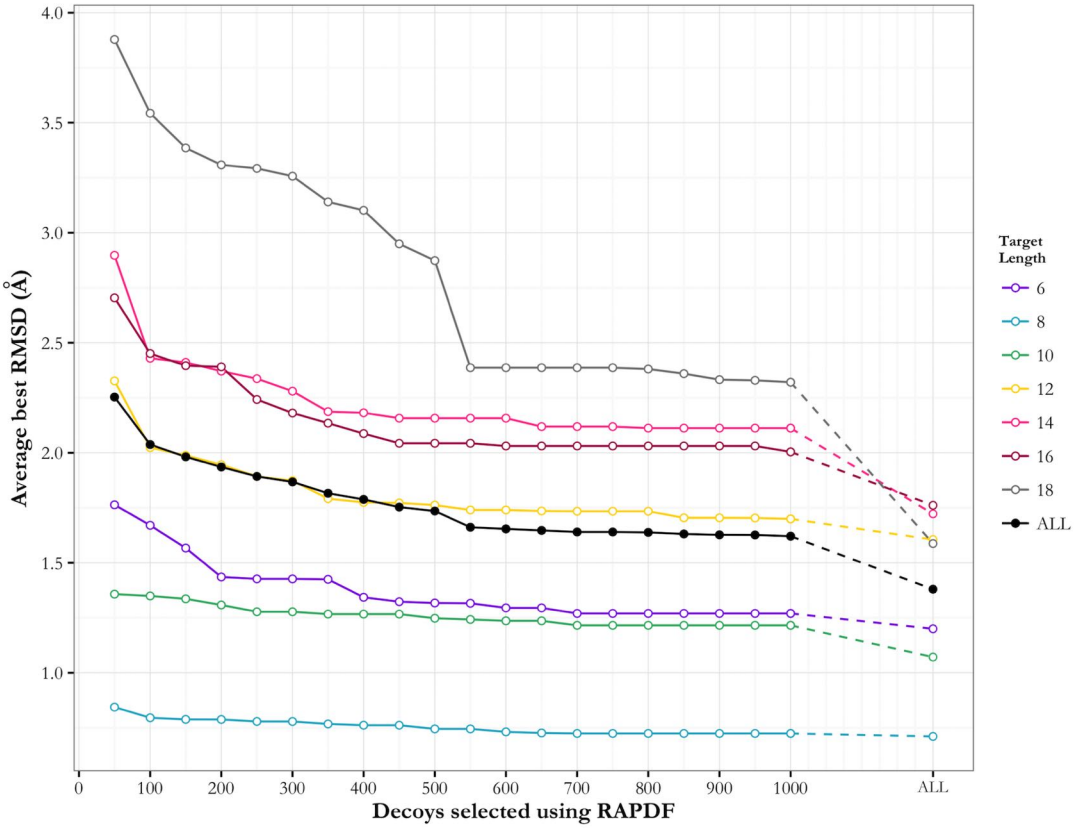

Figure 4.8: Decoy selection using the H3-specific energy function. Choosing the top 500 decoys according to their scores minimises the number of decoys that need to be dealt with in any further ranking steps, whilst ensuring that the set still contains and is enriched with the highest-quality decoys that were generated.

run so as to generate the same number of decoys for each target as Sphinx (around 19,000 on average). FREAD, on average, returned 16 decoys per target. For this set of targets, the average running time for Sphinx was 2.75 hours (when using one core of a Linux server with 128 GB RAM, and a 2.1 GHz processor). MECHANO's running time was similar to that of Sphinx, while FREAD was much faster, taking under one minute in the majority of cases. Though Sphinx is far more time-consuming than knowledge-based methods, it compares favourably to other *ab initio* methods — RosettaAntibody, for instance, can take days to generate its decoy ensemble (Messih *et al.*, 2014).

When considering all decoys generated by each algorithm, Sphinx is able to generate more accurate conformations than either FREAD or MECHANO — the average RMSD of the best decoy produced by Sphinx across all loop lengths is 1.31 Å, whilst FREAD and MECHANO achieve 1.76 Å and 1.74 Å respectively. The standard deviations of

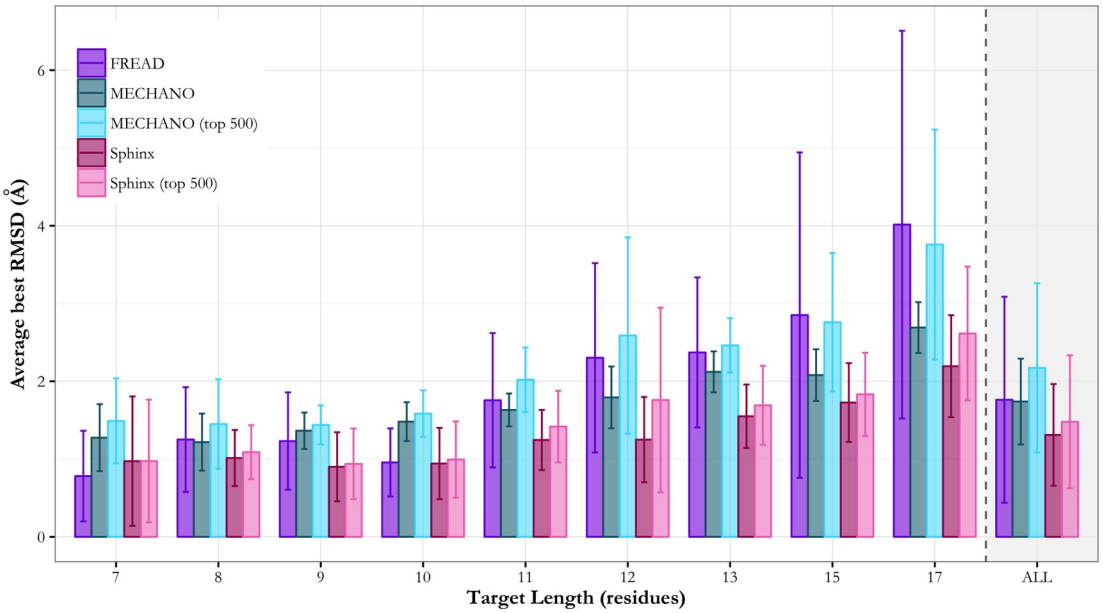

Figure 4.9: Performance of FREAD, MECHANO and Sphinx on the 90-loop test set. Results shown are the average RMSDs of the best decoy produced for each target, split by loop length. 'Sphinx' and 'MECHANO' are the results considering all decoys generated by each method, while 'Sphinx (top 500)' and 'MECHANO (top 500)' consider only the reduced set of decoys selected using our H3-specific, knowledge-based energy function. Sphinx is able to produce more accurate conformations than either FREAD or MECHANO, even when considering only the top 500 decoys.

the results from Sphinx, FREAD and MECHANO were 0.65 Å, 1.32 Å, and 0.55 Å respectively. In addition to the low RMSDs, Sphinx generated a prediction in every case, while FREAD failed to make a prediction for 19 of the 90 targets (79% coverage).

When the H3-specific energy function is introduced and the reduced set of 500 decoys selected, the best decoy is still more accurate on average than the one produced by either FREAD or MECHANO (giving an average best RMSD of 1.48 Å). This is only 0.17 Å worse than the result using all decoys, even though the number of decoys has been reduced substantially from 19,000 (on average) to 500.

In addition to looking at the best decoy produced for each target, we examined the overall quality of the decoy sets produced by Sphinx and MECHANO by looking at how the RMSDs of all decoys generated are distributed (Fig. 4.10). From this figure, it can be seen that the distribution of the Sphinx decoy set is shifted to the left in comparison to that of MECHANO — Sphinx therefore produces far more low-RMSD decoys than the *ab initio* algorithm. Of the decoys generated by MECHANO, 0.4% have an RMSD

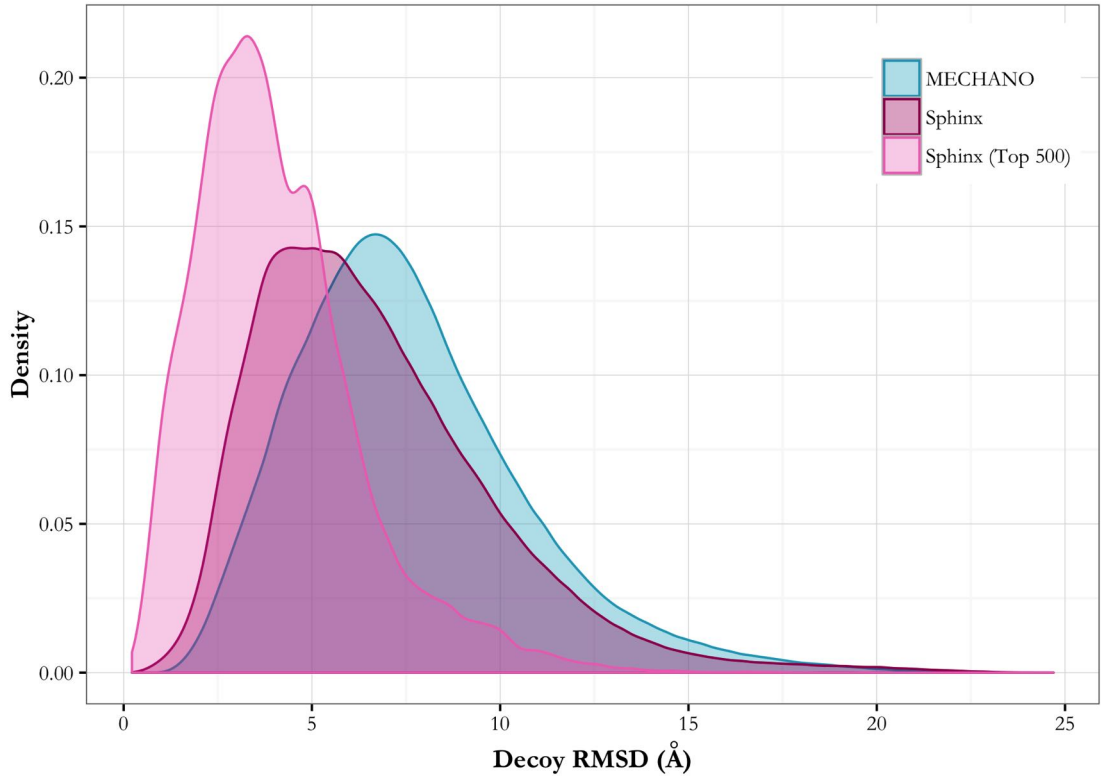

Figure 4.10: RMSD distributions for all decoys produced for all targets by MECHANO, Sphinx, and the top 500 Sphinx decoys. Sphinx produces more low-RMSD conformations than MECHANO, and the distribution is improved further by implementing the ranking step.

of below 2 Å — for Sphinx, this value is 1.5%. The distribution is improved still further when the ranking step is included — the smaller decoy set contains proportionally more low-RMSD decoys and fewer high-RMSD decoys than the full set. The decoy ensemble produced by Sphinx is therefore enriched with near-native structures — the proportion of decoys with an RMSD below 2 Å increases markedly to 14.3%.

As an example, the best decoy generated by Sphinx for the target 3GI9 is shown in Fig. 4.11. This H3 loop is 15 residues in length. FREAD fails to make any near-native decoys for this target, achieving a best RMSD of 5.33 Å. Sphinx is able to produce a useful decoy, with an RMSD of 1.64 Å, using a fragment of 14 residues. This fragment has a similar overall shape to the target structure (Fig.4.11A), but is one residue too short and is oriented incorrectly when its anchor residues are aligned to the target's (Fig. 4.11B). Sphinx, by contrast, is able to use the structural information contained within this fragment to generate a decoy that closely matches the target structure

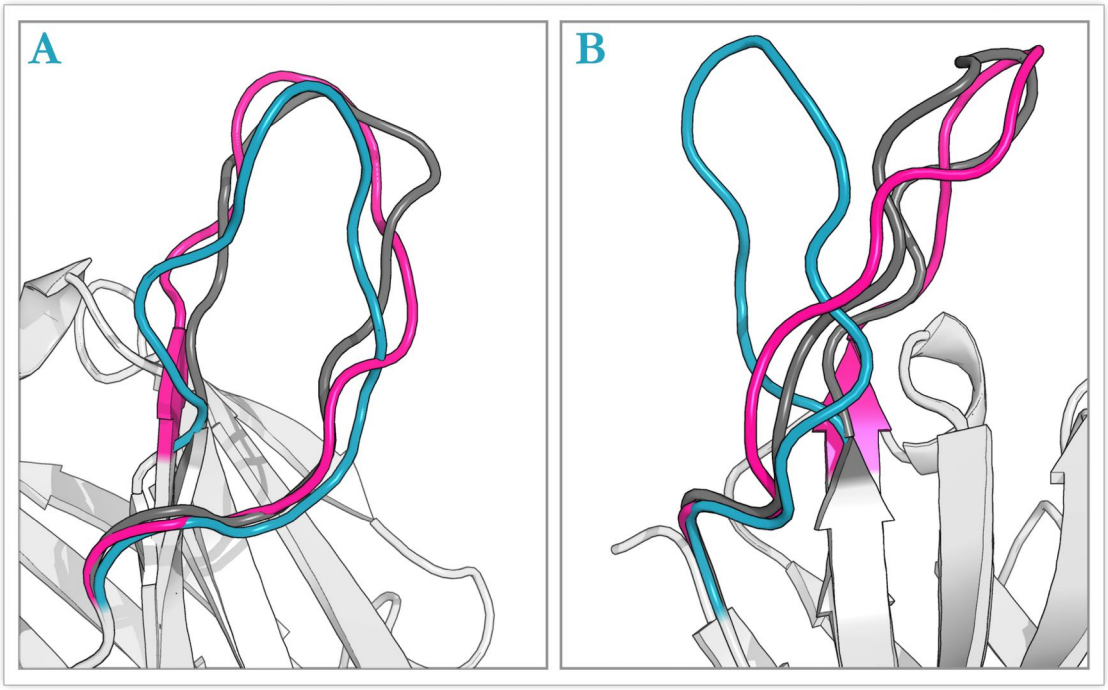

Figure 4.11: An example decoy generated by Sphinx. Parts A and B of this figure show the same structures from different points of view. The native structure of the target (the H3 loop of antibody structure 3GI9) is shown in grey, the best decoy generated by Sphinx is shown in pink (RMSD = 1.64 Å, and the fragment used to produce this decoy is shown in blue. Framework residues are shown in grey. In this case, the fragment used was one residue shorter than the target, and had a similar structure but was oriented incorrectly. Sphinx was able to use the structural information contained within this fragment to generate a near-native decoy.

(with an RMSD of 1.64 Å).

To verify that the knowledge-based scoring function is improved by using H3-specific data, we compared its accuracy to a general protein version (*i.e.* one that uses data from many different types of protein). The average best RMSDs achieved are shown in Figure 4.12, and the RMSD distributions for the decoy sets are displayed in Figure 4.13. The general energy function performs worse than the H3 version, giving an average best RMSD of 1.73 Å for the top 500 decoys, compared to 1.48 Å achieved by the H3-specific version. The H3 version also produces decoy sets with better RMSD distributions, *i.e.* with a greater number of near-native structures. The general version does improve the distribution compared to the set of all decoys generated, but not by as much.

We have compared the best decoys generated by Sphinx to the H3 loop in the PDB that has the closest structure to the target. For each target in the test set, all length-matched H3 loops from the PDB were extracted using SAbDab (Dunbar *et al.*, 2014).

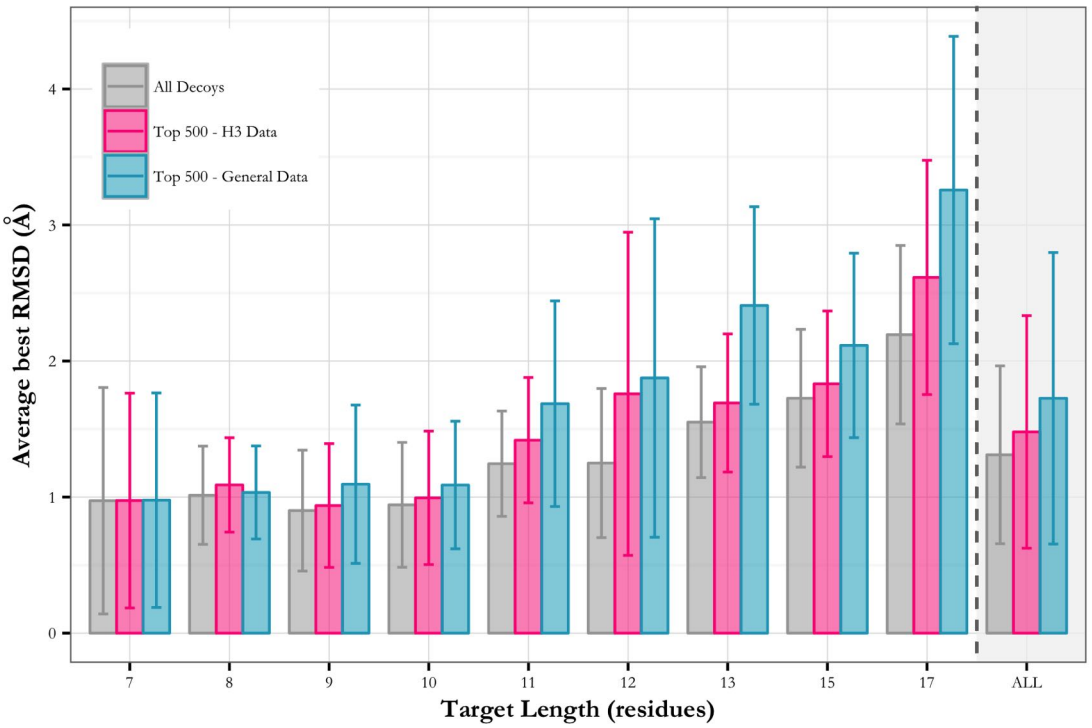

Figure 4.12: Results achieved using an H3-specific scoring function, compared to one that uses data from non-specific protein types ("general" function). The function ranks our decoys more accurately when using H3 data.

The anchor residues of each were aligned to those of the target, and the RMSD to the target structure calculated. On average, the best structure had an RMSD of 1.55 Å — Sphinx gave an average best RMSD of 1.31 Å. The lowest-RMSD decoy generated by Sphinx is better in two thirds of cases than the closest H3 structure available. It is therefore able to produce better conformations than anything available in the PDB, which means that, even given an ideal scoring function, knowledge-based algorithms would be unable to achieve the accuracies of Sphinx.

Switching

By allowing loop building to 'switch' from one anchor to to the other after a failure, the preferred building direction is allowed to dominate. For each target in the test set, we have examined this preference by looking at the direction in which the decoys in the set were generated. The outcome is shown in Figure 4.14. For most targets, the majority of decoys are built starting from the C-anchor — this is true for all but 9 of the 90 loops.

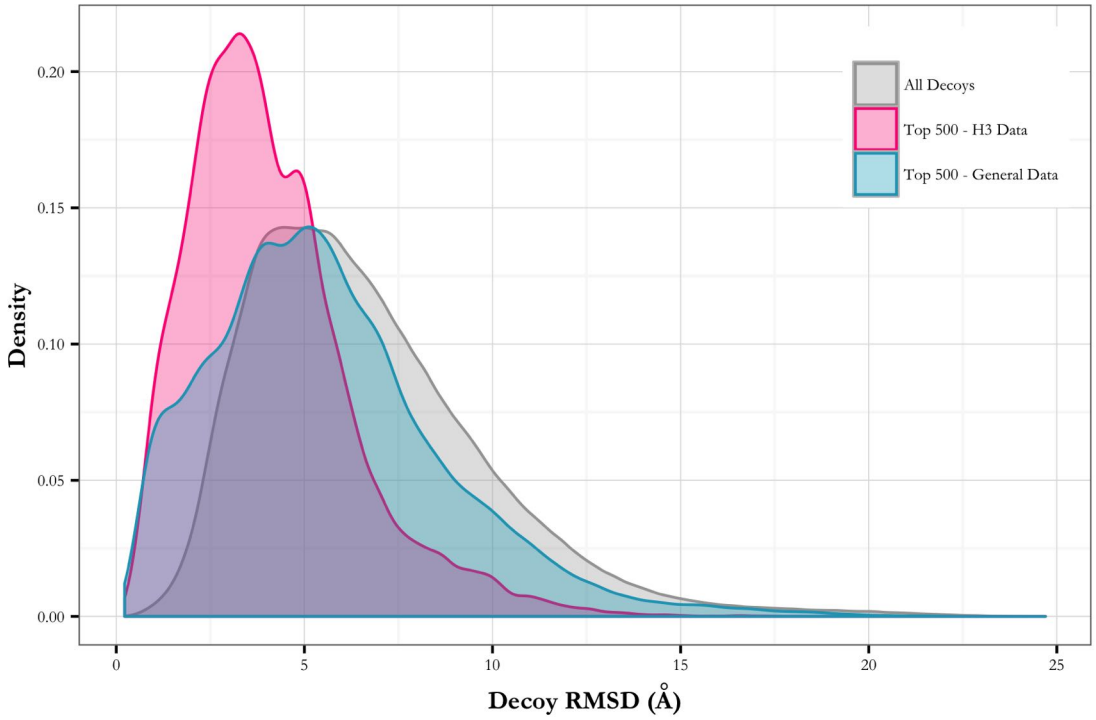

Figure 4.13: Comparison of RMSD distributions for the reduced decoy sets produced using an H3-specific and general scoring function. The reduced decoy sets contain more near-native decoys when the scoring function uses H3 data.

On average, 60% of the decoy set was built in the C to N direction. The preference becomes more obvious when looking at the reduced decoy sets; on average, 80% of decoys that are ranked in the top 500 are built from the C-anchor, and a preference for the N direction is only displayed by two targets. This implies that the lowest-RMSD decoys in the set are built from the C-anchor. The preference of the best ten decoys (by RMSD) is more variable, but overall 62% are built in the C-N direction.

We have attempted to explain this result by looking at the conformation (kinked or extended) of the C-terminal end of the H3 loops. We defined the targets as kinked or extended based on the work of Weitzner *et al.* (2015). The conformation can be described using two angles (Figure 4.15): τ_{101} (the pseudo-bond angle between the Cα atoms of residues 100x, 101 and 102, where residue 100x is the one immediately preceding residue 101 and when using Chothia numbering) and α_{101} (the pseudo-dihedral between the Cα atoms of residues 100x, 101, 102 and 103). A given loop is designated as kinked if τ_{101} is in the region 84.2-117.8° and α_{101} is within the range 3.6-74.4°. Of the 90 targets, 9

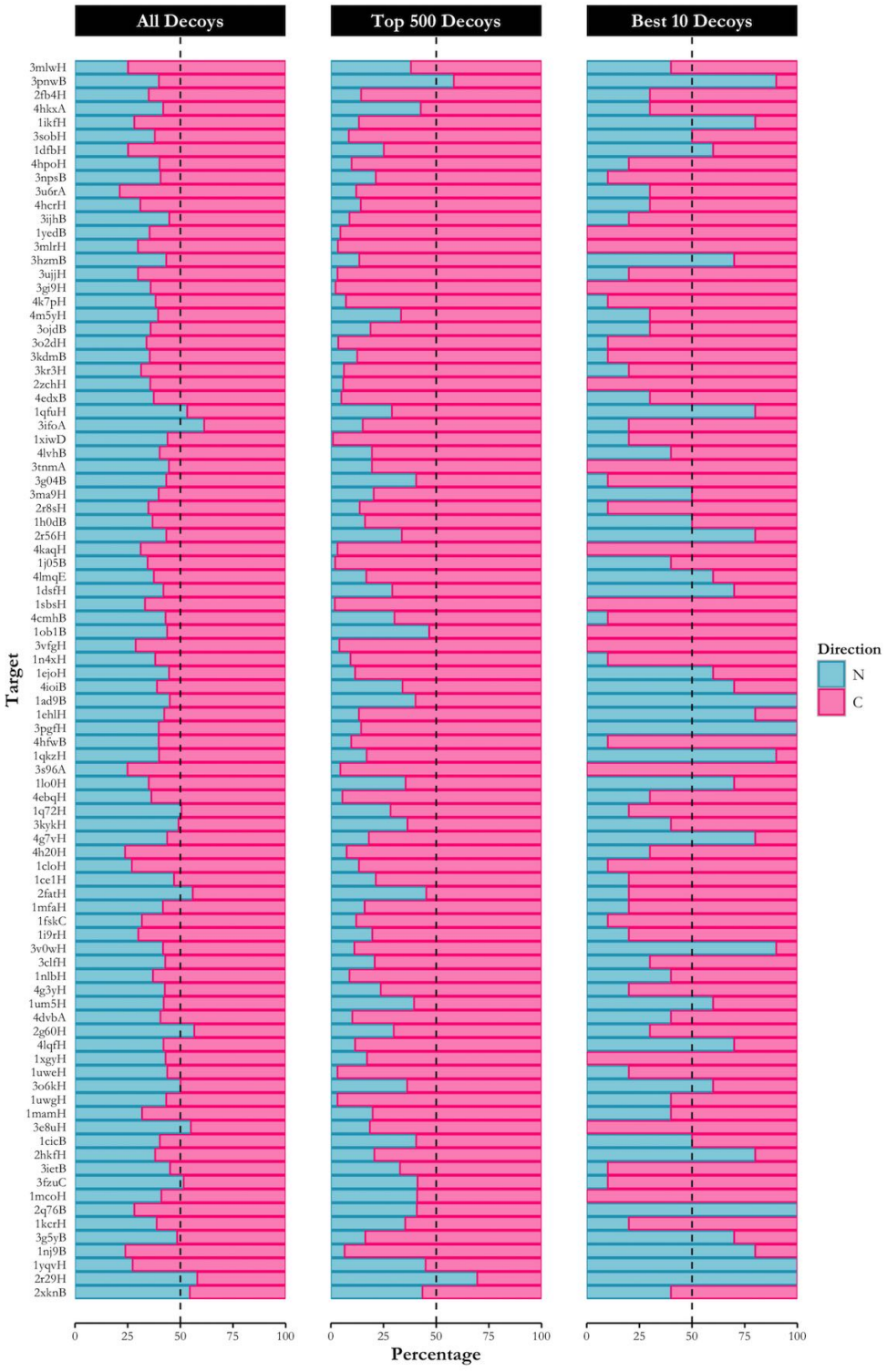

Figure 4.14: By switching the direction of loop building after a failure, the preferred direction is allowed to dominate. These figures show the proportion of decoys built in the N (blue) and C (pink) directions for the whole decoy set (left), the top 500 decoys as ranked by the H3-specific knowledge-based energy function (middle), and the best ten decoys by RMSD (right). In the majority of cases, the C direction is preferred.

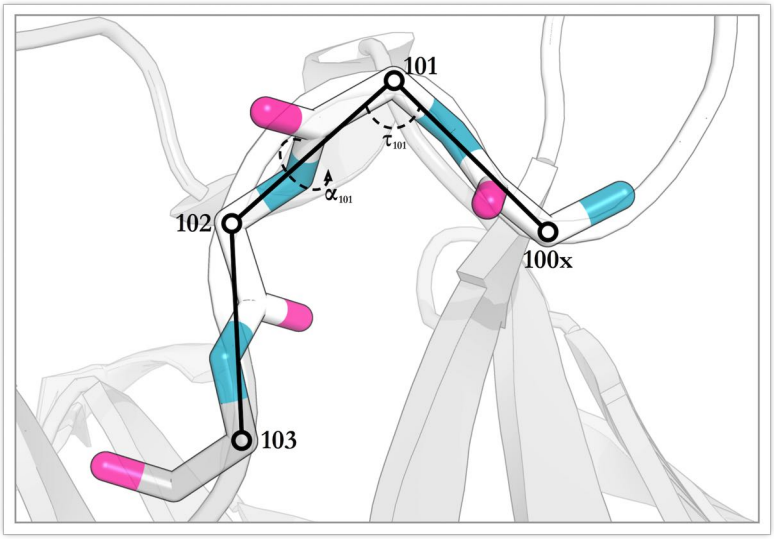

Figure 4.15: Angles used to describe the C-terminal kink present in most antibody H3 loops, as defined by Weitzner *et al.* (2015).

are of the extended conformation — all of these had a preference for the C-N direction (when considering the reduced decoy sets), with an average of 72% of decoys being built this way. All of the targets that favoured the N-C direction, then, were kinked — however on average, the 81% of decoys for kinked targets were built from the C-anchor. Examining the structures by eye has also yielded no clues; more thought and research is required to conclude why there is a preference in building direction.

4.6.1 RosettaAntibody Benchmark

To compare our results to other software, we used Sphinx to generate decoy sets for the benchmark targets used to test the H3 modelling algorithm within RosettaAntibody, one of the leading methods, and one whose performance was assessed during AMA-II (Sivasubramanian *et al.*, 2009; Weitzner *et al.*, 2014). This set of targets includes 53 H3 loops, divided into five categories depending on loop length: very short (4-6 residues), short (7-9), medium (10-11), long (12-14) and very long (17-22). There are 3, 22, 14, 10 and 4 loops in each grouping respectively. We also obtained predictions using RosettaAntibody for comparison, generating 500 decoys for each target; the number used by Stein and Kortemme (2013) and the same number as in the Sphinx reduced set. Fig. 4.16 shows how the average RMSD of the best decoy produced by

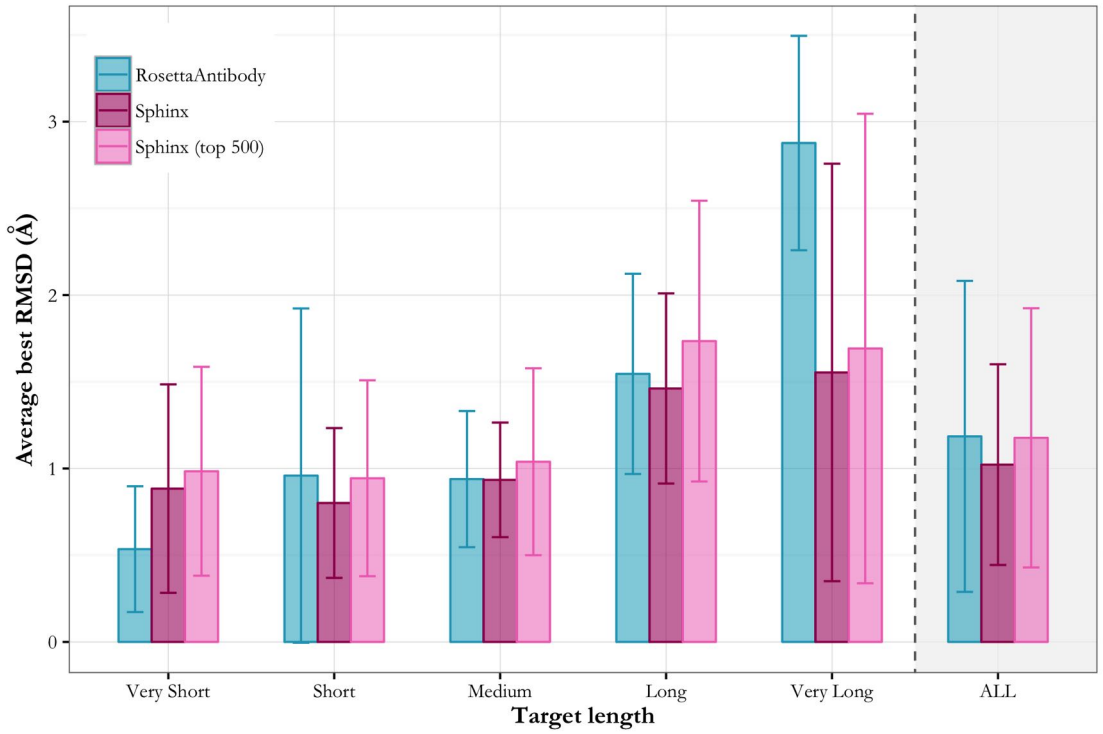

Figure 4.16: Comparison of Sphinx to RosettaAntibody. The best decoy generated by Sphinx has a lower RMSD on average than the best produced by RosettaAntibody. When the top 500 decoys are considered, on average the best decoys produced have very similar RMSDs (1.18 Å and 1.19 Å for Sphinx and RosettaAntibody, respectively). Sphinx offers noticeably better decoys for loops in the 'very long' category.

Sphinx compares to the average RMSD of the best decoy from RosettaAntibody. On average, the best decoy produced by Sphinx has an RMSD of 1.04 Å — this compares favourably to the performance of RosettaAntibody, which produced an average best RMSD of 1.19 Å. When the H3-specific energy function is implemented and the top 500 decoys considered, on average the best decoy produced by Sphinx is still very similar to the best produced by RosettaAntibody (achieving an average best RMSD of 1.18 Å). In particular, Sphinx is able to generate better decoys than RosettaAntibody for the targets within the 'very long' category.

The RMSD distributions obtained by the two different algorithms are shown in Figure 4.17. The decoy set from Sphinx has a greater proportion of near-native decoys compared to RosettaAntibody, making it more likely that a subsequent ranking method will be successful in choosing a good prediction. This is demonstrated by the results shown in Figure 4.18 — the probability of choosing a decoy from the set at random,

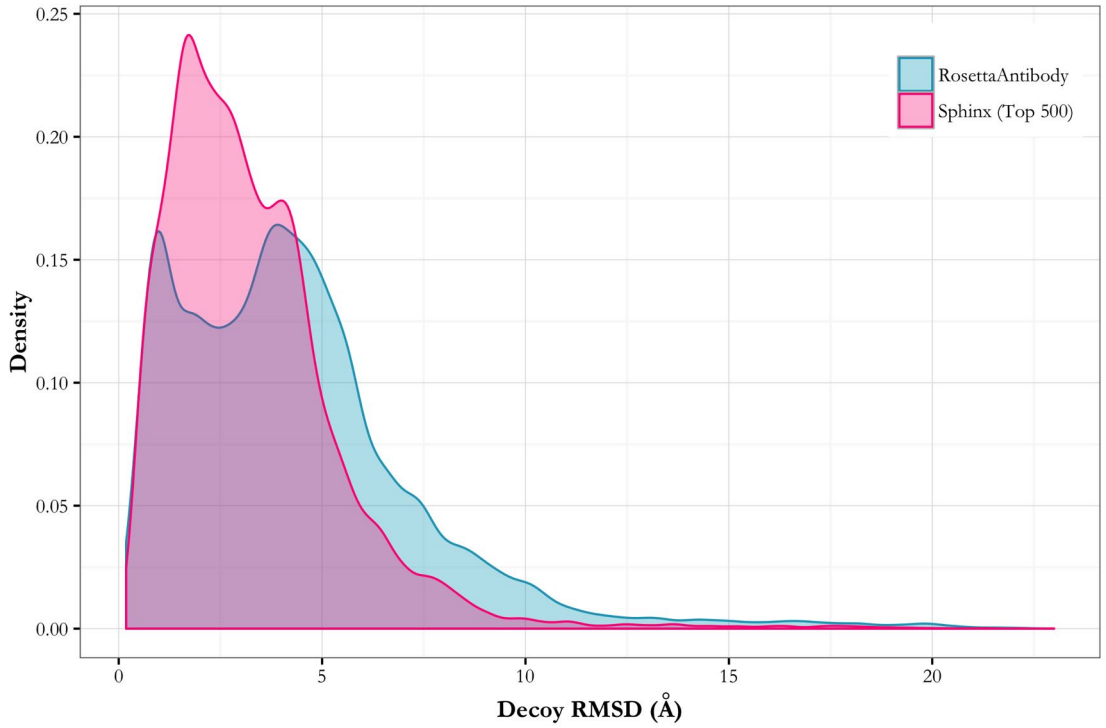

Figure 4.17: RMSD distributions for Sphinx (reduced set) and RosettaAntibody, for the targets in the RosettaAntibody benchmark set used by Sivasubramanian *et al.* (2009). More low-RMSD conformations are made using Sphinx.

with an RMSD below a cutoff value, is almost always higher using Sphinx. In addition, Sphinx is much faster — RosettaAntibody can take up to an hour to generate a single decoy structure (and therefore approximately 500 hours to generate all required decoys), while Sphinx takes approximately two hours (on average) to produce its entire decoy set.

The H3-specific, knowledge-based algorithm H3Loopred (Messih *et al.*, 2014) has also been compared to RosettaAntibody, building loops onto model structures generated by PIGS (Marcatili *et al.*, 2008). In the model environment, anchor residues are in a non-native conformation, making prediction more difficult. We generated model structures of the 53 antibody structures in the RosettaAntibody benchmark set using ABodyBuilder (Leem *et al.*, 2016), and used Sphinx to generate decoy sets for the H3 loops (results shown in Fig. 4.19). For these runs, we implemented a date restriction to ensure a fair comparison to the results of H3Loopred, whose results are dependent on the availability of antibody structures (any fragments from antibody structures deposited in the PDB after 15th October 2012, the date quoted by Messih *et al.*, were

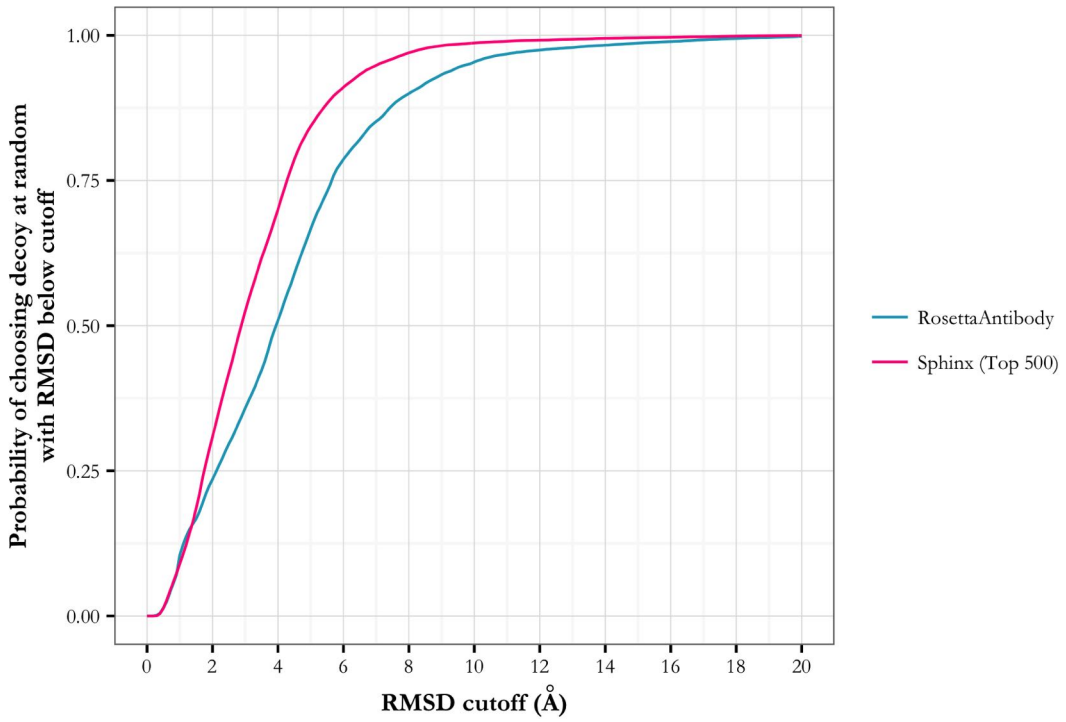

Figure 4.18: The probability of choosing a decoy at random from the decoy set with an RMSD below a cutoff value, for the decoys produced by Sphinx and Rosetta. The probability is normally higher when using Sphinx.

ignored). In this case, we compare Sphinx's results to those reported by the authors of the two methods (Sivasubramanian *et al.*, 2009; Messih *et al.*, 2014).

Sphinx produces decoys that are closer to the native structure than RosettaAntibody, achieving an average best RMSD of 1.21 Å when considering all decoys and 1.36 Å for the top 500, compared to 2.09 Å for RosettaAntibody. Sphinx seems to be minimally affected by the non-native environment, with a reduction in RMSD of only 0.19 Å. Sphinx also compares favourably to H3Loopred, for which the best decoy generated had an average RMSD of 1.47 Å. It should be noted, however, that the antibody models onto which the loop decoys were built were generated using different software — the conformations of the anchor residues will therefore be slightly different for each method's set of results.

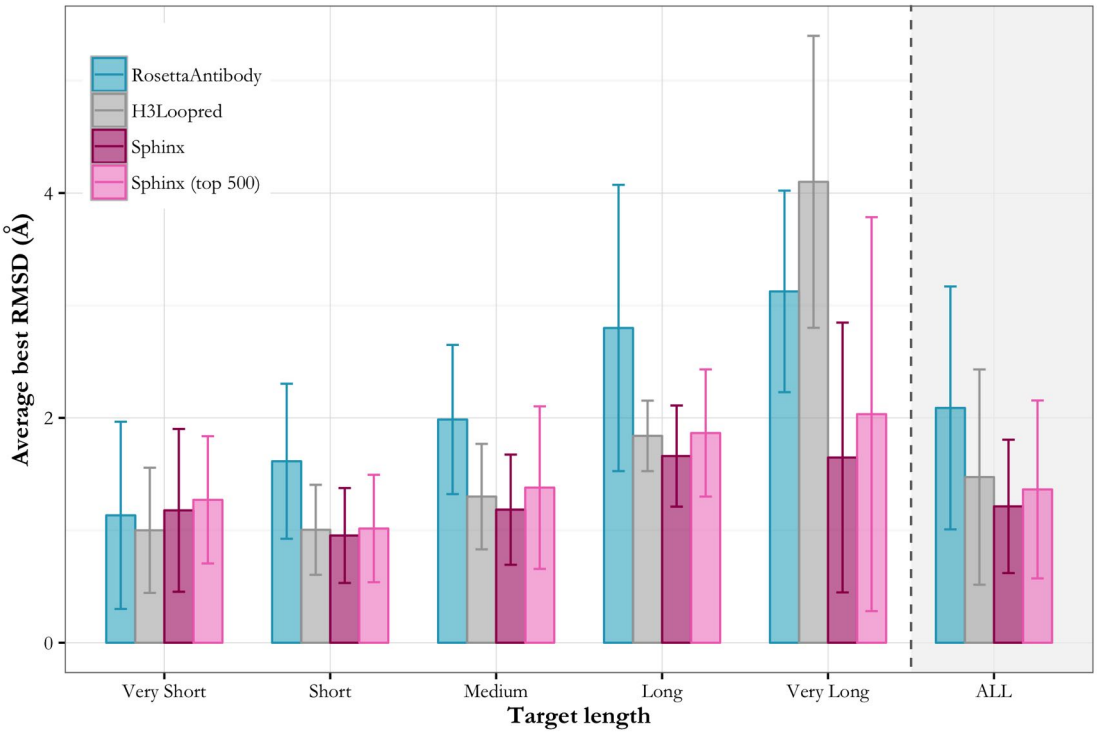

Figure 4.19: Comparison of Sphinx to RosettaAntibody and H3Loopred on model structures (*i.e.* a non-native environment). Sphinx is able to produce more near-native decoys than RosettaAntibody, and similar quality decoys to H3Loopred. Sphinx is particularly good at predicting the structures of long loops.

4.7 General Loop Version of Sphinx

Since Sphinx can achieve excellent results when generating decoys for the difficult problem of H3 prediction, we decided to develop a more general algorithm that can be applied to non-specific protein types. The following changes have been made:

- fragment database — we created a new database that contains structures of loops from the PDB. The loop structures of all PDB entries with non-identical sequences (according to PISCES (Wang and Dunbrack, 2003), using the structure with the best resolution for each sequence), plus three residues on each side, were extracted and split into all possible fragments of 3 to 30 residues in length. Loop regions were located using DSSP (Joosten *et al.*, 2011), and were defined as regions connecting secondary structures of at least three residues in length. This led to a database containing the loop structures of 65,108 protein chains (created October 2015).

- Ramachandran distributions — we changed the ϕ/ψ angle distributions used in loop building from the H3-specific ones used previously, to a set of general distributions. These distributions were obtained from the loop structures that were selected for the database.

- scoring function — we no longer use the H3-specific function, but the general one described in Section 4.5.4.

4.7.1 Reparameterising Sphinx

It is possible that the parameters used in Sphinx-H3 (*e.g.* the number of decoys to make per fragment) are not appropriate when generating decoys for different protein loop types. We therefore carried out the same tests that were run for Sphinx-H3, using the new general version. For this purpose, we created a new target set — from the general protein loops that were extracted to make the fragment database described above, a series of loops were selected with no more than 40% sequence identity, resolution below 2 Å, and no B-factor above 30. From these loops a dataset of 70 targets was selected, with 10 examples at each even loop length from 6 to 18 residues. Fragments from proteins with an identical sequence to the target are ignored.

Decoys Per Fragment

We generated 150 decoys from each fragment that passed the anchor geometry filter and has an alignment score of over 25 (these filters have not changed from the H3 version — see Section 4.2 for details). The results are similar to those obtained for Sphinx-H3; making more than 100 decoys does not improve results enough to make the generation of more decoys beneficial.

Number of Fragments and Minimum Fragment Length

Figure 4.21 shows how the RMSD of the best decoy generated by Sphinx varies as the number of fragments selected (according to their ESS scores) and the minimum fragment length considered are varied. One hundred decoys were randomly selected for each fragment from those produced during the previous investigation into the

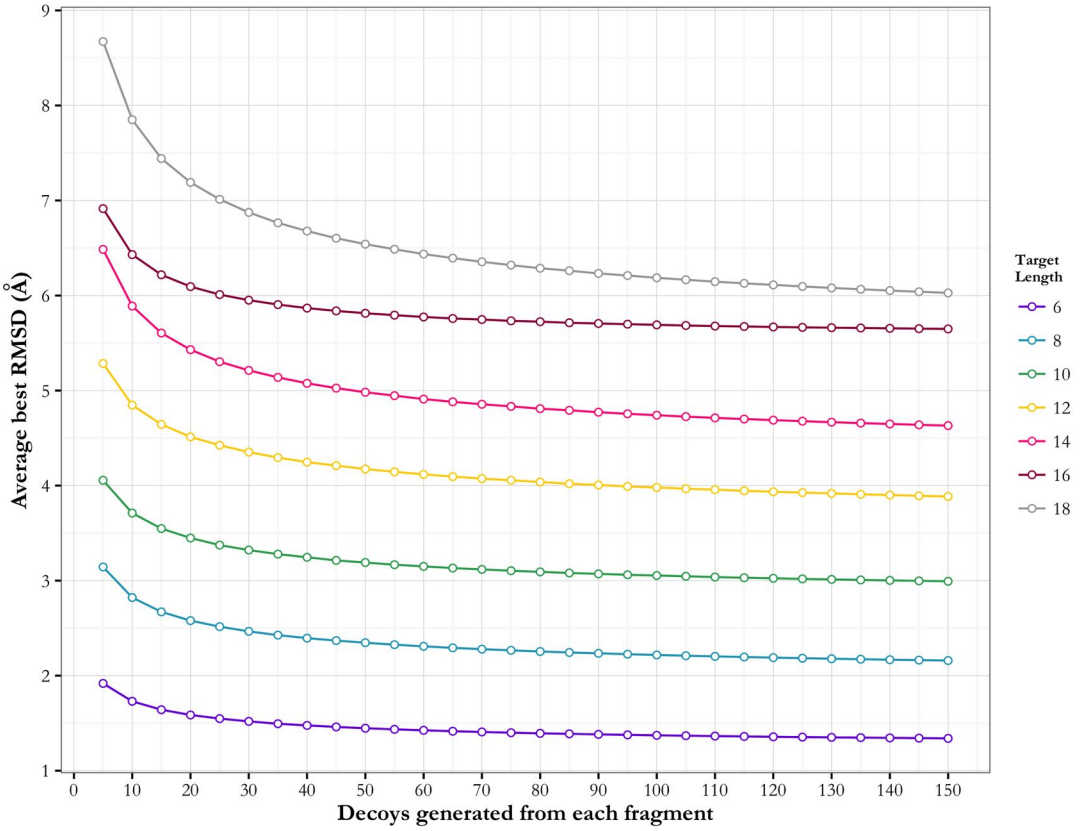

Figure 4.20: Dependence of the average best RMSD on the number of decoys generated from each fragment. Result shown is the RMSD of the best decoy produced by a fragment, averaged over all fragments (those that pass the anchor geometry and alignment score filters), for the general loop set. As for the H3-specific version, minimal improvement occurs after 100 decoys are made.

optimum number per fragment. Again, the number of fragments required is length dependent (*i.e.* more are required for longer loops). However, fewer fragments can be used in comparison to the H3 version. Considering the results shown in Figure 4.21, we decided on the following parameters:

$$\text{minimum target length} = \begin{cases} n-3 & \text{if } n \leqslant 12 \\ n-4 & \text{if } n > 12 \end{cases}$$

$$\text{number of fragments} = \begin{cases} 12.5(n-2) & \text{if } n \text{ is even} \\ 12.5(n-1) & \text{if } n \text{ is odd} \end{cases}$$

The number of fragments is therefore half of that required for Sphinx-H3.

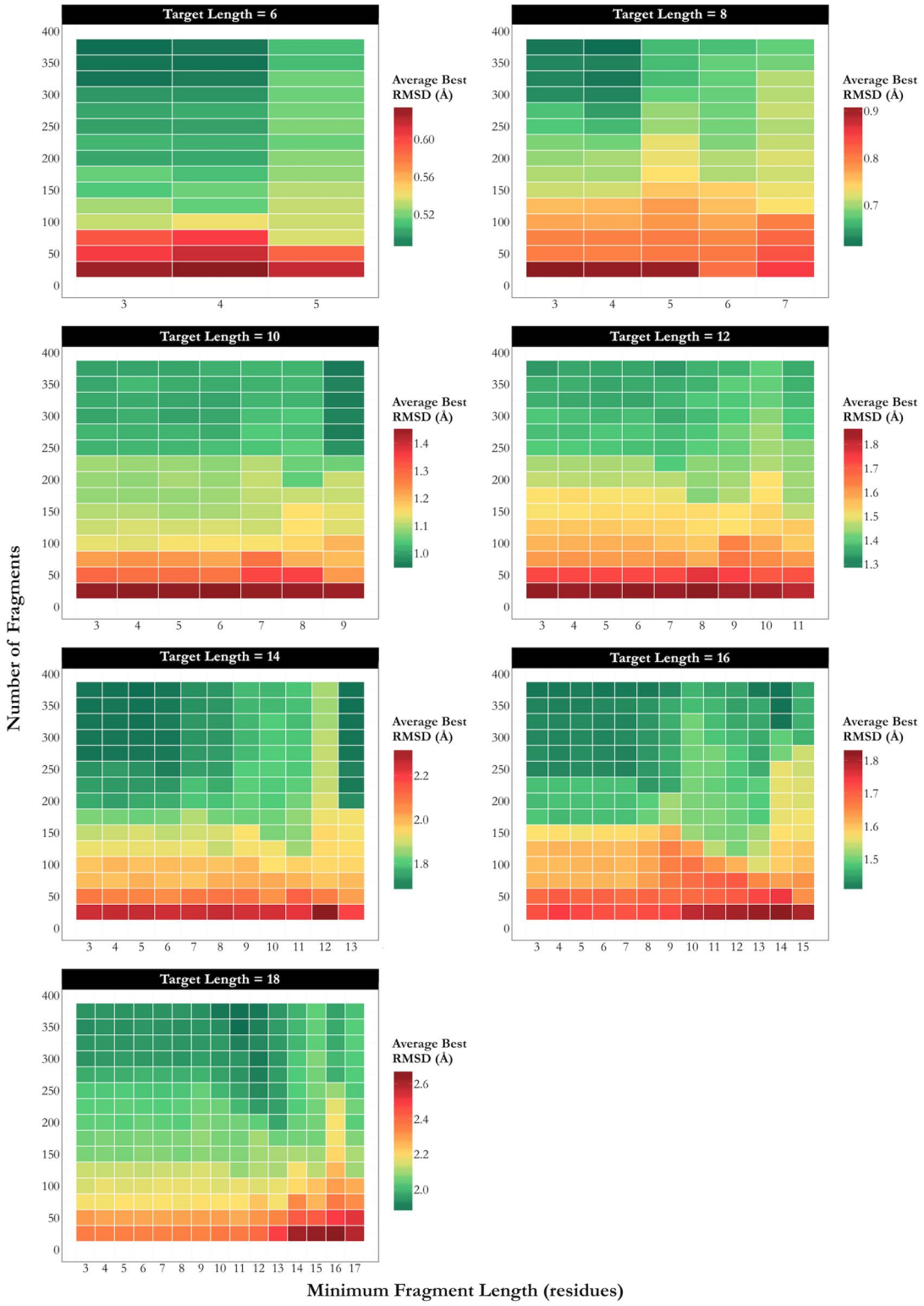

Figure 4.21: Dependence of the average best RMSD on the number of fragments used and the minimum fragment length, for each target length present in the general loop set.

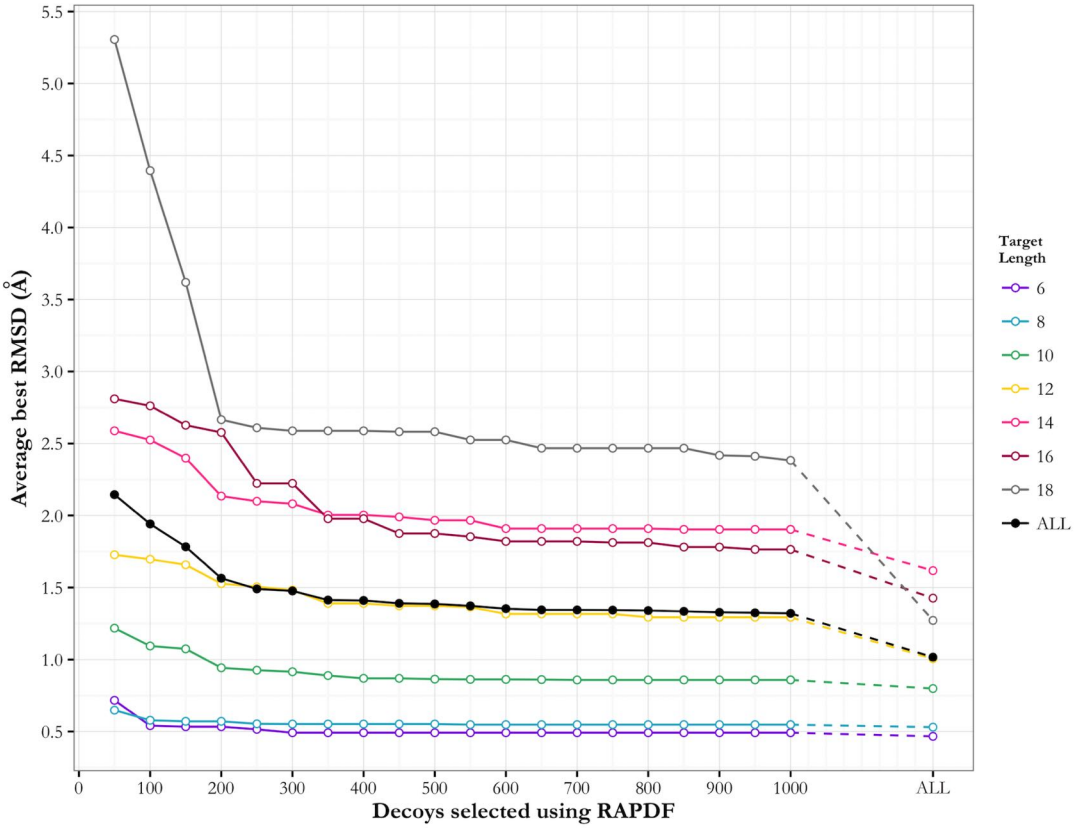

Figure 4.22: Decoy selection for general loops using the knowledge-based energy function. As in the case of H3 loops, choosing the top 500 decoys according to their scores minimises the number of decoys that need to be dealt with in any further ranking steps, whilst ensuring that the set still contains and is enriched with the highest-quality decoys that were generated.

General Scoring Function

We then investigated how many decoys should be selected to make up the reduced set, when using the general knowledge-based energy function. As before, we monitored the RMSD of the best decoy in the set as the number of decoys selected is increased (Figure 4.22). The optimum number of decoys for all target lengths is again 500, since the average RMSD of the best decoy remaining in the set does not improve significantly by making the selection larger.

4.7.2 Results

Figure 4.23 shows the ability of Sphinx to generate near-native conformations for general loops. The other methods shown are the algorithms on which Sphinx is based: the knowledge-based method FREAD (Deane and Blundell, 2001; Choi and Deane, 2010)

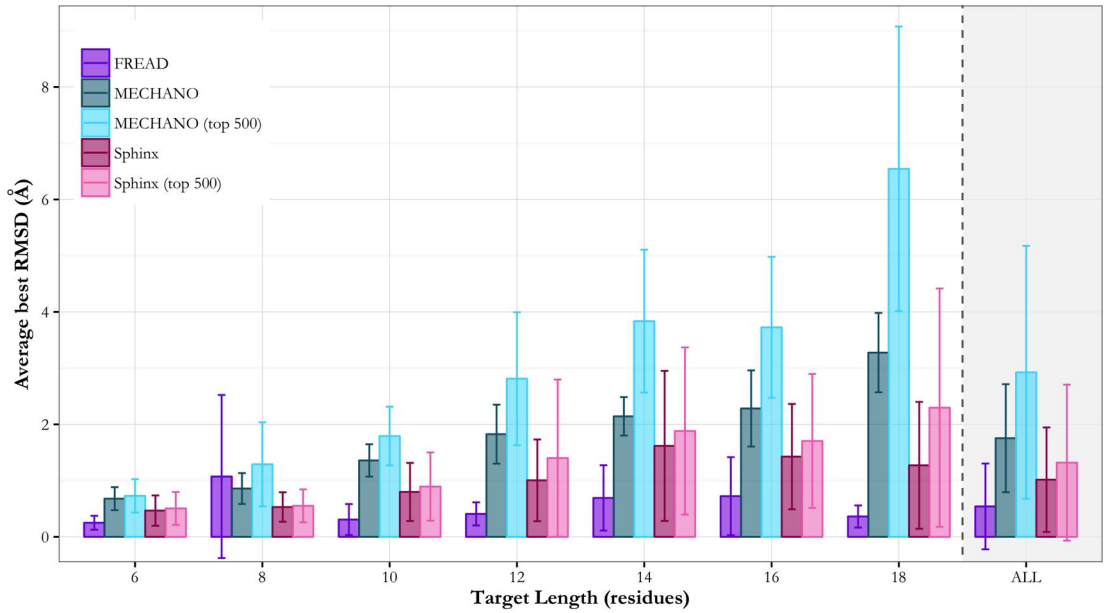

Figure 4.23: Performance of FREAD, MECHANO and Sphinx on the general loop target set. Results shown are the average RMSDs of the best decoy produced for each target, split by loop length. 'Sphinx' and 'MECHANO' are the results considering all decoys generated by each method, while 'Sphinx (top 500)' and 'MECHANO (top 500)' consider only the decoys selected using our knowledge-based energy function.

and our own *ab initio* method, MECHANO. We ran FREAD using the same fragment database as used by Sphinx (see Section 4.7), and on average it returned 7 decoys per target. To ensure a fair comparison, MECHANO was run so as to generate the same number of decoys for each target as Sphinx (around 11,000 on average), and its reduced sets of 500 decoys were obtained in the same way.

Figure 4.23 shows that FREAD is more accurate than Sphinx on average (with an average best RMSD of 0.54 Å compared to 1.02 Å respectively), but there were 26 targets for which FREAD failed to find suitable fragments in the database (63 % coverage). Sphinx, on the other hand, returned a prediction in every case. For the 44 targets for which FREAD produced a prediction, Sphinx was able to generate a better or equally accurate decoy in 19 cases. For 47 of the 70 targets, Sphinx produced a decoy with an RMSD of below 1 Å; 8 more than FREAD.

Sphinx, by including information from fragments, is able to considerably outperform its base *ab initio* algorithm, MECHANO. The average best RMSD achieved by Sphinx is 1.02 Å; for MECHANO this value is 1.75 Å. When considering the top 500 decoys, the

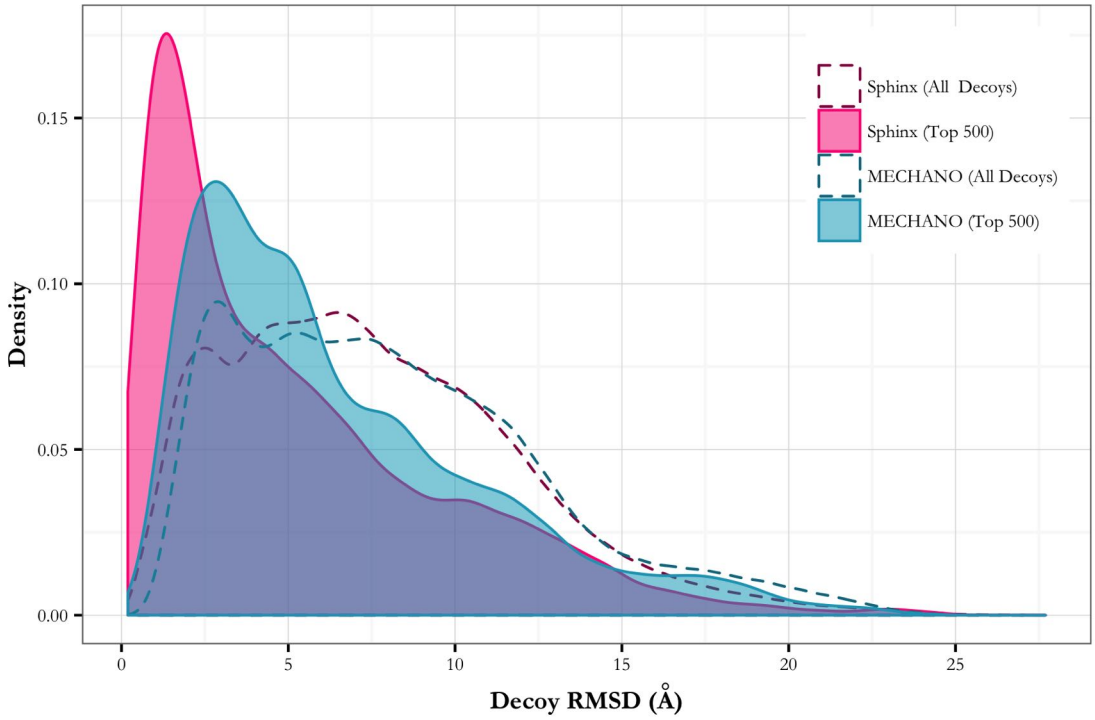

Figure 4.24: RMSD distributions for the decoy sets (both the reduced set and the set of all decoys generated) produced by Sphinx and its base *ab initio* method, MECHANO. Decoy sets produced by Sphinx contain proportionally more near-native decoys. Data shown is for all targets in the general loop dataset.

difference between the two methods is increased — the average RMSD of the best decoy produced was 1.32 Å, compared to 2.93 Å for MECHANO, and the lowest-RMSD decoy generated by Sphinx was better than that of MECHANO in 58 of the 70 cases. The reduced decoy sets of MECHANO only contained a conformation with an RMSD below 1 Å for 15 targets; Sphinx produced sub-Ångström decoys for 45.

The improved ability of Sphinx to produce near-native loop conformations is further illustrated by the RMSD distributions of the decoy sets produced (Figure 4.24). There is a greater proportion of low-RMSD decoys in the decoy set generated by Sphinx, indicating that the inclusion of fragment information helps to direct the conformational search towards near-native structures, and the proportion of low-RMSD structures in the set is improved after using the scoring function.

4.8 Conclusions

We have successfully developed a loop modelling algorithm, Sphinx, which combines both knowledge-based and *ab initio* methodologies. Sphinx is capable of generating high-quality decoy sets for the H3 loop of antibodies, a particularly challenging loop. We have shown that our hybrid approach is able to produce more near-native loop conformations than can be achieved using either the knowledge-based or *ab initio* algorithms separately. This is true using both our own software and leading H3-specific *ab initio* (RosettaAntibody) and knowledge-based (H3Loopred) techniques. Sphinx, unlike some knowledge-based methods, is also able to give 100% coverage; *i.e.* a prediction is always made. In addition, we have demonstrated that Sphinx is minimally affected by the presence of anchor residues in their non-native conformations — this is very important, since this will be the case for most real-life applications.

Sphinx, on average, generates decoys that are closer to the native structure than any other H3 loops present in the PDB. This means that, even with a perfect decoy selection method, traditional knowledge-based approaches are unable to achieve RMSDs as low as Sphinx.

Sphinx is the first loop modelling algorithm that completely integrates both knowledge-based and *ab initio* approaches. As this is the case, it is likely that other methods could benefit from the concept, for instance the leading H3-specific algorithms, RosettaAntibody and H3Loopred.

Using H3 fragments that are shorter than the target loop, along with resampled Ramachandran distributions, has enabled us to successfully use H3 loop data, despite the limited amount available. Our H3-specific, knowledge-based energy function has been shown to perform effectively, minimising the number of decoys that must be dealt with in subsequent ranking steps, whilst ensuring near-native decoys remain in the set. The distribution of RMSDs for the reduced Sphinx decoy set is also enhanced compared to the complete set, showing that Sphinx is able to produce a relatively small ensemble of decoys that is enriched with near-native structures.

The Sphinx algorithm only uses fragments that have fewer residues than the target loop. The results achieved by the method indicate that these fragments are a valuable source of structural information that can be used when predicting loop structures, reinforcing the findings of Nowak *et al.* (2016) that loops of different lengths can still have very similar structures. Since this approach works well for the difficult H3 loop, we developed a version for non-specific protein loop types, and showed that it also performs well. Results for general loops are better than those for H3 loops — this is likely because there is far more structural data available than for the specific case of H3, and H3 loops are more variable.

A key area where more research is required is the ranking of decoys. While Sphinx contains an effective H3-specific energy function that produces small, high-quality decoy sets, it can not be used reliably to select a final loop model. This issue is not unique to our algorithm — in the second Antibody Modelling Assessment, imperfect ranking was seen to limit prediction accuracy Teplyakov *et al.* (2014). In the next chapter, we investigate this problem and test various methods of decoy selection.

Pre-Chapter Image: DNA bound to a DNA methyltransferase enzyme via its loop regions.

5 Sphinx: Decoy Selection

Contents

5.1 Introduction

The ability of any structure prediction algorithm to produce accurate results depends on two factors: the conformational search, and the selection of the final prediction. As described in the previous chapter, Sphinx is able to produce a high-quality set of 500 decoys. However, for most practical uses, it is necessary to select only a few of these

(ideally one) as a final prediction. A variety of selection methods have therefore been developed, that attempt to accurately identify which decoy in the set has the closest structure to that of the native protein.

In this chapter, we test multiple approaches to the ranking problem, and select the best to be incorporated into Sphinx as the final part of the algorithm. Initially, we test the different methods on the decoys generated by Sphinx for the general loop target set described in Chapter 4 (a set of 70 loops ranging from 6 to 18 residues in length).

5.2 Clustering

The first approach we investigated was clustering. Many loop prediction algorithms use clustering as part of the decoy selection procedure, including PLOP (Jacobson *et al.*, 2004), Kotai Antibody Builder (Shirai *et al.*, 2014) and the H3 prediction method created by Accelrys (Fasnacht *et al.*, 2014). Clustering works on the assumption that near-native structures will have more structural neighbours in the decoy set than those that are far away from the native — the native structure is typically found at the bottom of a broad energy well, since protein structures are flexible and the native state is actually an ensemble of closely-related structures (Shortle *et al.*, 1998).

Clustering protocols divide the decoy set into smaller groups, with the decoys belonging to each group possessing similar structures. The representative structures from each cluster are reported and ranked according to the size of their clusters (since structures with the most neighbours, *i.e.* in a larger cluster, are assumed to be closer to the native). Several algorithms have been developed, such as Calibur (Li and Ng, 2010), SPICKER (Zhang and Skolnick, 2004b), Pleiades (Harder *et al.*, 2012), ONION (Li *et al.*, 2012), Durandal (Berenger *et al.*, 2012) and Scud (Li and Zhou, 2005). Usually, for protein structures, the distance used to cluster decoys is given by the RMSD between them.

5.2.1 Unweighted Pair Group Method with Arithmetic Mean (UPGMA)

UPGMA is a hierarchical clustering approach. These methods work by creating a dendrogram (or tree), which shows the relationship between the different decoys. Algorithms of this type can either be agglomerative or divisive. Agglomerative algorithms begin with each element — in this case each decoy — occupying its own cluster; pairs of clusters are then merged together to form larger ones. Divisive algorithms, on the other hand, start with one large cluster and break it up into smaller ones.

UPGMA (Unweighted Pair Group Method with Arithmetic Mean) is a simple agglomerative hierarchical clustering method (Sokal and Michener, 1958). The algorithm starts with a distance matrix of size $n \times n$ (where n is the number of decoys), where each row and column represents one decoy, and the value in cell (i, j) is the backbone RMSD between decoys i and j. In our case, the value of n is 500.

The first step is to locate the smallest value in the distance matrix; this corresponds to the two decoys that are the most similar in structure. These decoys are combined into a cluster, and a new distance matrix is formed of size $(n-1) \times (n-1)$ (since two decoys have merged into one cluster). The distances in the row and column relating to the cluster must be re-evaluated; the new values are the average distances between the members of the new cluster and the other decoys. This process continues, with a pair of decoys/clusters being merged each time, until the minimum distance left in the matrix becomes higher than a threshold value. At this point, the clusters can be ranked according to the number of members. We defined the representative decoy within a cluster to be the one with the smallest average distance to all other members (the medoid).

The results of using the UPGMA method to rank the decoys generated by Sphinx for the general loop set are shown in Figure 5.1, and an example set of clusters are shown in Figure 5.2, for the six-residue target 1trbA_289_294. We tested four different thresholds — 1 Å, 1.5 Å, 2 Å and 3 Å. When considering only the top-ranked structure (*i.e.* the representative decoy of the largest cluster), there is minimal difference in ranking ability when comparing the different threshold values. A threshold of 1.5 Å is marginally better than the others, with an average prediction RMSD of 3.62 Å across all loop lengths,

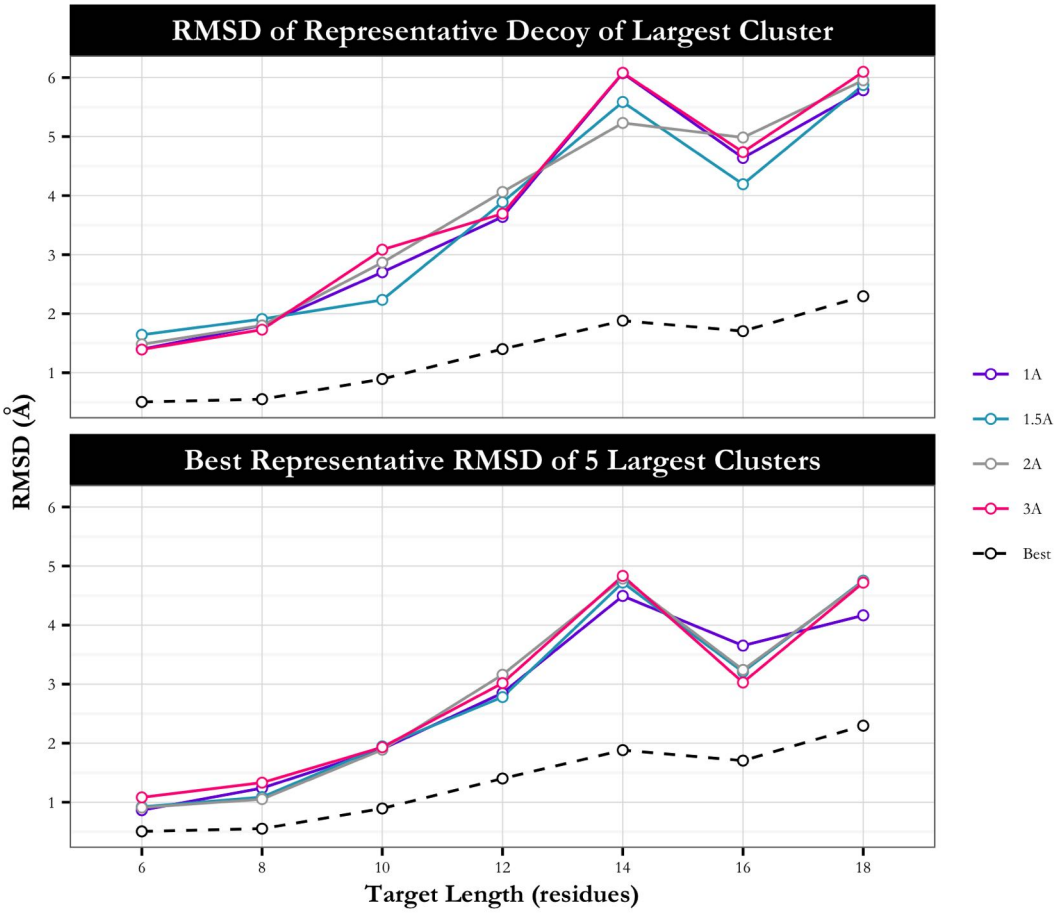

Figure 5.1: Result of using UPGMA to cluster the decoy sets generated for the 70-loop general loop target set, considering the top prediction (the representative decoy of the largest cluster, top) and the best representative of the top five clusters (bottom), using different RMSD cutoff values (1 Å, 1.5 Å, 2 Å, and 3 Å). The prediction accuracy achieved using UPGMA is considerably worse than the best possible (shown as a black dashed line).

compared to 3.72 Å, 3.77 Å and 3.83 Å for thresholds of 1 Å, 2 Å and 3 Å respectively. As the average RMSD of the best decoy for each target, and therefore the best possible result, is 1.32 Å, there is clearly room for improvement here — the top-ranked decoy can have an RMSD to the native structure as much as 4 Å higher than the actual best. Overall there is a trend of decreasing prediction accuracy as the loop length increases — this is partly due to the increase in the best possible result, but the gap between the best possible and the predicted decoy also increases with loop length, indicating that the ranking method itself is affected by loop length. For a threshold of 1.5 Å, the best decoy was found in the top-ranked cluster in 17 of the 70 cases; however this decoy was not the representative of the cluster, and hence not the top-ranked decoy.

The results for each threshold are also very similar when considering the best representative decoy of the largest five clusters (on average 2.74 Å, 2.77 Å, 2.82 Å and 2.85 Å for thresholds of 1 Å, 1.5 Å, 2 Å and 3 Å respectively. The same trends also apply; prediction RMSD increases with loop length, as does the RMSD difference between the predicted and best decoy.

The only significant difference between the four sets of results is in the cluster sizes. Intuitively this makes sense, since more decoys will be merged together if the RMSD between them is allowed to be higher. Similarly, if the threshold is low, decoys that may otherwise have been clustered together are split apart, and can lead to a large number of decoys not being grouped with others (singletons). This is indeed the case; for the highest threshold (3 Å), an average of 10 decoys were found to be singletons, while for the lowest threshold studied (1 Å), this number rose to 174.

The size of the largest cluster also decreases as loop length increases. This is because longer loops are able to have larger RMSDs between them, due to the increased conformational space. This may mean that a threshold that is suitable for one loop length may not be appropriate for another.

5.2.2 Calibur

The problem with pre-defining a cutoff value during clustering is that, if set too high, clusters that ideally should be separate could be merged together, or if too low, structurally similar decoys may be found in different clusters. We therefore decided to test Calibur (Li and Ng, 2010), an algorithm which automatically determines the cutoff value. Calibur clusters decoys in a different way to our simple UPGMA method: only Cα atoms are used in the RMSD calculation, and instead of building up the clusters gradually, each cluster is formed in one step. The first cluster is identified by finding the decoy that has the most neighbours within a distance threshold. These decoys are taken out of the set, and the procedure is repeated until no further clusters can be made. The RMSD threshold is chosen such that x percentage of the pairwise distances are below it; for a set of 500 decoys the value of x is 10. As Calibur superimposes structures before clustering them, we provided the whole protein structure including

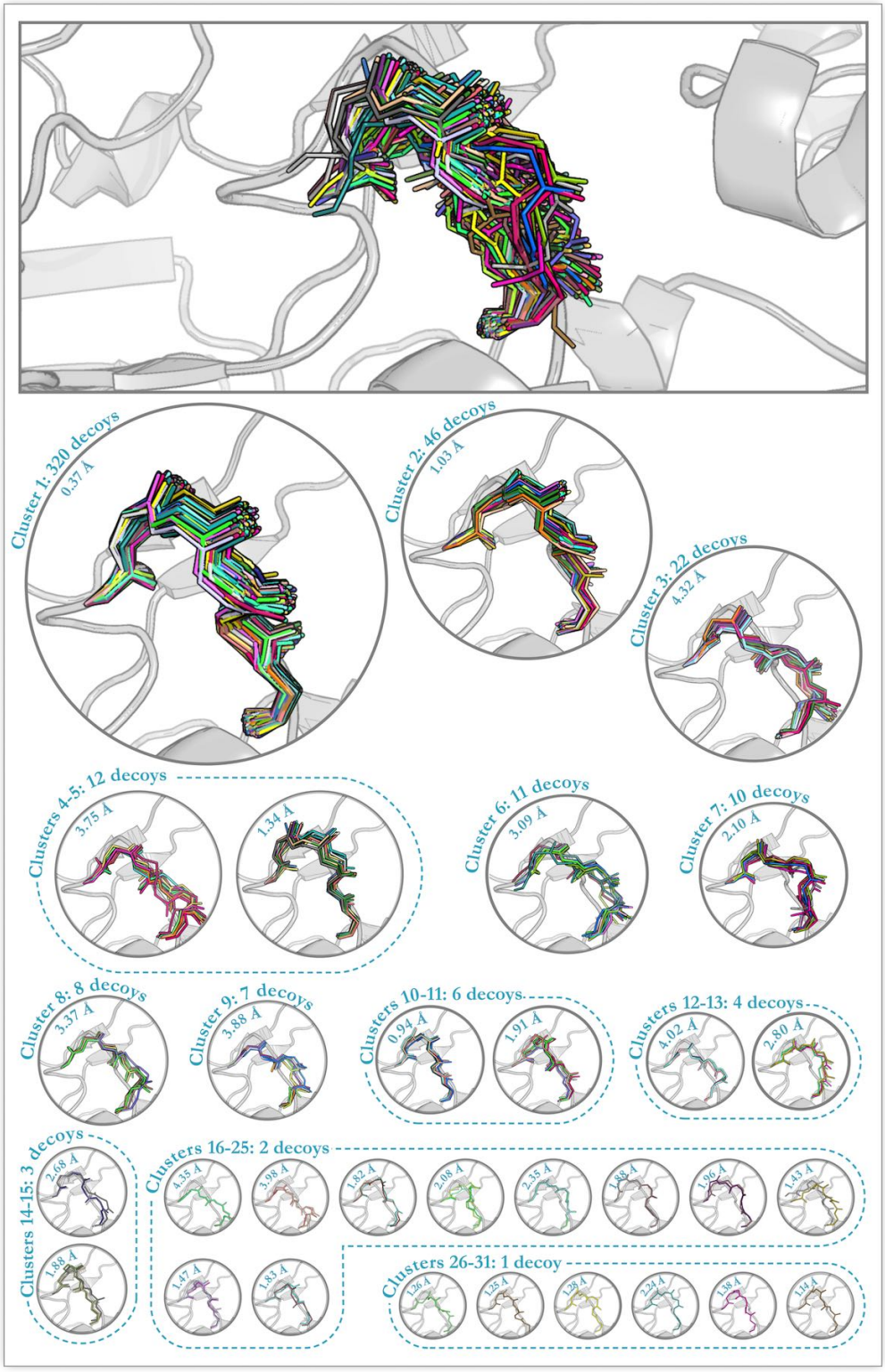

Figure 5.2: An example set of clusters produced by the UPGMA method (with a threshold value of 1Å, for the target 1trbA_289_294. RMSD values shown are those of the representative decoys in each cluster. The best structure generated has an RMSD of 0.21 Å, this is found in the largest cluster of 320 decoys — however the representative decoy of that cluster, and therefore the one given by this method as the top-ranked structure, has an RMSD of 0.37 Å.

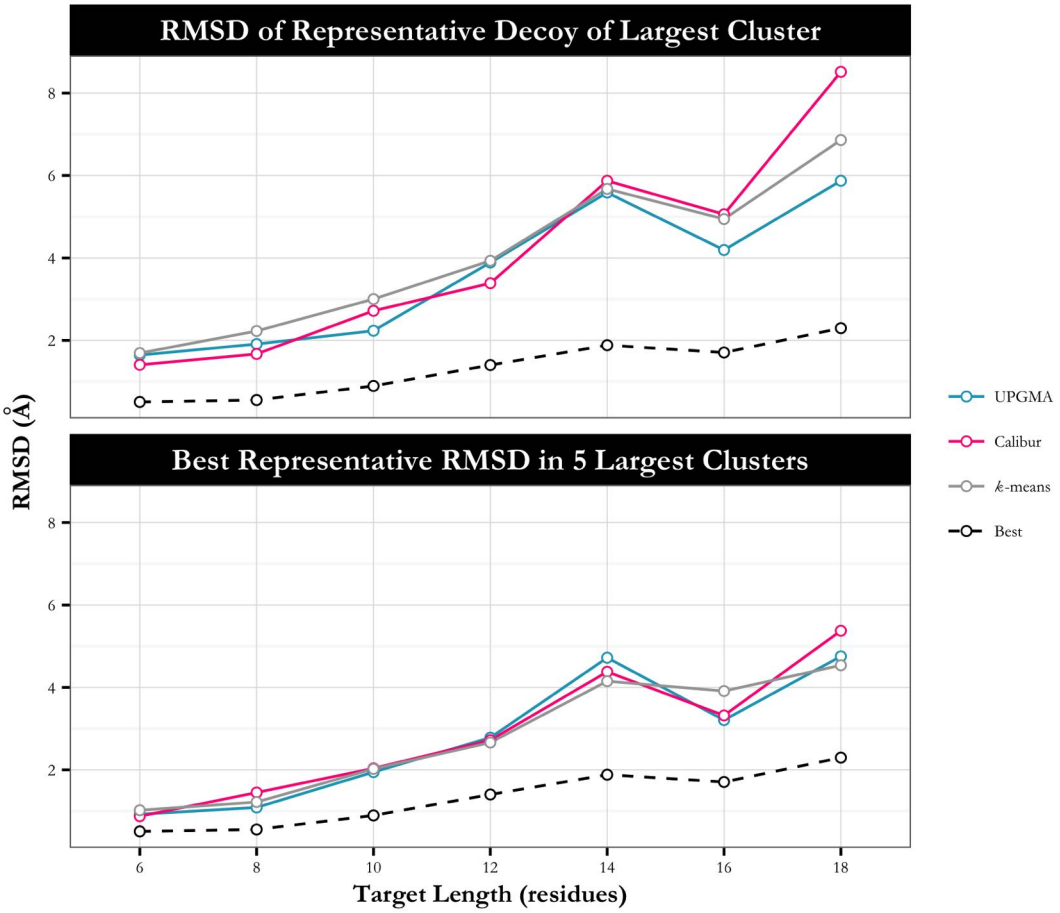

Figure 5.3: Results of using Calibur and *k*-means clustering to rank the decoys generated for the 70-loop general loop target set, compared to UPGMA with an RMSD cutoff of 1.5 Å. The two panels show the RMSD of the representative decoy from the largest cluster (top) and the best of the five largest clusters (bottom). *k*-means results shown are those generated using a *k* value of 45 (which produced the best top prediction). The best possible accuracy is shown as a black dashed line. All methods produce similar results.

the decoys as input, meaning that the loops should remain approximately in the correct position when the full structures are aligned.

A comparison between the ranking ability of Calibur and UPGMA (with a threshold of 1.5 Å) is shown in Figure 5.3. Calibur performs very similarly to UPGMA, with the same overall trends, however long loops of 16/18 residues are predicted more poorly. The average prediction RMSD is 4.09 Å, compared to 3.62 Å for UPGMA. The main difference between the two methods is in the sizes of the resulting clusters; while UPGMA is highly length-dependent in this regard, Calibur, by automatically calculating the threshold to be used, is not as affected. However, overall prediction

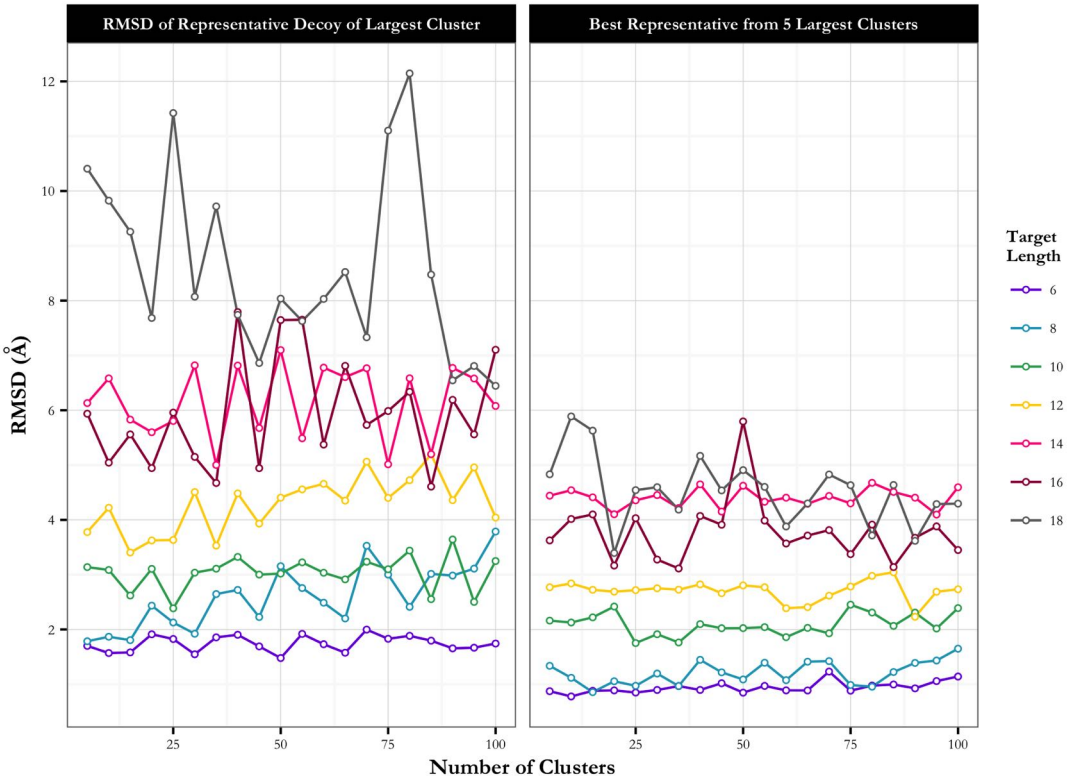

Figure 5.4: Result of using *k*-means clustering to rank the decoys of the general loop target set, whilst varying the number of clusters to divide the decoys into (*i.e.* the value of *k*). The left panel shows the average RMSD of the representative decoy from the largest cluster, and the right panel shows the best representative from the five largest clusters. There is no clear trend to indicate which value of *k* is the most appropriate.

accuracy is not improved.

5.2.3 *k*-means

Another method for clustering decoys is the *k*-means algorithm, in which the number of clusters (*k*) is pre-defined by the user. Starting with *k* decoys randomly selected as the first set of centroids (or means), the other decoys are partitioned into clusters depending on which of the centroids they are closest to. The centroids of each cluster are then re-evaluated, and the decoys are reassigned to clusters accordingly. This is repeated until the algorithm converges; *i.e.* the cluster each decoy is assigned to no longer changes. The clusters can then be treated in the same way as before, by ranking them according to size and choosing the representative of the largest as the final prediction.

To investigate the *k*-means method, we performed tests that varied the value of *k*

whilst monitoring the prediction accuracy (Figure 5.4). Values of k ranged from 5 to 100, in increments of 5. There is no clear trend in the data to indicate the optimum number of clusters, and prediction accuracy is generally poor — on average, the RMSD of the top-ranked decoy is 4.63 Å, approximately 1 Å worse than UPGMA.

5.3 Testing Scoring Software

Decoys can also be selected using a scoring function. These functions can be broadly classified into four types: physics-based, knowledge-based, empirical and machine learning. For protein loop structure prediction, physics-based and knowledge-based functions are most often used. Physics-based functions, or force fields, estimate the energy of a system through the summation of a number of terms, each representing the contribution to the energy of different features of the protein structure. These features include information about bonds within the structure, as well as non-bonded interactions.

Knowledge-based or statistical potentials, like the RAPDF-based function of Sphinx (described in Chapter 4), use experimentally observed structures to score protein models. Assuming that experimentally-determined conformations reflect energetics (*i.e.* structural motifs observed often are of low energy), comparing aspects of a model's structure to what has been observed in nature allows an energy of the system to be inferred. In theory, this means that solvation and any poorly understood interactions can be automatically considered, unlike a physics-based potential where all aspects of the system contributing to energy must be represented in the terms of the function. Knowledge-based potentials are often faster than physics-based force fields, and the resulting energy landscapes smoother. However, many loop modelling algorithms use physics-based potentials to select decoys and they have been shown in some circumstances to be more accurate (*e.g.* de Bakker *et al.*, 2003).

A drawback of using energy functions is that sidechains must first be added to the decoy structures in order for them to be scored correctly, since many of the interactions

that influence protein structure occur between sidechain atoms. Sidechain prediction is normally considered as a combinatorial search problem; each type of sidechain has a discrete set of structures characterised by their torsion angles, called rotamers. Prediction algorithms attempt to find the best combination of these rotamers (Miao *et al.*, 2011); examples of these algorithms are SCWRL (Canutescu *et al.*, 2003), RASP (Miao *et al.*, 2011), and TreePack (Xu and Berger, 2006). For Sphinx, we use SCWRL4 (Krivov *et al.*, 2009), since it is relatively computationally inexpensive — on average, the computational time required for sidechains to be predicted for a decoy structure is on the order of seconds. We only model the sidechains of the loop itself; the remainder of the sidechains remain in the same conformation as in the X-ray structure.

To investigate whether an energy function would be able to select near-native decoys from the sets generated by Sphinx, we selected nine freely available methods and used them to score and rank the decoys produced for the general loop target set. The nine methods are calRW and calRWplus (Zhang and Zhang, 2010); dDFIRE (Yang and Zhou, 2008); DOPE and DOPE_HR (Shen and Sali, 2006); the energy function used within FoldX (Schymkowitz *et al.*, 2005); GOAP (Zhou and Skolnick, 2011); the Rosetta scoring function (O'Meara *et al.*, 2015); and SOAP-Loop (Dong *et al.*, 2013). We also ran the RAPDF-based function used by Sphinx to select the reduced set of 500 decoys. Of the ten methods, two are physics-based force fields (FoldX and Rosetta) and the rest are statistical potentials. The results achieved using these ten methods are shown in Table 5.1, for both the top prediction and the best of the five top-ranked decoys.

On the whole, the energy functions achieve better results than were obtained using clustering approaches. An average prediction RMSD of 3.62 Å was the best result obtained using clustering (from UPGMA with a threshold of 1.5 Å); all but two of the scoring functions are more accurate than this, and the best value achieved was 2.71 Å (by dDFIRE). However, it is not clear which of the methods is the best — all of the methods apart from three (GOAP, Rosetta and the Sphinx function) produce results that are the best across all ten for certain lengths. There is not one method that consistently outperforms the others.

Table 5.1: Results when using ten published scoring functions to rank general loop decoys. The best method for each length is highlighted and the best accuracy possible is given in the top row in pink. The two tables refer to the RMSD of the top-ranked prediction (top) and the RMSD of the best decoy in the top five (bottom). No one method performs significantly better than the others.

Average RMSD of Top Prediction (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.51	0.55	0.89	1.41	1.89	1.70	2.30	1.32
calRW	1.03	1.28	2.30	2.79	3.78	4.31	4.87	2.91
calRWplus	1.04	1.20	2.16	**2.44**	3.74	4.10	**4.85**	2.79
dDFIRE	0.93	1.28	**1.38**	2.56	3.70	3.87	5.25	**2.71**
DOPE	1.04	1.02	2.33	2.85	3.69	3.84	5.55	2.86
DOPE_HR	0.92	**0.91**	1.60	2.77	**2.90**	3.76	6.67	2.73
FoldX	**0.70**	1.05	1.55	2.75	6.14	5.03	5.79	3.29
GOAP	2.21	1.58	1.99	4.18	4.14	4.99	7.33	3.75
Sphinx Function	1.62	1.85	3.76	4.98	4.35	7.30	10.28	4.80
Rosetta	0.87	1.49	1.94	3.23	6.04	4.91	6.29	3.54
SOAP-Loop	0.84	0.92	1.94	3.19	4.23	**3.09**	6.67	2.93

Average RMSD of Best of Top 5 (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.51	0.55	0.89	1.41	1.89	1.70	2.30	1.32
calRW	0.87	0.94	1.43	**1.78**	2.76	3.31	**3.31**	2.06
calRWplus	0.87	0.97	1.29	**1.78**	2.77	3.32	4.14	2.16
dDFIRE	**0.61**	0.95	1.26	2.06	2.73	3.15	3.36	2.02
DOPE	0.75	0.61	1.34	2.37	2.45	2.93	3.58	**1.98**
DOPE_HR	0.73	0.73	1.33	2.44	**2.21**	2.90	4.42	2.07
FoldX	0.63	0.70	**1.13**	2.32	4.43	3.87	5.18	2.61
GOAP	1.94	1.22	1.28	3.27	3.43	4.03	6.26	3.03
Sphinx Function	1.12	1.22	2.66	4.35	3.78	4.19	6.10	3.30
Rosetta	0.66	1.10	1.51	2.89	4.17	3.64	5.52	2.79
SOAP-Loop	0.67	**0.59**	1.53	2.58	3.18	**2.89**	4.81	2.29

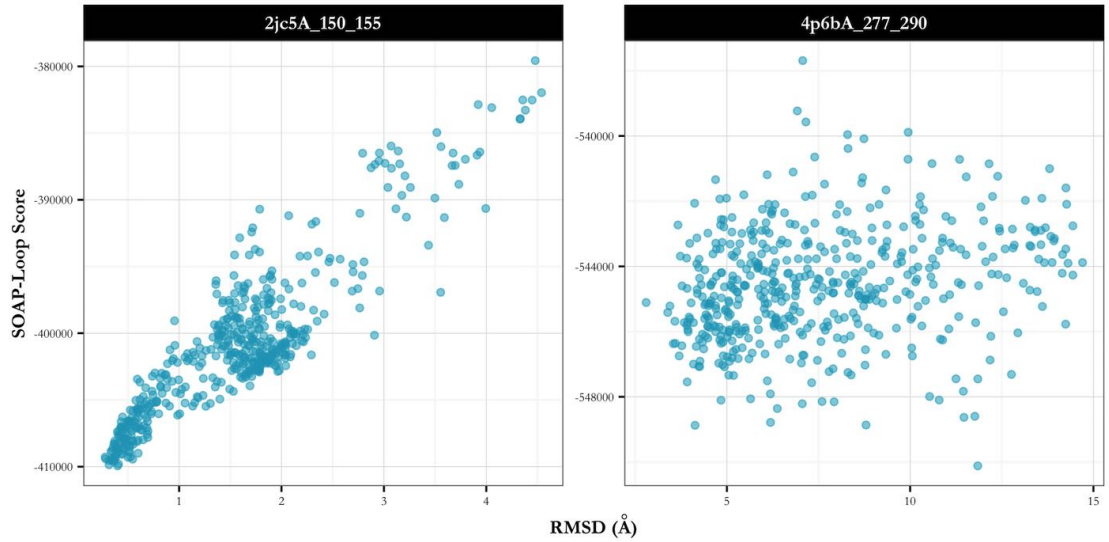

Figure 5.5: Example correlations of score and RMSD, for a good case (left, 2jc5A_150_155) and a poor one (right, 4p6bA_277_290). Pearson correlations between score and RMSD are 0.90 and 0.24 for 2jc5A_150_155 and 4p6bA_277_290 respectively. The scoring function used here is SOAP-Loop.

The accuracy of ranking varies between different targets. The target 2jc5A_150_155 is generally well-predicted — for example, SOAP-Loop selects a near-native decoy as the top prediction (with an RMSD of 0.4 Å), and the correlation between RMSD and the calculated score is high (see Figure 5.5). On the other hand, there are some targets for which the scoring functions are unable to select good decoys — for the target 4p6bA_277_290, the correlation between score (using SOAP-Loop) and RMSD is low, leading to a poor prediction (with an RMSD of 11.84 Å) .

We looked at each result on a target-by-target basis, to see which method produced the best result in each case (Figure 5.6). In terms of the top-ranked decoy, SOAP-Loop appears to be the best method, being more accurate than (or equal best with) the other methods on 24 occasions. SOAP-Loop also appears to be better at ranking decoys for long loops. The worst methods are GOAP and the Sphinx scoring function, achieving the best or equal-best result for 11 and 5 targets, respectively. When considering the top five decoys, SOAP-Loop still performs well, producing the best (or equal best) result for 29 targets, but is beaten by dDFIRE (achieving the best accuracy 31 times). Again, the worst methods are GOAP and the Sphinx scoring function, although Rosetta also performs poorly.

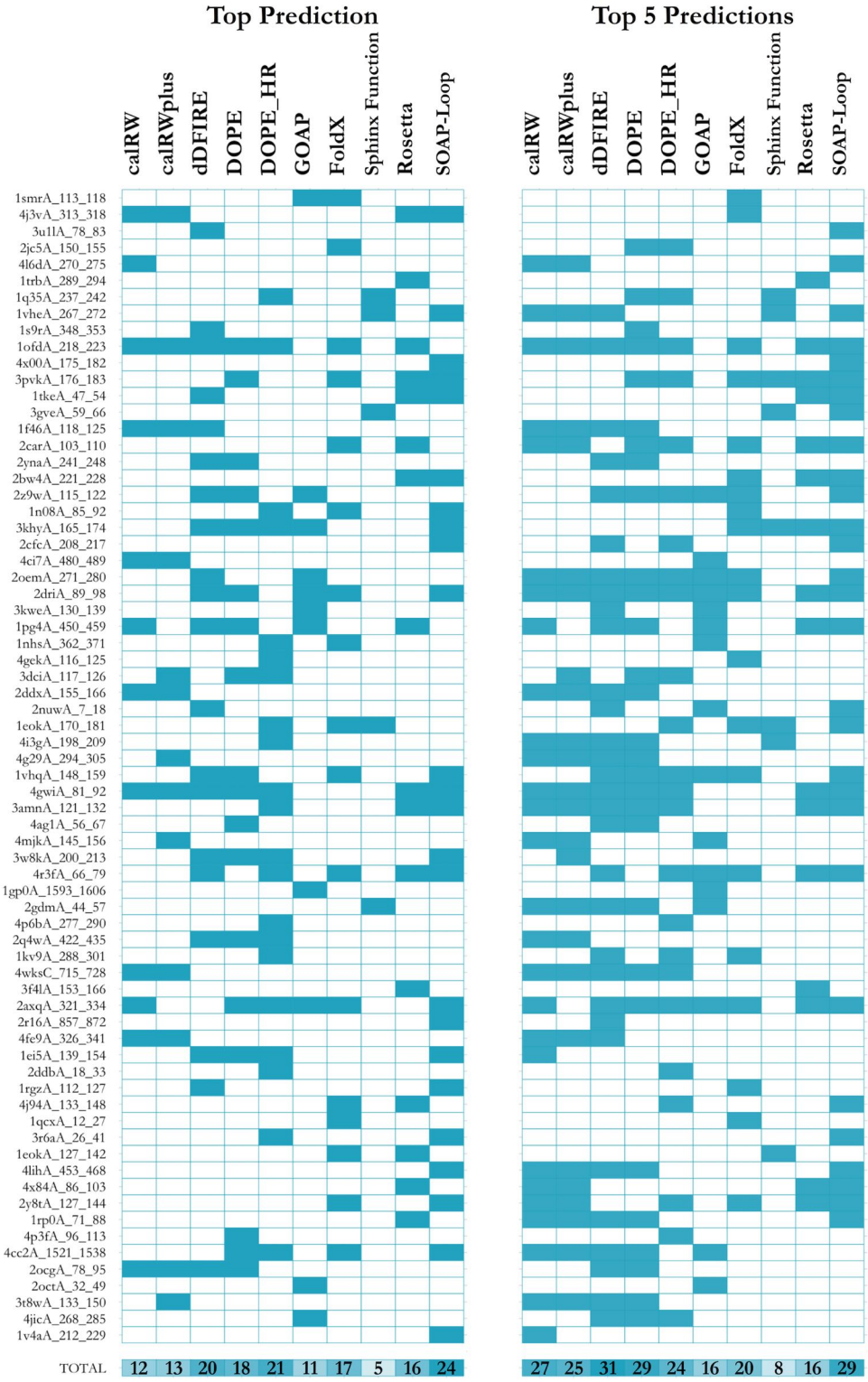

Figure 5.6: Which scoring function produced the best result for each target (ordered by length), considering the top prediction (left) and the best of the top five (right). The method that produced the best accuracy is highlighted in blue; if more than one function achieves the same accuracy, then all are highlighted. The bottom row shows the total number of times each method produced the best prediction.

We then investigated which methods were able to select the best decoy in the set, either as the top prediction or as one of the top five (Figure 5.7). Again, SOAP-Loop achieves the best results when looking at the top predictions, finding the most near-native decoy in the set for 10 targets. The worst method was the Sphinx function, which only found the lowest-RMSD conformation once. Also as before, the best method in terms of the best of the top five decoys is dDFIRE, placing the best decoy in the top five 24 times out of 70. This is still less than half of all targets, though, meaning most of the time the methods struggle to rank the decoys accurately.

There are some targets for which all scoring methods perform well. It appears to be the case that if there is a high-quality decoy in the set (with an RMSD of approximately 1 Å or below), it will be selected as the final prediction. For example, for the target 2driA_89_98, the best decoy in the set has an RMSD of 0.50 Å, and all but one of the methods ranked this decoy in the top five. On the other hand, for the target 1eokA_127_142, the best decoy generated had an RMSD of 2.85 Å, but the best prediction accuracy achieved by the ten methods was 9.67 Å.

5.3.1 Removing Small Clusters

Instead of using a clustering approach to select the best decoys, it could alternatively be used to remove poor quality decoys from the set before scoring with another method. Using the clusters from UPGMA with a threshold of 1.5 Å, we investigated whether removing decoys placed into small clusters, and subsequently ranking the remaining decoys with the dDFIRE statistical potential (Yang and Zhou, 2008), improves results. The results of this investigation are shown in Table 5.2. Three protocols were compared: the first involved ranking all decoys with dDFIRE with no clustering included; in protocol 2 all singleton decoys (those not clustered with others) are removed before ranking; and for the third protocol all decoys in clusters with three decoys or fewer are removed.

On the whole, protocols 2 and 3 result in approximately the same or worse predictions than simply ranking all decoys with dDFIRE. Some of the best decoys are found in these smaller clusters, and hence the average RMSD of the best decoy left

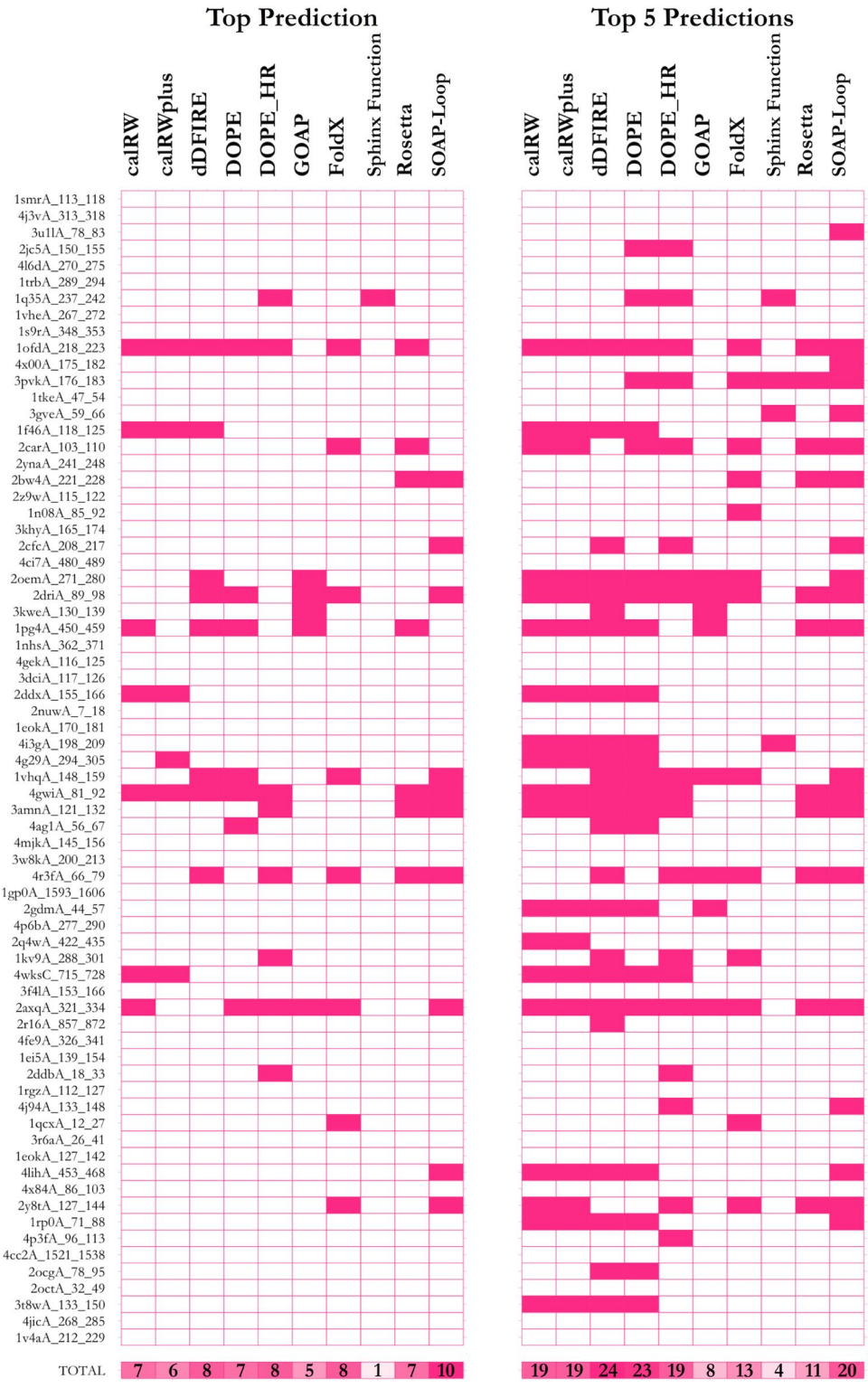

Figure 5.7: Which scoring functions were able to select the best decoy from the set, considering the top prediction (left) and the best of the top five (right). The methods that selected the best possible decoy are highlighted in pink. The bottom row shows the total number of times each method was able to select the best decoy.

Table 5.2: The effect of removing decoys that form small clusters from the decoy set, and subsequently ranking with dDFIRE. Protocol 1 — all decoys are ranked by dDFIRE; Protocol 2 — decoys that are not clustered into groups with others are removed; Protocol 3 — all clusters of three or fewer decoys are removed. Clustering is carried out using UPGMA with a threshold of 1.5 Å.

	Average RMSD of Best Decoy in Ranking Step (Å)		
Length	1	2	3
6	0.51	0.51	0.52
8	0.55	0.55	0.55
10	0.89	0.90	0.98
12	1.40	1.64	1.73
14	1.88	2.14	2.96
16	1.70	1.99	2.20
18	2.30	3.04	4.52
ALL	1.32	1.54	1.92

	Average Number of Decoys Ranked		
Length	1	2	3
6	500	496	484
8	500	483	437
10	500	462	371
12	500	433	341
14	500	344	224
16	500	317	186
18	500	209	102
ALL	500	392	306

	Average RMSD of Top Prediction		
Length	1	2	3
6	0.93	0.93	0.93
8	1.28	1.28	1.23
10	1.38	1.45	1.94
12	2.56	2.34	2.55
14	3.70	3.79	4.65
16	3.87	3.91	3.99
18	5.25	4.93	5.17
ALL	2.71	2.66	2.92

	Average Best RMSD of Top 5		
Length	1	2	3
6	0.61	0.61	0.61
8	0.95	0.95	0.99
10	1.26	1.29	1.32
12	2.06	2.08	2.15
14	2.73	2.73	3.54
16	3.15	3.02	3.41
18	3.36	4.04	4.95
ALL	2.02	2.10	2.42

in the set increases for both protocols. The average prediction RMSD is decreased slightly when removing singletons (from 2.71 Å to 2.66 Å), however the best RMSD of the five top-ranked decoys increases (from 2.02 Å to 2.10 Å). Protocol 3 is worse for both measures of accuracy.

Since there are fewer small clusters for short loops, the number of decoys left in the set after removing them does not decrease by much. For longer loops, the number is reduced more significantly. This means that the time needed to run dDFIRE on

the decoy set is reduced; however, since the accuracy of prediction is not improved, we did not select this methodology.

5.4 Combining Scores

It is possible that the accuracy of decoy selection could be improved by combining the results from the different methods. We first tried ranking the decoys for each target by calculating the average rank for each decoy from each method, and re-ranking by this value to give a consensus prediction (Table 5.3). This approach has been applied before by Cao *et al.* (2014) in their MULTICOM software, which was assessed during CASP10 and performed well (Kryshtafovych *et al.*, 2014).

Another option we have investigated is to average the scores themselves. Since the raw scores of the different methods and for the different targets cannot be directly compared, however, we first standardised them by converting them into z-scores:

$$z = \frac{x - \mu}{\sigma}$$

where x is the decoy score, and μ and σ are the mean score and standard deviation calculated across the whole decoy set.

The results achieved using these two consensus methods are shown in Table 5.3. The 'Average Ranks' approach does not improve prediction accuracies for any loop length, except for 6-residue loops when considering the five top-ranked decoys. The average prediction RMSD obtained using this method is 3.10 Å; six of the ten methods achieve accuracies better than this when run independently.

Conversely, ranking through averaging the z-scores of the decoys does improve results, giving an average top-prediction RMSD of 2.62 Å and giving a more accurate prediction than the ten individual methods for loops of length 6, 10 and 12. This is not a considerable reduction in RMSD, however; dDFIRE produces predictions that are (on average) only 0.09 Å less accurate.

Table 5.3: Combining scores by averaging ranks or z-scores from each of the ten scoring functions (bottom two rows), compared to the results achieved by each scoring function separately. The two tables show results considering the top-ranked decoy (top) and the best of the five top-ranked decoys (bottom). The best accuracy possible is shown in pink in the first row of each table. The best performing method for each loop length is highlighted.

Average RMSD of Top Prediction (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.51	0.55	0.89	1.41	1.89	1.70	2.30	1.32
calRW	1.03	1.28	2.30	2.79	3.78	4.31	4.87	2.91
calRWplus	1.04	1.20	2.16	2.44	3.74	4.10	**4.85**	2.79
dDFIRE	0.93	1.28	1.38	2.56	3.70	3.87	5.25	2.71
DOPE	1.04	1.02	2.33	2.85	3.69	3.84	5.55	2.86
DOPE_HR	0.92	**0.91**	1.60	2.77	**2.90**	3.76	6.67	2.73
FoldX	0.70	1.05	1.55	2.75	6.14	5.03	5.79	3.29
GOAP	2.21	1.58	1.99	4.18	4.14	4.99	7.33	3.75
Sphinx Function	1.62	1.85	3.76	4.98	4.35	7.30	10.28	4.80
Rosetta	0.87	1.49	1.94	3.23	6.04	4.91	6.29	3.54
SOAP-Loop	0.84	0.92	1.94	3.19	4.23	**3.09**	6.67	2.93
Average Ranks	0.80	1.00	1.87	2.64	4.69	3.98	6.72	3.10
Average Z-Scores	**0.67**	1.08	**1.14**	**2.41**	3.30	3.51	6.02	**2.62**

Average RMSD of Best of Top 5 (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.51	0.55	0.89	1.41	1.89	1.70	2.30	1.32
calRW	0.87	0.94	1.43	**1.78**	2.76	3.31	**3.31**	2.06
calRWplus	0.87	0.97	1.29	**1.78**	2.77	3.32	4.14	2.16
dDFIRE	0.61	0.95	1.26	2.06	2.73	3.15	3.36	2.02
DOPE	0.75	0.61	1.34	2.37	2.45	2.93	3.58	1.98
DOPE_HR	0.73	0.73	1.33	2.44	**2.21**	2.90	4.42	2.07
FoldX	0.63	0.70	1.13	2.32	4.43	3.87	5.18	2.61
GOAP	1.94	1.22	1.28	3.27	3.43	4.03	6.26	3.03
Sphinx Function	1.12	1.22	2.66	4.35	3.78	4.19	6.10	3.30
Rosetta	0.66	1.10	1.51	2.89	4.17	3.64	5.52	2.79
SOAP-Loop	0.67	**0.59**	1.53	2.58	3.18	**2.89**	4.81	2.29
Average Ranks	**0.60**	0.76	1.16	2.31	3.63	3.12	4.65	2.32
Average Z-Scores	0.57	0.77	**1.07**	2.11	2.58	3.14	3.35	**1.96**

5.5 Minimisation

As described in Section 5.3, in order to score using the ten functions it was necessary to add sidechains to our Sphinx decoys. We observed that often, these sidechains are not placed optimally and have minor clashes with the rest of the structure. These minor clashes may be causing problems for the scoring functions, causing them to record poor scores, even for correct backbones. In order to remove these clashes we decided to use a minimisation procedure.

We used the minimisation procedure within Rosetta. The protocol uses the kinematic closure (KIC) approach described in Chapter 1 to refine loop conformations (Mandell *et al.*, 2009). A kinematic or KIC move begins with the identification of three 'pivot' residues. New values for the dihedral angles of the non-pivot residues are then resampled, which opens the loop — the loop is re-closed by calculating dihedral angles for the pivot residues that produce a continuous backbone. Iterations of these kinematic moves and sidechain repacking are carried out, within a simulated annealing Monte Carlo framework — moves are accepted or rejected based on their relative energy to the previous conformation, with the probability of acceptance decreasing as the number of steps increases. In this way, the algorithm tries to find the minimum energy of the system according to the Rosetta energy function. This energetic minimum should, in theory, correlate to the true structure — the protocol should therefore move the decoys closer to the native and make it easier for the scoring functions to rank them correctly.

The effect of the Rosetta minimisation protocol on decoy RMSD is shown in Figure 5.8. The refinement procedure does not consistently move the decoy structures closer to the native conformation. Overall, 59.1% of decoys have lower RMSDs after minimisation, with an average improvement of 0.97 Å. Decoys that before minimisation had a high RMSD are more likely to be improved by the minimisation protocol; 60% of decoys that begin with an RMSD of over 1.5 Å are improved (with an average reduction in RMSD of 1.12 Å), while 55% of those with RMSDs below 1.5 Å are moved closer to the native structure (with an average RMSD improvement of 0.31 Å).

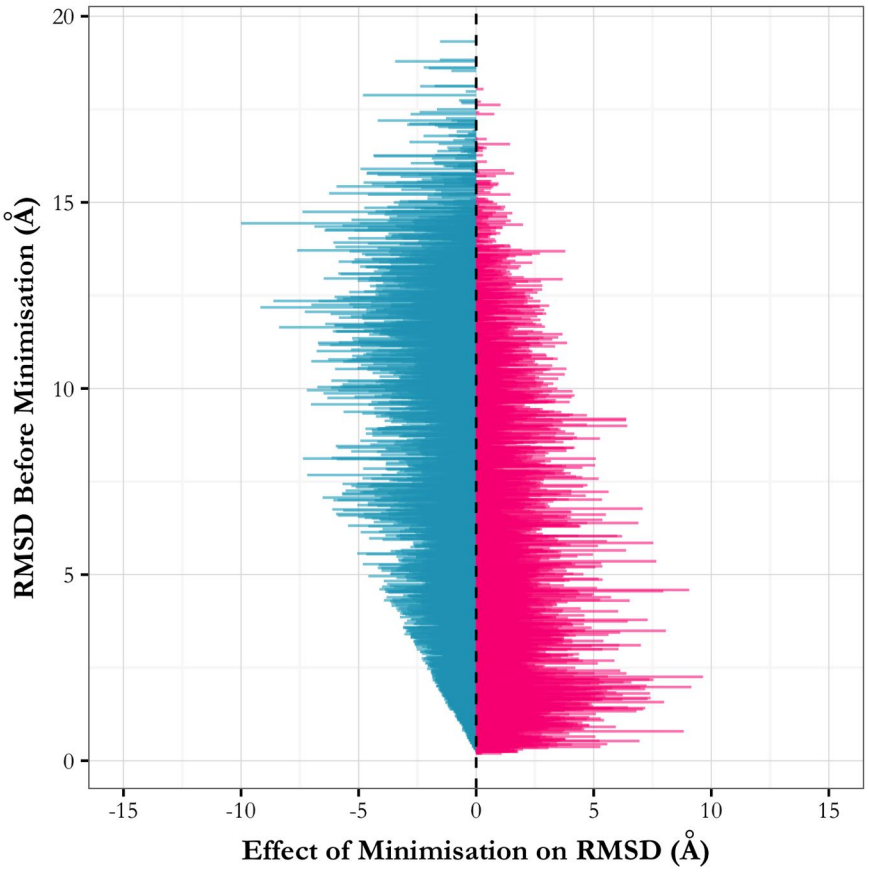

Figure 5.8: The effect of Rosetta minimisation on RMSD. The change in RMSD of each decoy is represented as a horizontal line — the position of the line on the y axis indicates the RMSD before minimisation, while the length of the line represents the change in RMSD. Decoys with improved RMSDs are shown in blue while those whose RMSDs were made worse by the minimisation step are coloured pink.

We tested the ten scoring functions described in Section 5.3 on the minimised decoy sets (Table 5.4). By comparing to the results in Table 5.1, it can be seen that, in general, minimisation does not have a large effect on the average RMSD of the top prediction, although the best result achieved falls from 2.71 Å (dDFIRE) to 2.31 Å (SOAP-Loop). However, there is a large effect on the average RMSD of the best decoy in the top five — for every method except one, the overall accuracy improves. The largest improvement is seen in the case of SOAP-Loop; the average RMSD goes down from 2.29 Å to 1.51 Å. This means that although the RMSDs of the decoys are not consistently improved by the minimisation procedure, it is improving the ability of the scoring functions to select good conformations from the set.

Table 5.4: Results when using ten published scoring functions to rank general loop decoys, after they have been refined using Rosetta. The best method for each length is highlighted (in bold and underlined). The best possible accuracy (the average RMSD of the best decoy in the set after minimisation) is given in the top row. The two tables show results considering the top-ranked decoy (top) and the best of the five top-ranked decoys (bottom). Results are improved by the inclusion of the minimisation procedure (compare to Table 5.1).

Average RMSD of Top Prediction (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.31	0.38	0.67	1.03	1.85	1.18	2.37	1.11
calRW	1.58	1.26	2.08	1.96	4.96	3.12	5.49	2.96
calRWplus	1.58	1.26	2.51	2.02	4.73	3.08	5.49	2.99
dDFIRE	1.35	0.94	**1.27**	2.50	4.58	4.03	5.26	2.88
DOPE	1.39	1.46	1.76	**1.90**	3.78	3.73	5.57	2.83
DOPE_HR	1.04	1.02	1.60	1.94	3.68	2.59	5.62	2.54
FoldX	0.80	**0.79**	1.91	2.17	4.51	2.35	6.78	2.82
GOAP	2.22	2.03	2.42	4.13	4.28	4.08	6.49	3.72
Sphinx Function	1.30	2.92	3.58	3.64	5.69	6.18	13.08	5.28
Rosetta	0.78	0.94	1.46	2.42	4.67	2.71	5.27	2.66
SOAP-Loop	**0.77**	1.06	1.57	1.93	**3.38**	**2.11**	**5.01**	**2.31**

Average RMSD of Best of Top 5 (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.31	0.38	0.67	1.03	1.85	1.18	2.37	1.11
calRW	1.00	1.02	0.83	1.54	3.32	1.93	3.02	1.83
calRWplus	0.92	1.03	0.83	1.36	3.33	1.74	3.02	1.77
dDFIRE	0.85	0.73	1.11	1.45	2.86	1.57	**2.93**	1.67
DOPE	0.87	0.81	**0.72**	1.43	2.63	1.75	3.02	1.63
DOPE_HR	0.65	0.77	0.80	**1.13**	2.57	1.97	4.05	1.73
FoldX	0.50	**0.48**	1.36	1.34	2.67	1.81	4.41	1.84
GOAP	1.53	1.52	1.07	2.00	2.71	2.92	4.65	2.37
Sphinx Function	1.02	1.00	2.50	3.08	4.61	5.21	6.70	3.49
Rosetta	**0.41**	0.52	1.02	1.76	3.20	1.79	3.50	1.78
SOAP-Loop	0.45	0.54	0.89	1.42	**2.44**	**1.44**	3.59	**1.57**

5.6 Crystal Contacts

During the investigations into decoy minimisation, we noticed that the decoys and the scoring performance for certain targets were not improved by the procedure. This lack of improvement could be due to the presence of crystal contacts. Crystal contacts are interchain or intermolecular contacts that occur solely as the result of protein crystallisation. They can be difficult to distinguish from interfaces that occur within the uncrystallised protein. Here we identify crystal contacts as atoms which which form hydrogen bonds to other proteins in the crystal, according to PyMOL (Schrödinger, 2010). Other copies of the protein within the crystal were generated using the 'symexp' function, and hydrogen bonds were detected using a distance cutoff of 3.6 Å and an angle of 63°. We found 34 loops in our general target set that had at least one hydrogen bond crystal contact.

By splitting the target set into two parts, one containing targets with crystal contacts and the other having the targets free of them, we can assess the effect of crystal contacts on prediction accuracy. The prediction results for the two sets, as ranked by the ten scoring functions described in Section 5.3, are shown in Figure 5.9. As most of the targets that do have contacts to other proteins in the crystal are long loops, we look at each length individually.

In general, loop targets with crystal contacts are less well predicted than those without, and the variation in prediction RMSD is greater. This is especially true for long loop lengths (*i.e.* 14 to 18 residue loops), where the difference in prediction quality can be up to 5 Å. The ability of the Sphinx algorithm to produce near-native conformations that remain in the 500-decoy reduced set is somewhat affected by the absence or presence of crystal contacts; the average RMSDs of the best decoy generated for loops with crystal contacts are 0.24 Å, 0.39 Å, 0.59 Å, 0.94 Å, 2.06 Å, 1.26 Å, and 2.60 Å for lengths 6 to 18 respectively. For those targets that do not form contacts with other protein molecules, Sphinx was able to generate decoys with RMSDs of 0.35 Å, 0.38 Å, 0.71 Å, 1.09 Å, 1.54 Å, 0.93 Å, and 1.97 Å. The main difference, however, stems from the abilities of the different scoring functions to rank the decoys. For example, Rosetta predicts 16-residue

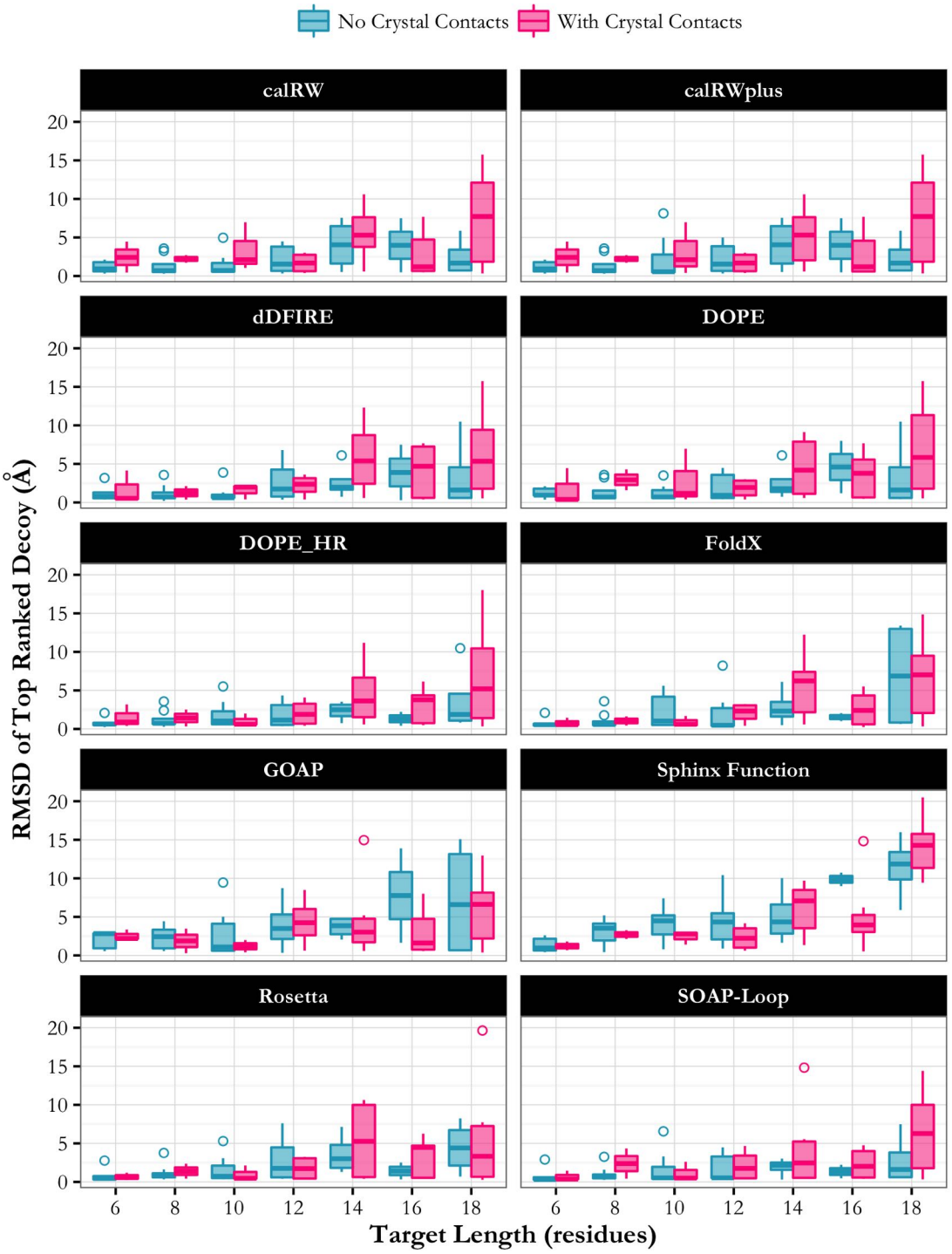

Figure 5.9: The effect of crystal contacts on prediction accuracy. Results shown are the RMSDs of the top predictions produced by each of the ten scoring functions. On the whole, prediction accuracy is worse for loops with crystal contacts.

loops without crystal contacts with an accuracy of 1.42 Å, but 16-residue loops that do form contacts with the rest of the protein are predicted with an average RMSD of 3.09 Å, even though the difference in the average best RMSD is only 0.33 Å. This is likely to be because none of the scoring functions used consider contacts to neighbouring proteins in the crystal — therefore the decoys are being scored in the incorrect environment.

We re-tested the scoring software as described in Section 5.3 to see if the minimisation procedure and the removal of targets that have crystal contacts has affected their results (Table 5.5). Overall, the ability of the methods to select good loop conformations is improved (compare to values in Table 5.1). On average, the RMSD of the top prediction returned by each method improves by 0.32 Å, but in some cases is reduced by more than an Ångström. The best performing method is SOAP-Loop, achieving an average prediction RMSD of 1.50 Å; 0.23 Å better than any other method. Also, when considering the five top-ranked decoys, SOAP-Loop managed to produce an average RMSD of 1.08 Å. This is the second best result of all the methods tested and is only 0.04 Å behind the first-placed DOPE_HR. SOAP-Loop produced the best result in 14 cases when considering the top-ranked decoy, and for 19 targets when looking at the top five, more than any other method. We have therefore selected SOAP-Loop as the scoring function to be used within Sphinx.

Since a short minimisation procedure improves the performance of Sphinx, we subsequently investigated if results could be further improved by carrying out a second, longer minimisation of the top five decoys, followed by a final SOAP-Loop scoring step. The final prediction accuracy is improved in 14 of the 36 cases, however on average, prediction accuracy is made worse by the second minimisation step. The average RMSD increases from 1.50 Å after the first minimisation to 1.99 Å after the second (from an initial RMSD, before any minimisation, of 2.56 Å). The average RMSD of the best decoy in the top five also increases, from 1.08 Å to 1.26 Å. There is also no clear length pattern — results that were improved came from a spread of target lengths, so there would be no benefit in using the second minimisation step for particular loop lengths (*e.g.* long loops).

Table 5.5: Results when using ten published scoring functions to rank general loop decoys, after they have been refined using Rosetta and targets forming crystal contacts to neighbouring proteins have been removed from the set. The best method for each length is highlighted (in bold and underlined). The best possible accuracy (the average RMSD of the best decoy in the set after minimisation) is given in the top row. The two tables show results considering the top-ranked decoy (top) and the best of the five top-ranked decoys (bottom).

Average RMSD of Top Prediction (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.35	0.38	0.71	1.09	1.54	0.93	1.97	0.89
calRW	1.14	0.98	1.52	2.14	4.04	3.99	**2.47**	1.98
calRWplus	1.14	0.98	2.23	2.24	4.04	3.99	**2.47**	2.13
dDFIRE	1.18	0.85	**1.16**	2.70	2.71	3.90	3.55	1.95
DOPE	1.23	1.04	1.29	1.95	2.61	4.60	3.54	1.92
DOPE_HR	0.82	0.90	1.84	1.85	2.32	**1.32**	3.77	1.73
FoldX	**0.78**	0.72	2.35	2.24	2.81	1.54	6.95	2.27
GOAP	2.01	2.07	2.94	3.95	3.64	7.77	7.23	3.65
Sphinx Function	1.34	2.98	4.08	4.53	5.09	9.87	11.41	4.73
Rosetta	0.84	0.81	1.66	2.85	3.62	1.42	4.44	2.07
SOAP-Loop	0.79	**0.69**	1.74	**1.80**	**1.91**	1.35	2.82	**1.50**

Average RMSD of Best of Top 5 (Å)

	Target Length							
Method	6	8	10	12	14	16	18	ALL
Best	0.35	0.38	0.71	1.09	1.54	0.93	1.97	0.89
calRW	0.73	0.70	0.88	1.47	2.74	1.17	2.01	1.27
calRWplus	0.65	0.71	0.88	1.47	2.78	1.17	2.01	1.26
dDFIRE	0.66	0.69	1.08	1.72	1.80	1.17	2.01	1.23
DOPE	0.75	0.70	**0.73**	1.58	1.84	1.25	**1.99**	1.16
DOPE_HR	0.48	0.67	0.78	**1.13**	**1.76**	1.18	2.03	**1.04**
FoldX	0.49	**0.43**	1.63	1.21	2.17	1.17	3.26	1.35
GOAP	1.51	1.73	1.15	1.86	2.25	5.51	4.86	2.24
Sphinx Function	1.20	1.07	2.77	3.84	3.93	7.90	7.27	3.27
Rosetta	0.42	0.44	1.06	2.10	3.34	1.17	2.45	1.42
SOAP-Loop	**0.39**	0.50	0.99	1.14	1.86	**1.15**	2.37	1.08

5.7 Final Protocol Results

From the results presented, we designed the ranking protocol of our sets of 500 decoys to be as follows:

1. Sidechain addition to the modelled loop conformations using SCWRL4 (Krivov *et al.*, 2009);
2. Minimisation with Rosetta, performing a short run on each decoy;
3. Final scoring and ranking using SOAP-Loop.

The following sections present the results achieved by using this protocol on the general target set as well as the H3 loop target set (excluding those target loops with crystal contacts), as described in Chapter 4.

5.7.1 General Loop Target Set

The results achieved by Sphinx, both with and without minimisation for comparison, for the 36 general loop targets are shown in Figure 5.10. Overall, the results achieved by Sphinx are improved by the inclusion of minimisation (Figure 5.10). This appears to be due to SOAP-Loop's enhanced ability to select near-native decoys — the mean RMSD of the top-ranked decoy is reduced from 2.56 Å to 1.50 Å, and the mean RMSD of the best decoy in the top five decreased from 1.78 Å to 1.08 Å. In addition, the number of sub-Ångström top-ranked predictions is increased from 18 to 24.

A drawback to including a minimisation step is the increased execution time. The average time increases by approximately ten hours (when using one core of a Linux server with 128 GB RAM, and a 2.1 GHz processor), since performing minimisation on 500 protein structures is computationally expensive.

We have compared Sphinx's performance using this protocol to that of its constituent algorithms, FREAD and MECHANO (see Chapters 2 and 3 for details of these two algorithms). The results are shown in Figure 5.10. FREAD is representative of using Sphinx to find fragments that are the same length as the target, and simply inserting

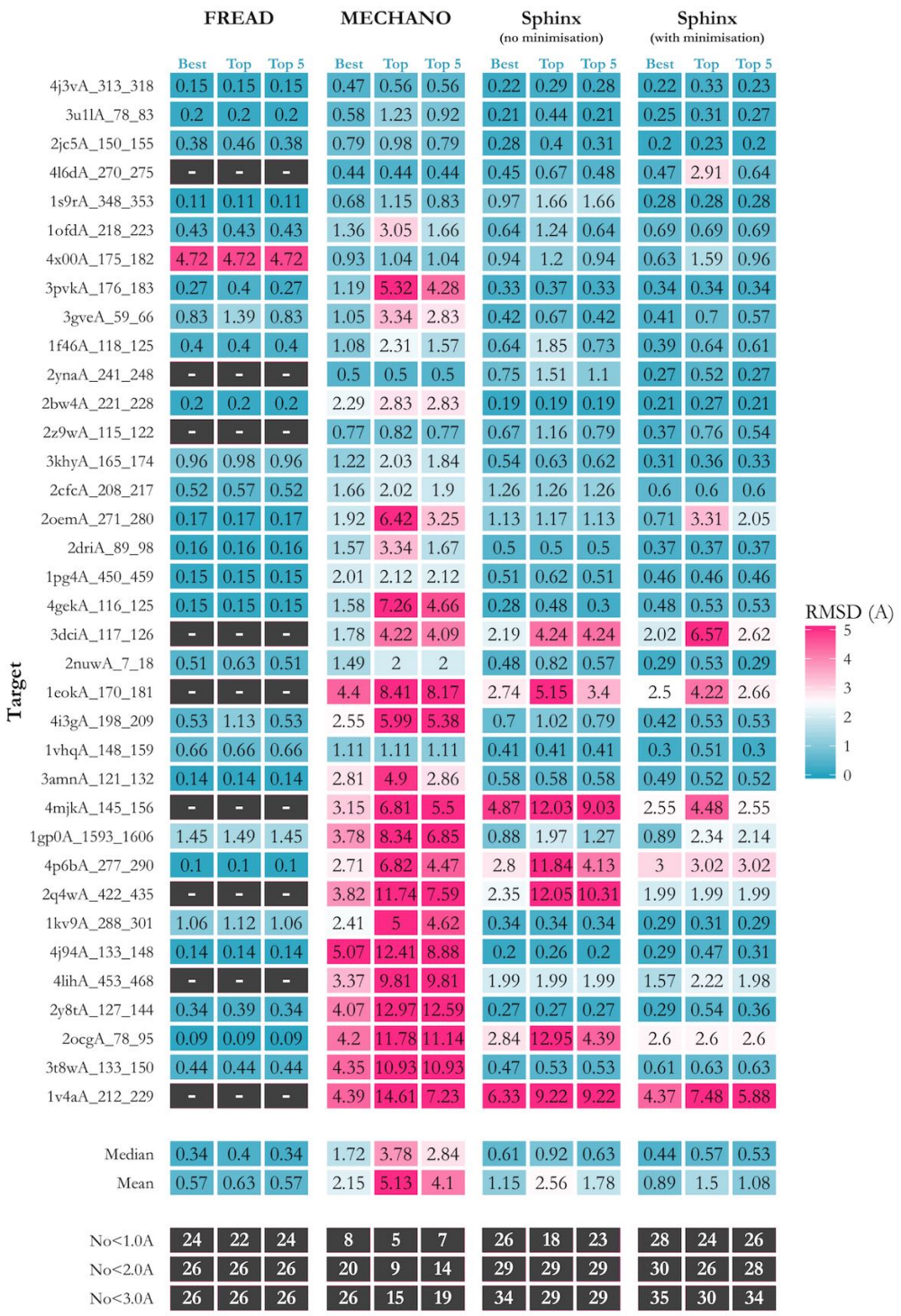

Figure 5.10: Comparison of Sphinx (with and without a minimisation step) to FREAD and MECHANO, for the general target set (after target loops with crystal contacts are removed). The three columns for each method show the results considering the best decoy generated, the top-ranked decoy, and the best of the top five. Sphinx outperforms its constituent *ab initio* method MECHANO both with and without minimisation, and gives 100% coverage (unlike FREAD).

them into the gap in the structure. MECHANO can be thought of as running Sphinx without any fragment information — *i.e.* all bond lengths, angles and dihedral angles are chosen randomly from the relevant distributions, allowing us to assess how the addition of structural data from fragments affects results. To ensure a fair comparison, MECHANO was run so as to generate the same number of decoys for each target as Sphinx (around 11,000 on average), and its decoys were ranked in the same way (excluding minimisation). The results in Figure 5.10 consider only the 500 decoys selected by our loop-specific energy function and the top decoys are selected with SOAP-Loop, either before or after minimisation with Rosetta.

The results show that FREAD is more accurate than Sphinx on average, but there are nine cases — a quarter of the target set — for which FREAD failed to find suitable fragments in the database. Sphinx, on the other hand, returns a prediction in every case. Sphinx also gives more top-ranked decoys with sub-Ångström RMSDs, with 24 compared to 22 for FREAD. On a case-by-case basis, Sphinx produced a better prediction than FREAD for 18 targets (including those for which FREAD failed).

Sphinx, by including information from fragments, is able to considerably outperform its base *ab initio* algorithm, MECHANO. When considering the top 500 decoys, Sphinx is able to generate more accurate conformations — the average RMSD of the best decoy produced was 0.89 Å, compared to 2.15 Å for the *ab initio* algorithm, and overall the lowest-RMSD decoy generated by Sphinx was better than that of the *ab initio* algorithm in 33 cases. The *ab initio* algorithm could only produce a conformation with an RMSD below 1 Å for 8 targets; Sphinx gave 28.

The RMSDs of the top-ranked decoys (as predicted by SOAP-Loop) are significantly better when using Sphinx compared to MECHANO, with an average RMSD of 1.50 Å compared to 5.13 Å for the *ab initio* algorithm. The five top-ranked decoys contained 26 and 7 sub-Ångström conformations for Sphinx and the *ab initio* method respectively. The improved accuracy is especially noticeable for longer loops — for the 16 targets of 12 or more residues, the *ab initio* method gave no sub-Ångström predictions, while Sphinx was able to produce 7. When considering the top prediction, MECHANO

could only outperform Sphinx for four targets, and only one when looking at the five top-ranked decoys.

5.7.2 H3 Prediction on Crystal Structures

In the previous chapter, we described an H3-specific version of our algorithm, Sphinx-H3 and its ability to produce decoys for a set of 52 H3 loops, taken from the set used to test RosettaAntibody (Sivasubramanian *et al.*, 2009). Since we have found that crystal contacts affect prediction quality, we have removed those structures with crystal contacts from this set, leading to a final number of 39 H3 loops. We have applied our final ranking protocol (Rosetta minimisation plus SOAP-Loop scoring) to these decoys, to give final predictions. As before, we compare Sphinx's predictions to those generated using RosettaAntibody, a leading H3-specific *ab initio* loop modelling algorithm (generating 500 decoys per target. These decoys are ranked within the RosettaAntibody algorithm using the Rosetta energy function. Figure 5.11 shows the results of this comparison.

As discussed previously in Chapter 4, the ability of Sphinx-H3 to generate near-native loop conformations is comparable to RosettaAntibody: the average RMSD of the best decoy generated is 0.93 Å for Sphinx-H3 (or 1.11 Å without minimisation) compared to 1.07 Å for RosettaAntibody. After decoy ranking, Sphinx-H3 outperforms RosettaAntibody, producing an average top-ranked RMSD of 2.5 Å compared to 3.38 Å, and giving predictions below 2 Å for 17 targets (compared to 13 for RosettaAntibody). On a target-by-target basis, Sphinx gave a more accurate prediction than RosettaAntibody 24 times, increasing to 28 when considering the best of the five top-ranked decoys.

In addition to achieving better accuracies, Sphinx is also much faster than RosettaAntibody. RosettaAntibody takes approximately one hour to generate a single loop conformation (and hence up to 500 hours to generate the whole decoy set); Sphinx took (on average) 14 hours per target to generate, minimise and rank all decoys (when using one core of a Linux server with 128 GB RAM, and a 2.1 GHz processor), with the majority of the time being spent on minimisation (approximately 12 hours).

Interestingly, the accuracy of the top prediction is decreased when a minimisation step is included — the average RMSD increases slightly from 2.43 Å to 2.50 Å, and the

number of sub-Ångström predictions goes down from 13 to 9. However, the results when considering the best of the five top-ranked decoys are improved. The average RMSD decreases from 1.93 Å to 1.52 Å and the number of sub-Ångström predictions goes up from 13 to 17. Both with and without minimisation, however, Sphinx performs better than RosettaAntibody.

Kinked or Extended H3 Conformations

H3 loops can be classified as either 'kinked' or 'extended' depending on the conformation of their C-termini, with the majority being kinked (Weitzner *et al.*, 2015). Using the definition of Weitzner *et al.*, 2015, we tested how often Sphinx correctly predicts a kinked or extended conformation. For the 30 kinked loops in the H3 target set, the top-ranked prediction of Sphinx was also kinked in 24 cases. This compares favourably to Rosetta, which only produced a top-ranked prediction of the correct conformation for 14 of the kinked targets. For both algorithms, five of the nine extended loops were predicted with the correct conformation. We believe this accuracy in predicting a kinked conformation is a result of using fragments; using sequence similarity to choose fragments, and copying across similar regions of structure from their structures, helps to guide Sphinx towards the correct kinked or extended conformation.

5.7.3 H3 Prediction on Model Structures

In Chapter 4, we also presented results that show how Sphinx is able to generate decoys when the rest of the protein structure is not in its native conformation by running it on the set of 52 targets from the RosettaAntibody paper (Sivasubramanian *et al.*, 2009). The decoys were built onto model structures of the 39 antibodies in the set for which the H3 loop formed no crystal contacts. Models were generated using ABodyBuilder (Leem *et al.*, 2016). We ran the ranking protocol on these decoys, to see how accuracy is affected (Figure 5.12). In the model environment, anchor residues are in a non-native conformation, making prediction more difficult. As expected, the results are not as good as when using the native environment — the average top-ranked RMSD increases from 2.5 Å to 3.21 Å, and fewer sub-Ångström predictions are made. Interestingly,

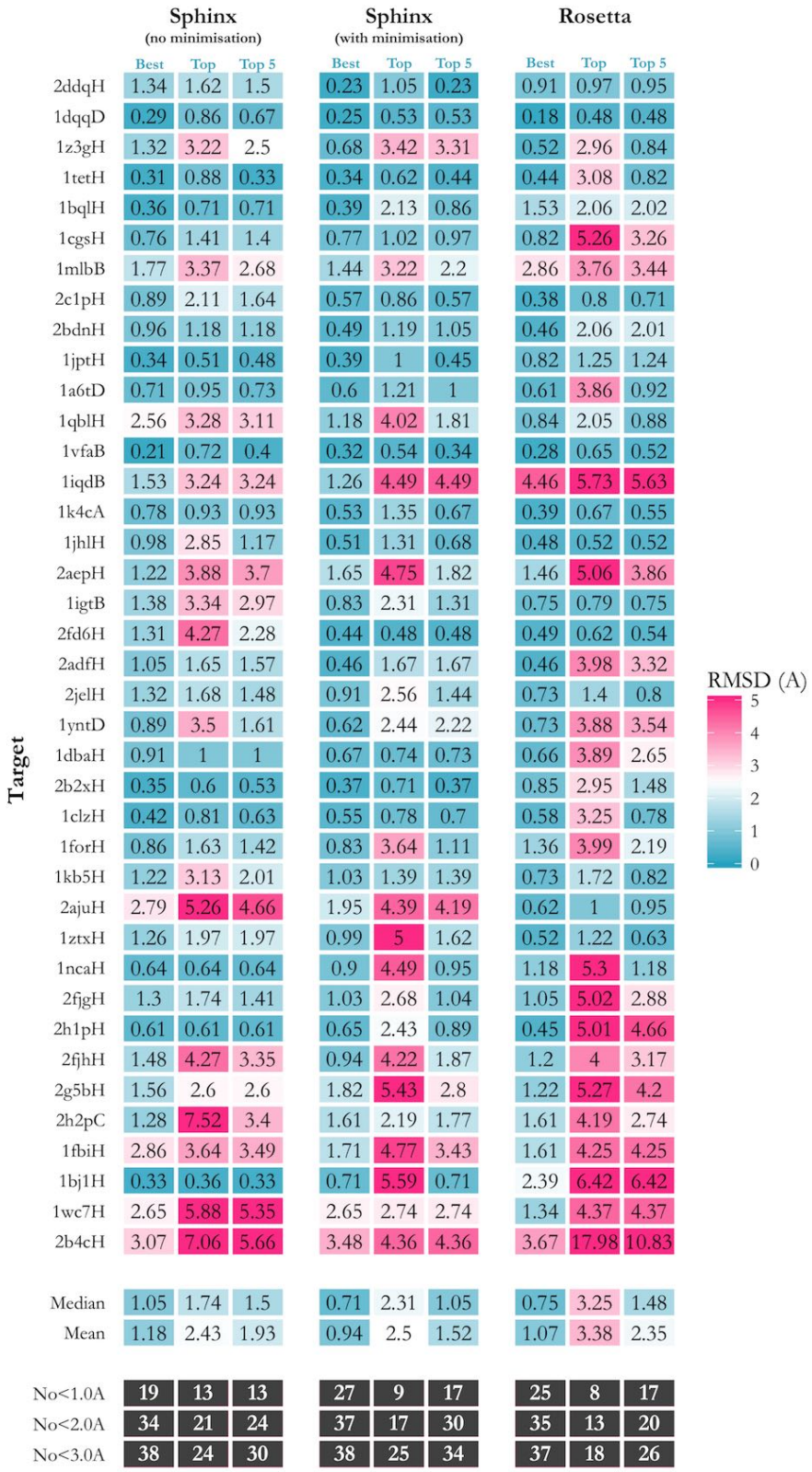

Figure 5.11: Results for using Sphinx (with and without minimisation) to predict the structures of 39 H3 loops (using SOAP-Loop to perform the final ranking), compared to RosettaAntibody. Target loops are in order of length, with the shortest at the top. Both versions of Sphinx produce better predictions, on average, than RosettaAntibody, especially for long loops.

Table 5.6: Results for using Sphinx to predict the structures of 4 H3 loops from the RosettaAntibody benchmark set in an inexact environment, with either no minimisation, minimisation of just the loop region, or minimisation of the whole structure.

Target	Length	No Minimisation			Loop Minimisation Only			Full Minimisation		
		Best	Top	Best of Top 5	Best	Top	Best of Top 5	Best	Top	Best of Top 5
1mlbB	7	1.99	2.99	2.71	1.59	3.52	3.52	1.90	3.75	2.56
2jelH	9	1.31	1.50	1.34	1.18	2.75	2.12	1.44	1.90	1.90
1forH	10	1.79	1.90	1.79	1.19	3.48	2.82	1.31	3.07	2.53
2h2pC	12	2.86	7.60	7.60	3.33	7.67	6.60	1.89	3.06	2.27

however, unlike the previous tests where minimisation improved results, here it appears to decrease accuracy. The average top-ranked RMSD and the average RMSD of the best decoy in the five top-ranked stay approximately the same, but the number of targets predicted with an RMSD below 2 Å decreases from 18 to 10.

While minimisation improves results for predictions carried out in the native environment, this does not occur when predicting onto model structures. On average, the RMSD of the best decoy in the set is improved, however ranking becomes less accurate. This could be because although the models onto which the decoys are built are good, they do not have atomic accuracy, and hence errors exist — especially in the conformations of the sidechains. As such, the loop minimisation is being performed in the non-native environment, which may affect results adversely. We tested whether minimising the entire structure, and not just the loop region, could improve results for four targets (Table 5.6). Again, we used Rosetta to minimise the structures, running the 'relax' protocol which features successive sidechain repacking and minimisation steps. Overall, it seems that minimising the entire structure is better than just minimising the loop region, however results still appear to be better when using no minimisation at all. An exception to this is the case of the target 2h2pC, where the original results with no minimisation were extremely poor.

We have compared our results to those published by the authors of RosettaAntibody (Sivasubramanian *et al.*, 2009) and H3Loopred (Messih *et al.*, 2014). While the best decoy produced by Sphinx is of the same or better quality to that of the other two

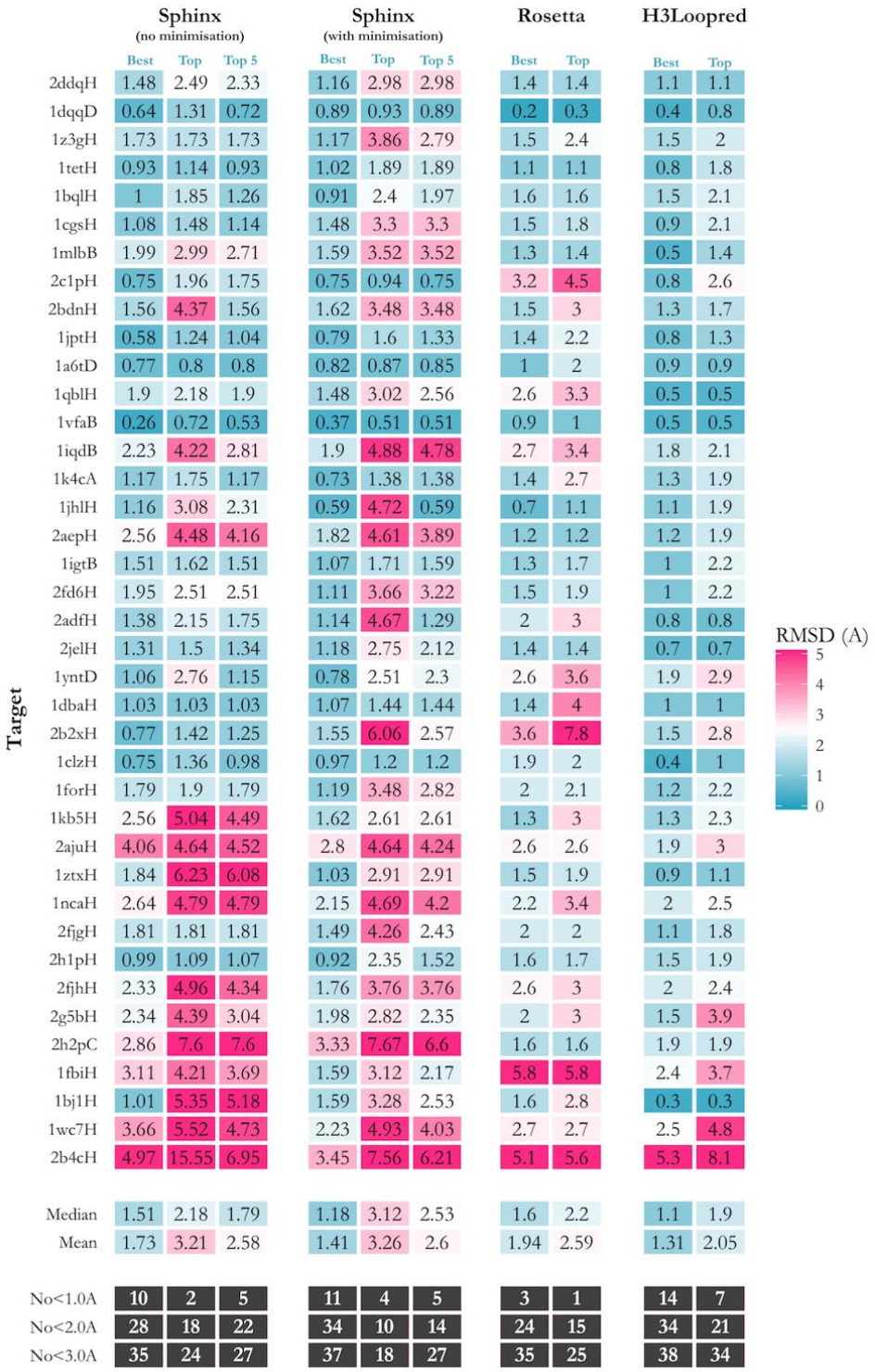

Figure 5.12: Results for using Sphinx (with and without minimisation) to predict the structures of 39 H3 loops in an inexact environment, compared to RosettaAntibody and H3Loopred. The three columns for each Sphinx method show the results considering the best decoy generated, the top-ranked decoy, and the best of the top five. Data was only available for the 'best' and 'top' columns for RosettaAntibody and H3Loopred. Target loops are in order of length, with the shortest at the top. While the decoys generated by Sphinx are of better or similar quality to the other two methods, the ranking procedure is not as effective.

methods (1.41 Å compared to 1.94 Å and 1.31 Å for RosettaAntibody and H3Loopred respectively), RosettaAntibody and H3Loopred are clearly better able to rank their decoys accurately. The data is not available to compare the best of the top five decoys, however when considering the top prediction, Sphinx is on average 0.67 Å worse than RosettaAntibody and 1.21 Å less accurate than H3Loopred. In addition, it should be noted that as the results for RosettaAntibody and H3Loopred were taken from the literature, the antibody models on which loops were predicted vary between methods.

5.7.4 Antibody Modelling Assessment II

In 2014, the second Antibody Modelling Assessment (AMA-II) was held; a CASP-style blind prediction test specifically for antibody structure prediction. The second round of this competition focused entirely on H3 prediction — participants were given the crystal structures of ten antibodies with the H3 loops removed, and asked to model the missing residues (allowing them to submit five possible models per target). We have used Sphinx to predict these H3 structures, to further compare our results with those of other groups (Figure 5.13). We used the same protocol as that used for the previous H3 loop set, except that fragments from PDB structures deposited later than March 2013 were ignored (to ensure a fair comparison). We report our results using the same RMSD measure as used in AMA-II — using only the carbonyl C and O atoms of residues 95-100x.

On average, the best of the five models returned by Sphinx has an RMSD of 1.41 Å. This compares well to the other groups, who achieved RMSDs of 1.86 Å, 2.09 Å, 1.97 Å, 1.25 Å, 2.41 Å and 1.12 Å (for Accelrys, CCG, the Gray group, the Shirai group, Macromoltek and Schrödinger, respectively). Only the Shirai group and Schrödinger achieved better results than Sphinx according to this measure.

When looking at the average RMSD across all models, Sphinx outperforms all the other methods with an RMSD of 2.17 Å compared to 2.89 Å, 3.08 Å, 3.22 Å, 2.43 Å, 3.08 Å, and 2.54 Å. Sphinx is therefore more consistent than the other methods. For two targets (Ab04 and Ab10) the best model from Sphinx had a lower or the same RMSD as the best produced from all other methods. In addition, Sphinx was able to

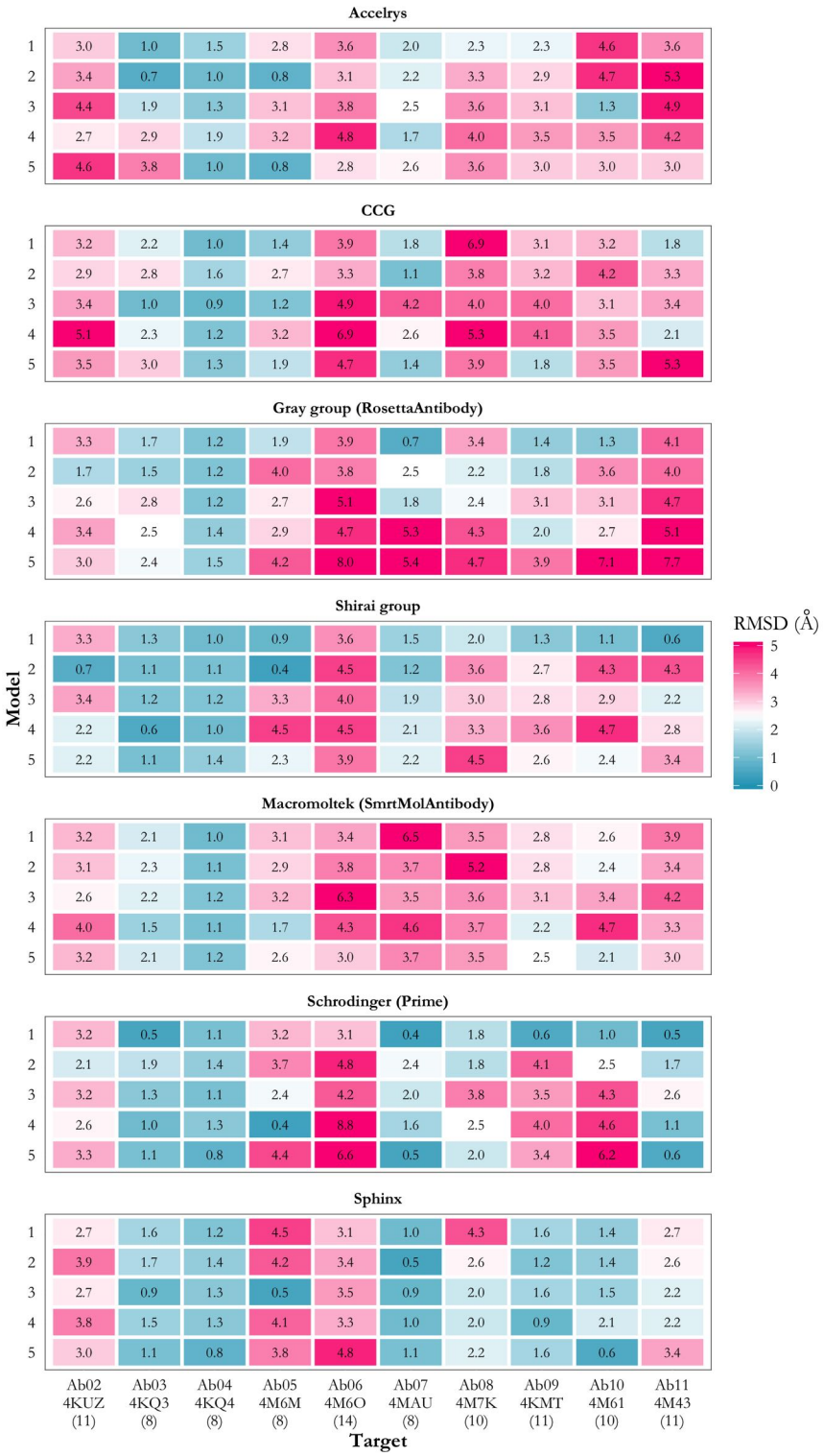

Figure 5.13: Performance of Sphinx on the target set from the Antibody Modelling Assessment II, compared to the actual participants. PDB codes and loop lengths are given underneath the target ID. In terms of the average RMSD of the best of the five models, Sphinx produces better results than all but two of the other methods (Prime and Kotai Antibody Builder).

produce a sub-Ångström prediction for six of the ten targets, which is the most achieved by any method (equal to the number generated by Schrödinger).

5.8 Conclusions

In this chapter we have described our investigations into various ranking methods, in an attempt to find an accurate decoy selection method that can be incorporated into the Sphinx algorithm. Clustering approaches were found to achieve poor accuracy when used to select decoys for the general loop target set; the best method tested was UPGMA when using a threshold of 1.5 Å, but the average prediction had an RMSD that was 2.30 Å higher than the best possible given the decoy sets. This may be due to the way in which Sphinx builds decoys — since fragments are used, there will be subsets of decoys with similar structures that could be detected using clustering — however, these decoys are not necessarily the most accurate.

After adding sidechains to each decoy structure using SCWRL4, we tested ten scoring functions (two physics-based and eight knowledge-based) to assess their ability to select near-native decoys. We observed that predictions for targets for which a very high-quality conformation is present in the decoy set are generally of greater accuracy; *i.e.* it seems that structures closer to the native structure are scored more accurately. This could be because errors in the backbone conformation lead to errors in sidechain prediction, and conversely near-native backbone structures allow SCWRL to find the correct sidechain conformations. Scoring functions may be less able to differentiate between low-RMSD and high-RMSD structures if the majority of the decoys in the set feature multiple errors in sidechain location.

We have demonstrated that, except potentially in the case of modelling H3 in the non-native environment, the inclusion of a minimisation procedure improves prediction accuracies, even though decoy RMSDs may not be consistently improved. This appears to be because the ability of the ranking method to correctly select near-native decoys is

enhanced. However, this does not occur when predicting onto model structures — this could be because although the models onto which the decoys are built are good, they do not have atomic accuracy, and hence errors exist — especially in the conformations of the sidechains. As such, the loop minimisation is being performed in the non-native environment, which may affect results adversely. Preliminary results suggest minimising the entire protein structure, not just the loop region, is the best next step.

The presence of hydrogen bonds between proteins in the crystal for some targets makes prediction more difficult. This is presumably because the modelling pipeline considers the protein chain in isolation, and ignores any interactions that may be present due to other molecules in the crystal. These crystal contacts may even force the protein loop to adopt a conformation that it would not normally do when found *in vivo* — removing these targets from the set, therefore, is a sensible choice, and one that improves the overall accuracy of prediction.

The results presented here in this chapter have led to the selection of the final ranking protocol to be incorporated into Sphinx: minimisation using Rosetta, followed by scoring and ranking using the energy function SOAP-Loop. We have therefore fully developed the loop modelling algorithm Sphinx, which is capable of generating high-quality decoy sets that are enriched with near-native conformations, and in addition, by including the ranking procedure described in this Chapter, produces significantly more accurate predictions than can be achieved using a straightforward *ab initio* algorithm.

The H3-specific version of our algorithm, Sphinx-H3, produces promising results with the inclusion of the ranking protocol, achieving comparable or better results than some of the leading H3-specific *ab initio* algorithms (Teplyakov *et al.*, 2014). The kinked/extended conformation of the H3 targets was correctly predicted in the majority of cases — even though no prior assumptions or predictions were made about the loops, and no restraints were imposed on the decoys to force a kinked/extended structure. Sphinx is also fast — for example, in comparison to RosettaAntibody, Sphinx gives increased or similar accuracy in under 3% of the computation time. Sphinx is therefore a more viable option for use in applications such as structure-based virtual screening.

The results achieved by our method indicate that fragments that are a slightly different length to the target loop are indeed a valuable source of structural information. Using a hybrid approach allows Sphinx to use the structural data that is available without being restricted to only returning loop conformations that have previously been observed. Also, since we have demonstrated that this approach can be applied successfully to prediction of general protein loop types as well as the difficult H3 loop, we believe that the algorithm can be easily adapted for other specific protein loop types and should continue to perform well. For example, it could be applied to membrane protein loops, or the other five antibody CDR loops, especially since the work into different length loops with the same structure was based on their canonical forms (Nowak *et al.*, 2016).

From the results presented here in this chapter, we believe that in a real loop modelling situation a knowledge-based method such as FREAD should be tried first (since such methods are usually very fast). However, if a suitable fragment cannot be found, then the next approach should be to use a hybrid algorithm such as Sphinx. Instead of reverting to a pure *ab initio* algorithm that ignores previously observed structures, this allows more of the available structural data to be used, leading to more accurate predictions.

As the results presented here demonstrate, by using the extra structural information contained within different length loops, Sphinx produces high-quality decoy sets and accurate predictions in a fraction of the time of some existing algorithms. As the PDB expands and more structural data becomes available, Sphinx's performance should improve further.

Pre-Chapter Image: A kinase enzyme bound to ATP — this protein's function depends on the movement of a long loop in its structure.

6 Loop Flexibility

Contents

6.1 Introduction

Experimentally determined protein structures, such as those solved using X-ray crystallography, can give a false impression of rigidity. Proteins can therefore often be treated as static objects whose structures do not vary. The standard in the protein

structure prediction community is to create predictions that match the static X-ray structure (Moult *et al.*, 1995; Teplyakov *et al.*, 2014). We have therefore used this approach throughout the work presented in this thesis so far; when predicting loop structures we have treated the X-ray determined structure as the 'correct' conformation, and measured our predictive accuracy based on that one structure. However, in reality, proteins are not static — they are constantly in motion. Consequently, there may be a set of conformations that a protein is able to assume, instead of just one. The structure that results from an X-ray crystallography experiment is an average of the structures adopted by each protein molecule across all unit cells, and is therefore weighted towards the most stable structure(s) in that environment.

Far from being just a complication that should be accounted for in protein research, flexibility is often an important component of protein function. In fact, it has been proposed that the diverse range of functions displayed by proteins is itself a result of conformational diversity and flexibility — a protein that can perform different functions, because it is present in a number of different conformations *in vivo*, may be the starting point for the evolution of new proteins that become specialised in their specific functions (James and Tawfik, 2003). This may be a factor in the affinity maturation of antibodies. The immune system must have antibodies available that bind to a huge range of possible antigens, however the sequence diversity is limited to what is achievable from an organism's DNA (*i.e.* without somatic hypermutation). If the antigen-binding sites of germline antibodies were flexible, however, it would be possible for them to bind to multiple epitopes (polyspecificity), allowing the immune system to protect against many more potentially harmful substances. Flexibility can then be reduced and binding affinity and specificity improved during the maturation process. There is some experimental evidence for this (Sethi *et al.*, 2006), and the same result has been obtained through computational studies (Babor and Kortemme, 2009; Willis *et al.*, 2013).

The size of movements that can occur within a protein vary enormously; from tiny vibrations, to conformational changes of residue sidechains, to local backbone movements, to motions of complete domains. In accordance with the work presented in the rest of this thesis, in this chapter we again focus on protein loops. The ability of

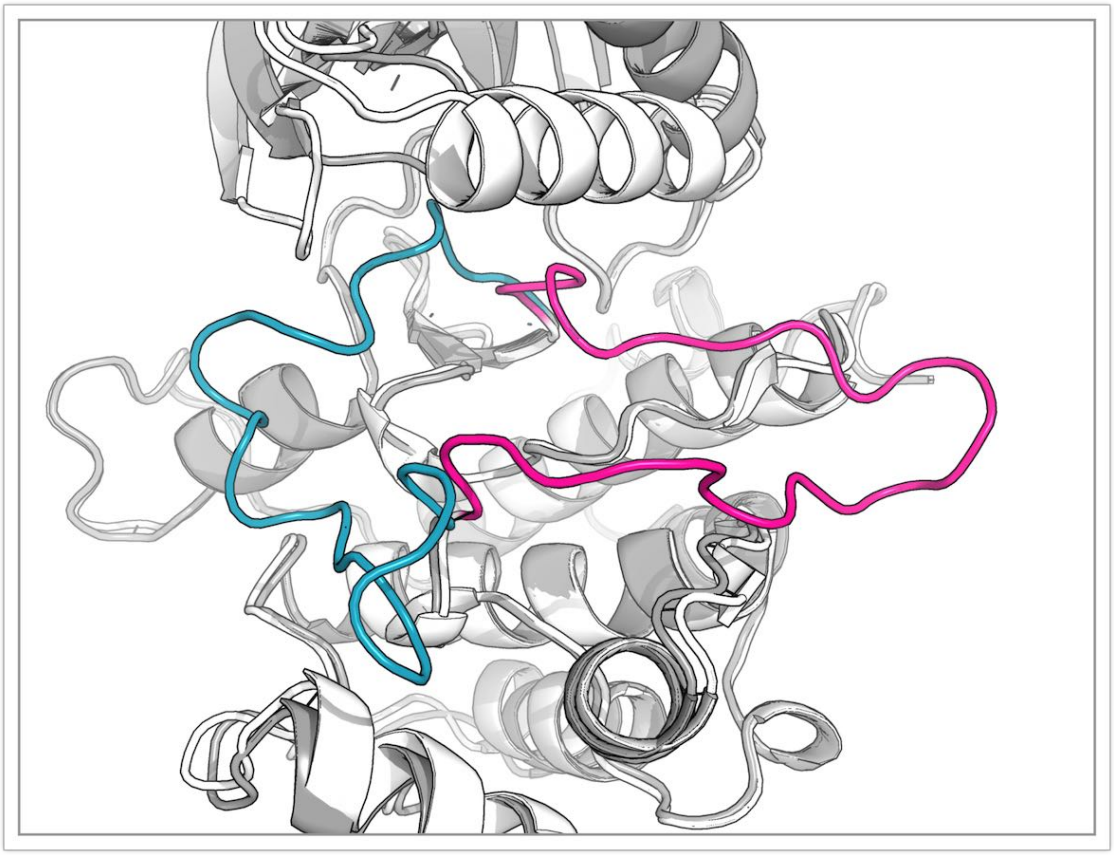

Figure 6.1: The activation loop of a protein kinase, in both inactive (blue, PDB entry 1IRK) and active (pink, PDB code 1IR3) conformations. Phosphorylation of the loop causes a shift from the inactive to the active form, allowing ATP to bind.

loops to adopt different conformations is often vital for protein function. A key example is the activation loop found in protein kinases — this 22-residue loop changes its shape considerably to switch between the active and inactive forms of the protein (Figure 6.1). Phosphorylation of the loop causes a conformational change, making the catalytic site of the enzyme accessible by its substrate, ATP (Adams, 2003). Other proteins whose biological function depends on a change in loop structure include triosephosphate isomerase (Derreumaux and Schlick, 1998) and lactate dehydrogenase (Qiu *et al.*, 2007).

Here we describe our investigations into loop flexibility, and how it relates to the problem of structure prediction. This is of particular interest when designing antibodies, since the CDRs often change their structure upon binding to an antigen — it is therefore necessary to understand this to accurately predict and make changes to the antibody's behaviour. To begin with, we have investigated the flexibility of loops from all proteins.

6.2 Preliminary Data Exploration

6.2.1 Same Loop, Multiple Structures

Our approach to studying loop flexibility is to look at proteins with multiple structures deposited in the PDB. Conformational changes of proteins have previously been investigated in this way (Chothia and Lesk, 1986; Kosloff and Kolodny, 2008), and databases of movements have been created (Echols *et al.*, 2003; Monzon *et al.*, 2013; Amemiya *et al.*, 2012; Juritz *et al.*, 2011), but to our knowledge these analyses have not been performed specifically for loop regions. By comparing the conformation of the same loop found in multiple PDB structures, we can see how much conformational variability is observed and hence infer how flexible a loop is. For example, if a particular loop is found in the same conformation in each protein structure, this loop can be assumed to be inflexible — other conformations are likely to be less favourable and are therefore not observed. Conversely, if a loop is seen adopting several different conformations, this is evidence that the loop is flexible.

Using PISCES (Wang and Dunbrack, 2003), we extracted from the PDB all structures solved using X-ray crystallography, with resolution below 2 Å, and placed them into sets of identical sequence. Any proteins that did not have any sequence-identical matches were ignored. The secondary structure of the PDB structures in each set was then assigned using DSSP (Joosten *et al.*, 2011), and the loop regions extracted. We considered a loop to be residues that connect helices or strands (assigned the labels H, I, G and E by DSSP) that are at least three residues in length. Where the exact definitions of loop regions varied between structures within a set, we extend the loop definition to include all possible loop residues. For example, if the loop of one structure in a set is defined to cover residues 10 to 16, but DSSP places the same loop in the other member of the set between residues 11 to 17, our definition of the loop is residues 10 to 17.

Figure 6.2 shows how many sets of loops the above procedure results in, for different loop lengths between 3 and 25 residues. In total, there are 65,697 loops for which we have multiple structures, the majority of which are short loops, with a peak at four residues (with 11,656 loop sets). This is not surprising, since in the whole PDB,

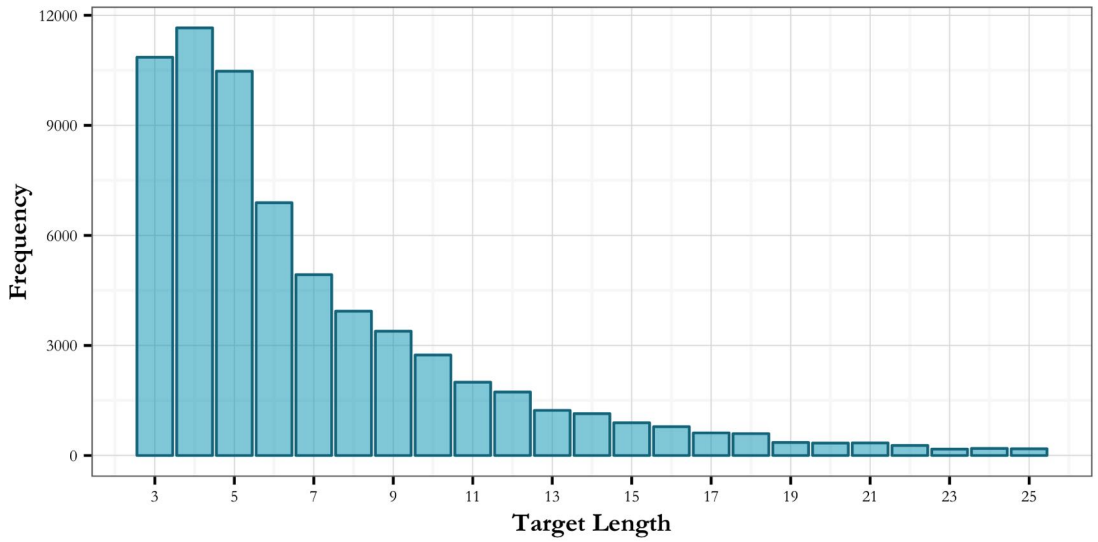

Figure 6.2: The number of loops with multiple structures in the PDB, for each loop length between 3 and 25 residues. Overall there are 65,697 loops with multiple structures, the majority of which are short loops.

short loops far outnumber long loops. However, there is still enough data available for longer loops (there are at least 180 loop sets for all lengths) that we are not limited to investigating short loops.

We then investigated how much structural variation is displayed by these loop sets, by calculating the backbone RMSD between every pair of loops in a set after superimposing the anchor residues (the two residues on either side of the loop). We consider the maximum pairwise RMSD for a loop set, since this is a measure of how much variation is observed. A density plot showing the distribution of maximum pairwise RMSDs of each loop set is shown in Figure 6.3. Most loops actually show very little variation between structures, with each loop length's peak occurring at less than 0.25 Å. Conformational variability increases with loop length, with longer loops having a greater number of sets with large maximum pairwise RMSDs — this is demonstrated most clearly by the decreasing height and movement along the x-axis of the peak in the distributions.

An alternative view of this data is displayed in Figure 6.4. This shows, for each loop length between 3 and 25 residues, how many sets of multiple loop structures there are with a maximum pairwise RMSD over a cutoff value. In general, there are fewer loop sets at each cutoff for longer loop lengths. However, since the majority of

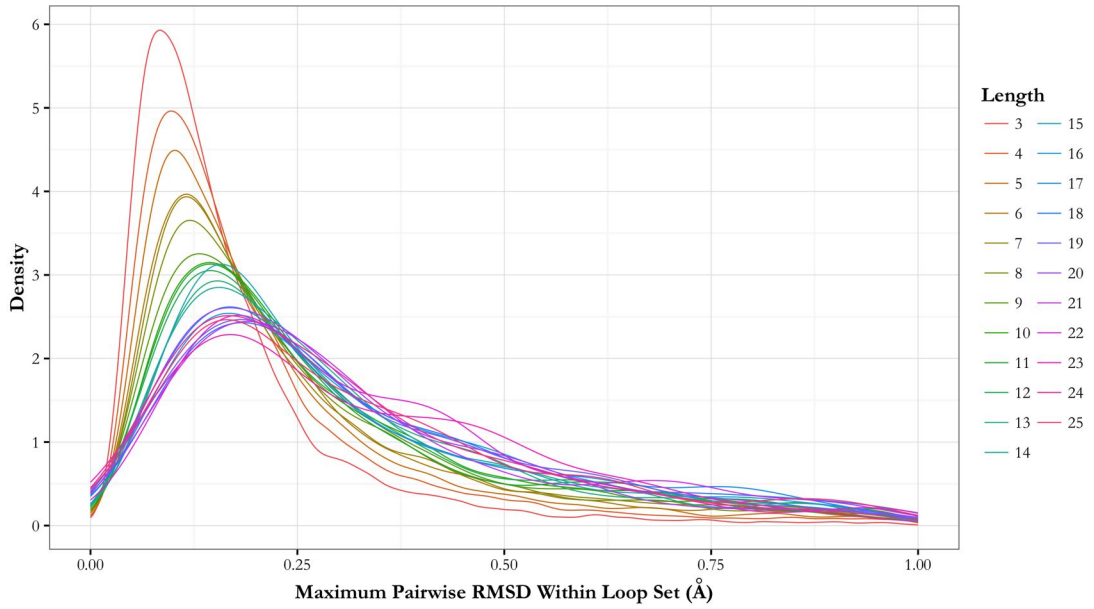

Figure 6.3: The distribution of maximum pairwise RMSDs, grouped by loop length. Most loop sets show variation of around 0.1-0.2 Å, and the amount of variation increases with loop length. This figure is a section of the whole plot, cropped to show maximum pairwise RMSDs up to 1 Å for clarity.

short loops have low maximum pairwise RMSDs, there are fewer examples of these loops that show high conformational variation. There are only 1,028 loop sets that have a maximum pairwise RMSD over 2 Å, which is only 1.6% of the total number of loops with multiple structures.

Figure 6.5 shows two examples of loops that show different amounts of conformational variation in their experimentally-determined structures. There are four structures of a particular pyranose 2-oxidase enzyme in the PDB. A 14-residue loop of this protein displays high variation — two of the structures have similar conformations (coloured yellow and purple in Figure 6.5), while the remaining two conformations are quite different. The maximum pairwise RMSD for this set of loop structures is 5.96 Å. This loop corresponds to the 'gating segment' identified by other researchers (Spadiut *et al.*, 2010), which is thought to control access to the active site. A 15-residue loop in the protein serine hydroxymethyltransferase, on the other hand, shows barely any variation in structure at all — the maximum pairwise RMSD in this case is 0.11 Å. To our knowledge, this loop does play a major role in the protein's function.

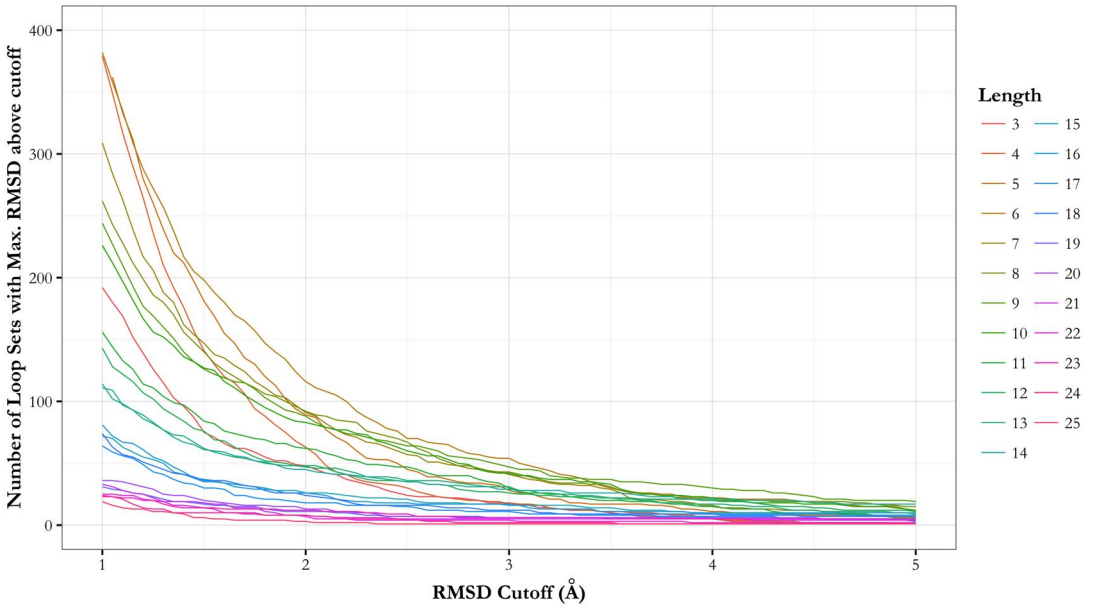

Figure 6.4: The number of loop sets with a maximum pairwise RMSD above a cutoff, for each loop length between 3 and 25 residues. There are fewer examples of loops displaying high variation for long lengths.

6.2.2 Anchor Secondary Structure and Conformational Diversity

To see whether loop flexibility is related to the secondary structure of the anchor residues, we split the loop sets into four groups: EE, EH, HE, and HH, where the first and second letters represent the secondary structure of the N- and C-anchor of the loop, respectively. 'H' is the DSSP notation for an α-helix, and 'E' corresponds to a β-strand. To stop the length of the loops in each set affecting results (*i.e.* if long loops are overrepresented in one category, it could be falsely assumed that they are more flexible), we selected the same number of sets from each loop length at random. In order to use as much of the data as we can, we selected the highest number possible — *i.e.* the number of sets in the smallest category at each length. We then compared the distribution of maximum pairwise RMSDs for each category. Our results indicate that loops between β-strands (EE) may be slightly more flexible than loops in the other categories (Figure 6.6). Loops between helices (HH) are the next most flexible, followed by the types strand-loop-helix (EH) and helix-loop-strand (HE).

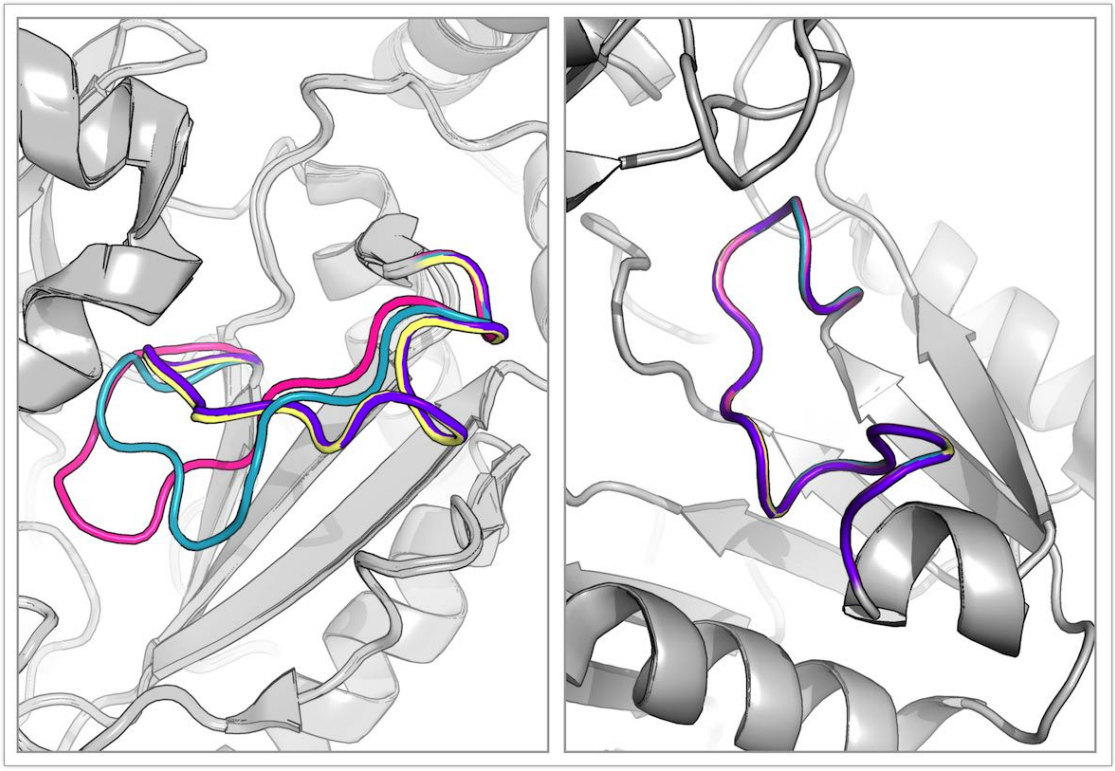

Figure 6.5: Examples of loops showing high (left) and low (right) structural variation. On the left are four different structures found in the PDB of a 14-residue loop found in the pyranose 2-oxidase enzyme (residues 451-464). PDB structures shown are 3PL8 (blue), 3LSM (pink), 2IGN (yellow) and 2IGO (purple). The maximum RMSD calculated between these four structures is 5.96 Å. On the right is a loop that displays very little structural diversity — a 15-residue loop of the serine hydroxymethyltransferase enzyme (residues 116-130). PDB structures shown are 2VMP (blue), 2VMO (pink), 2VMQ (yellow) and 2VMN (purple), and the maximum pairwise RMSD is 0.11 Å.

6.2.3 Structure Determination Method

So far we have only considered structures that were determined using X-ray crystallography. To see if structures solved using different methods produce the same results, we performed the analysis for NMR data. As before, we used DSSP to define loop regions and calculated RMSDs between loop structures after first aligning the anchor residues, but this time the members of each set of identical loops are simply the ensemble of models found within each PDB file. These models represent structures that are possible given the restraints found during the NMR experiment. The results of this analysis are shown in Figure 6.7, along with the results from a set of protein loops solved using X-ray crystallography (a randomly-selected set with the same number of loops and the same distribution of loop lengths) for comparison. There is a significant difference

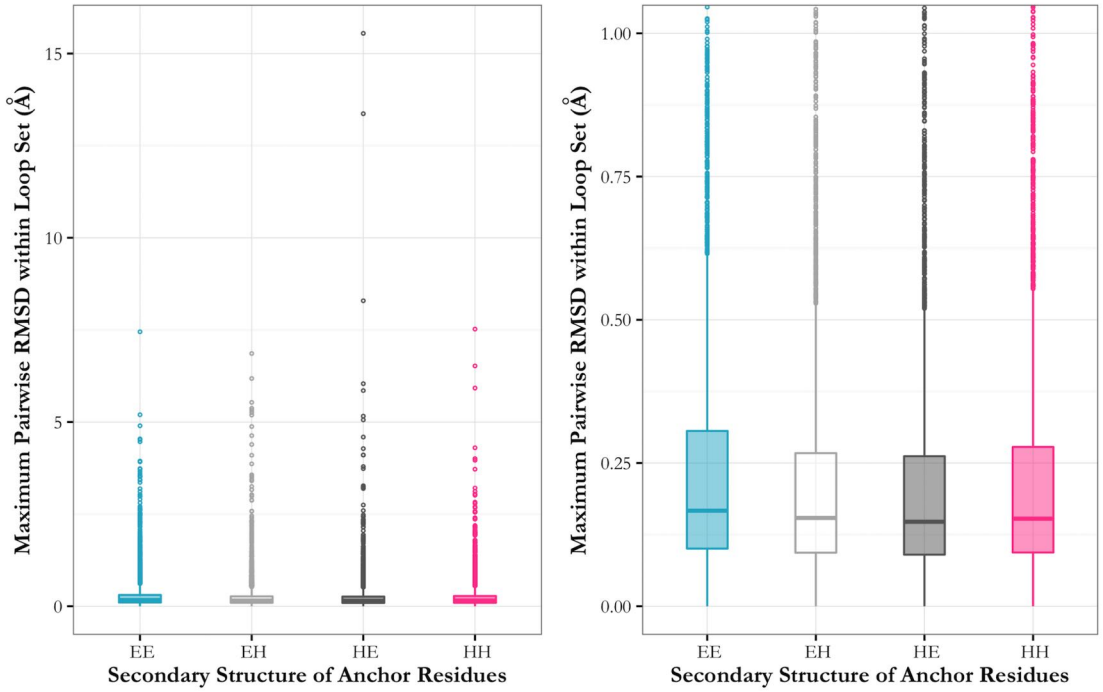

Figure 6.6: The relationship between anchor secondary structure and the variability of loop structures. 'EE', 'EH', 'HE' and 'HH' represent loops in the following structural environment: strand-loop-strand, strand-loop-helix, helix-loop-strand, helix-loop-helix. The plot on the right is a 'zoomed-in' version of the one on the left, allowing the differences to be seen more clearly. Loops between β-strands are slightly more flexible than those located between other secondary structure types.

between the two structure determination methods; loop structures solved with NMR spectroscopy display far more conformational variation than those determined with X-ray crystallography. The distribution of maximum pairwise RMSDs for NMR data is far broader, and peaks at around 0.8 Å while the X-ray data has a sharp peak at approximately 0.2 Å. There is a significant number of NMR structures in which the RMSD between the models is zero — it is difficult to tell whether this is due to there being high confidence in the loop conformation, or whether this is an artefact of how the structures were deduced. The majority of loop structures solved using NMR have maximum pairwise RMSDs of over 1 Å (63.4%); this is true for only a small fraction of the X-ray determined structures (3.7%). This raises some interesting questions: does X-ray crystallography, by forcing the proteins into an ordered arrangement, prevent us from observing flexibility that would otherwise be present? Or does NMR data overestimate flexibility? Interpreting the possible structures that are obtained from

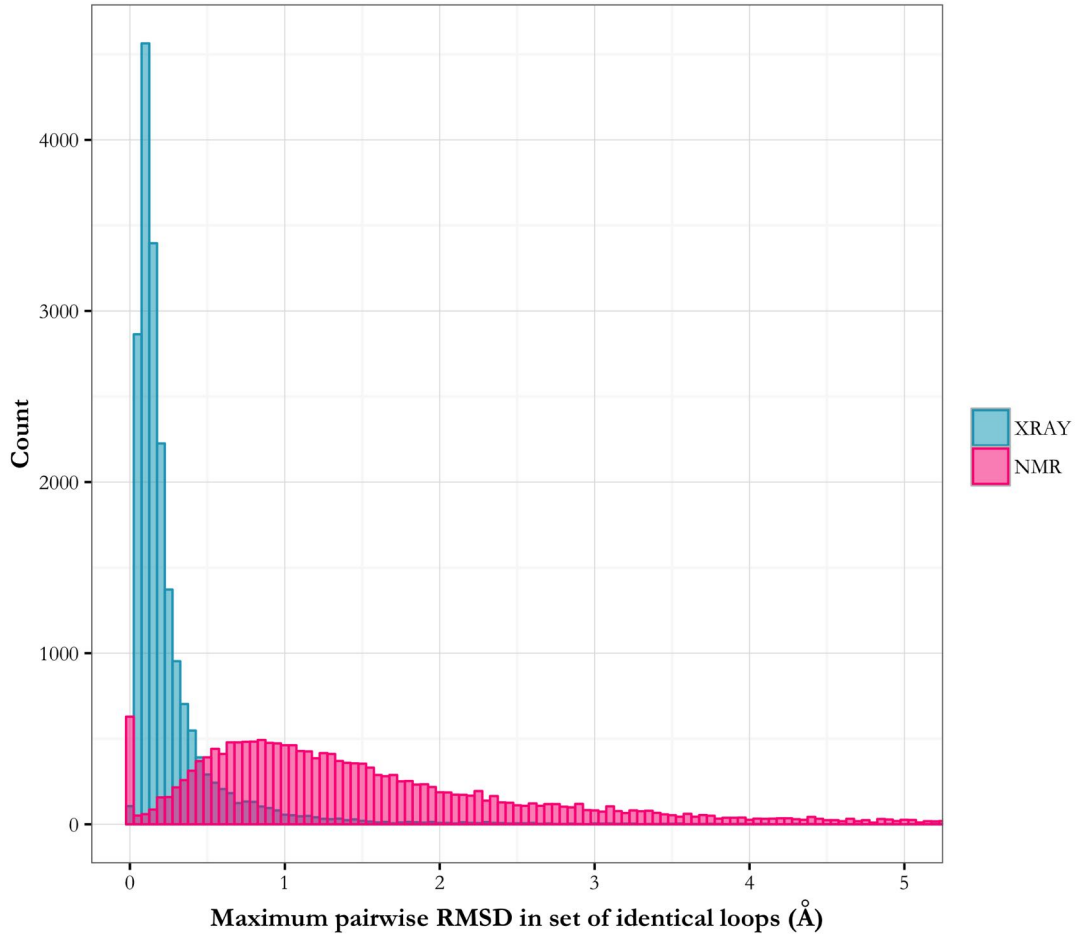

Figure 6.7: The conformational variation of loop structures for which we have multiple structures, determined by either X-ray crystallography and NMR spectroscopy. The two sets contain the same number of loops and have the same length distribution. NMR structures are more conformationally diverse than X-ray structures.

the identified restraints as true conformations that can be adopted by the protein may be fallacy; instead there may be other experimental factors that result in a lack of structural information.

6.2.4 Antibody CDRs

Since the main focus of our work is antibodies, we performed the same analysis on CDR loops, as defined by Chothia (Chothia and Lesk, 1987). Since the amount of structural data available for antibodies is limited, we were unable to implement a resolution cutoff as severe as 2 Å; instead we used a cutoff of 3 Å. At this resolution, there are 512 antibody chains with multiple structures, giving 256, 259, 260, 242, 243, and 228 loops for L1,

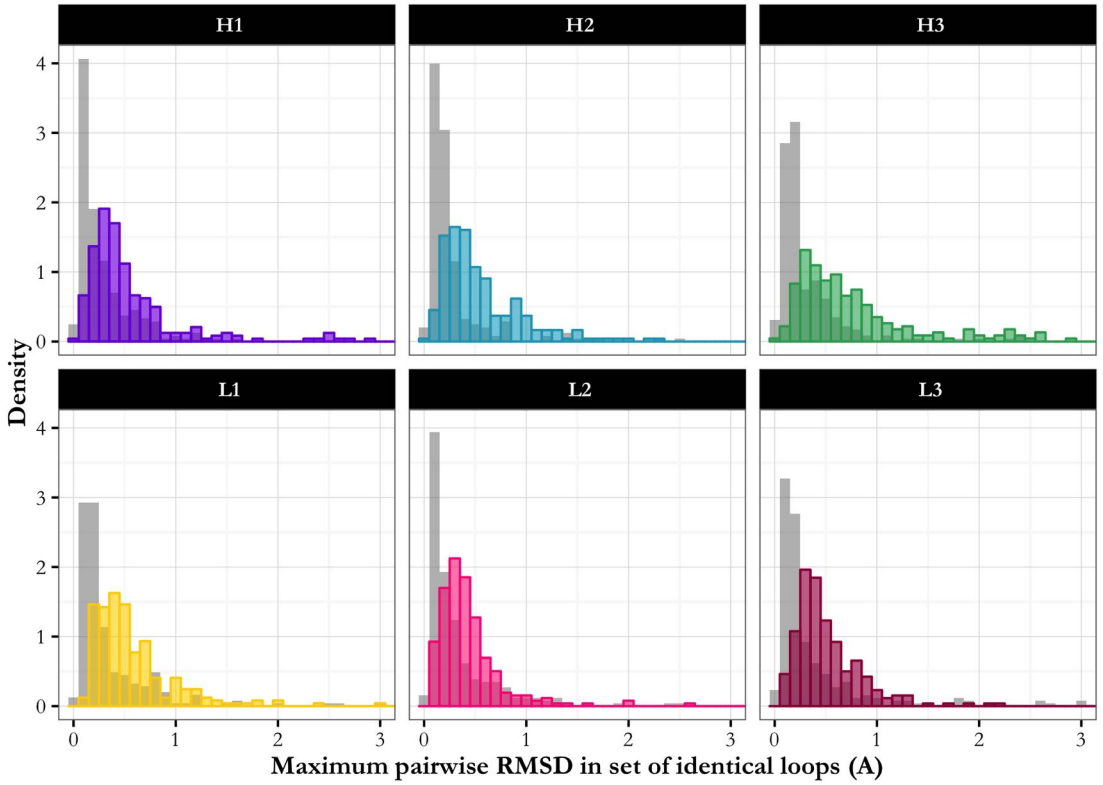

Figure 6.8: Structural variation in antibody CDR loops. For each antibody chain in the PDB (X-ray structures only) for which sequence-identical structures can be found, we calculated the RMSD between the different CDR loop structures, using Chothia numbering (Chothia and Lesk, 1987). In grey, we show the equivalent distribution for a set of general protein loops with the same length distribution. CDR loops show greater conformational variability than general loop types.

L2, L3, H1, H2 and H3 respectively. For each of the six CDRs, we selected a set of loops from non-specific protein types to compare to — we ensured that the loop length distribution was the same as the CDR in question (in the same way as that used in the anchor secondary structure investigation in Section 6.2.2), and only selected from loops located between β-strands (since CDR loops themselves are found in this structural environment). The comparisons between antibody and general protein loop structure variation are shown in Figure 6.8. For each CDR, the conformational variation shown by antibodies is greater than that displayed by general protein types. The most extreme case is the H3 loop, where 25% of the loop sets have a maximum pairwise RMSD of over 1 Å; only 6% of the general loop sets show structural deviation over this value. The large differences displayed by antibody CDR loops may be due to the use of a more lax resolution cutoff; the fact that there will tend to be more comparisons between bound

and unbound antibody structures; or differences in the light/heavy chain that the chain in question is bound to (as in this study we only considered one chain at a time).

6.3 Loop Prediction

6.3.1 High Variation vs Low Variation Loop Targets

When modelling loops, in general the aim is to produce a single conformation that closely resembles the native structure. The quality of a prediction algorithm is assessed by calculating the distance of a predicted structure from that of a single, experimentally-determined structure. This means that the ability of some loops to adopt different conformations is not considered. It may be that the loop model selected as the final prediction, though not identical to a particular experimental structure, is not incorrect: the decoy selection method may preferentially choose a different but equally valid conformation. In addition, there might be a mixture of different valid conformations present in the set of top-ranked decoys.

To see if decoys that are structurally close to different conformations are produced using our loop prediction methodology, we ran Sphinx on two target sets: one containing loops that are found in different conformations in the PDB, and one with loops whose structures do not vary between multiple PDB structures. From the sets of loops that we have found with multiple structures determined using X-ray crystallography in the PDB (see Section 6.2), we extracted two groups dependent on the maximum pairwise RMSD calculated for each set — one with sets whose maximum pairwise RMSD is below 0.3 Å (*i.e.* displaying low structural variation), and another containing sets with a maximum pairwise RMSD of above 2 Å (high variation). From these two groups, we selected five loop sets at random for each loop length between 10 and 16 residues, inclusive. We therefore had two target sets of 30 loops, each with the same length distribution. We ran the general protein version of the Sphinx algorithm on one randomly-chosen protein structure from each target loop set, using the general loop database detailed in Chapter 4, adding sidechains to the top 500 decoys using SCWRL (Krivov *et al.*, 2009) and using

dDFIRE (Yang and Zhou, 2008) to score and rank the decoys. Structural information from proteins identical in sequence to the target loop were ignored.

Our results are shown in Figure 6.9. For each target we have presented the RMSD of the best decoy in the top 500 as ranked by our knowledge-based scoring function, along with the RMSDs of the five top-ranked decoys, when compared to each conformation of that loop that was found in the PDB. For the 'Low Variation' target set, therefore, there is only one RMSD displayed per decoy, since there is only one conformation that has been observed for that loop. The number of conformations present for the 'High Variation' set ranges from two to four. Loop structures within each target loop set were separated into different conformations using the UPGMA clustering algorithm, with an RMSD cutoff of 1 Å (see Chapter 5). Where multiple loops belong to the same conformation, the RMSD reported is an average of the RMSDs between the decoy and each conformation.

Imagine shooting an arrow in a random direction and hoping to hit a target; if the number of targets is increased, there is a greater chance of hitting one. Intuitively, therefore, one may think that loops with different conformations would be easier to predict, since there are multiple correct answers. However, this is not what we observe. It appears that accurately predicting loops with high conformational variation is much more difficult than predicting loops that only adopt a single structure. The average RMSD of the best decoy in the top 500 (considering the RMSD to the nearest conformation for each decoy) for the high variation set is 1.04 Å; for low variation loops it is 0.80 Å. In addition, for twenty of the loop targets with low variation, a decoy exists in the top 500 that has an RMSD to native structure of less than 0.5 Å. This is only true for five of the high variation loops.

When the the final ranking step is performed, the difference becomes even more apparent. The average RMSD of the top-ranked decoy (considering the lowest RMSD value across all conformations) is 3.42 Å (median of 2.77 Å) for high variation loops and 2.01 Å (median 0.58 Å) for low variation loops. For loops with multiple conformations, Sphinx could produce a prediction within 1 Å of a native conformation in only 9 cases, while for low variation loops 20 sub-Ångström predictions were made. Considering

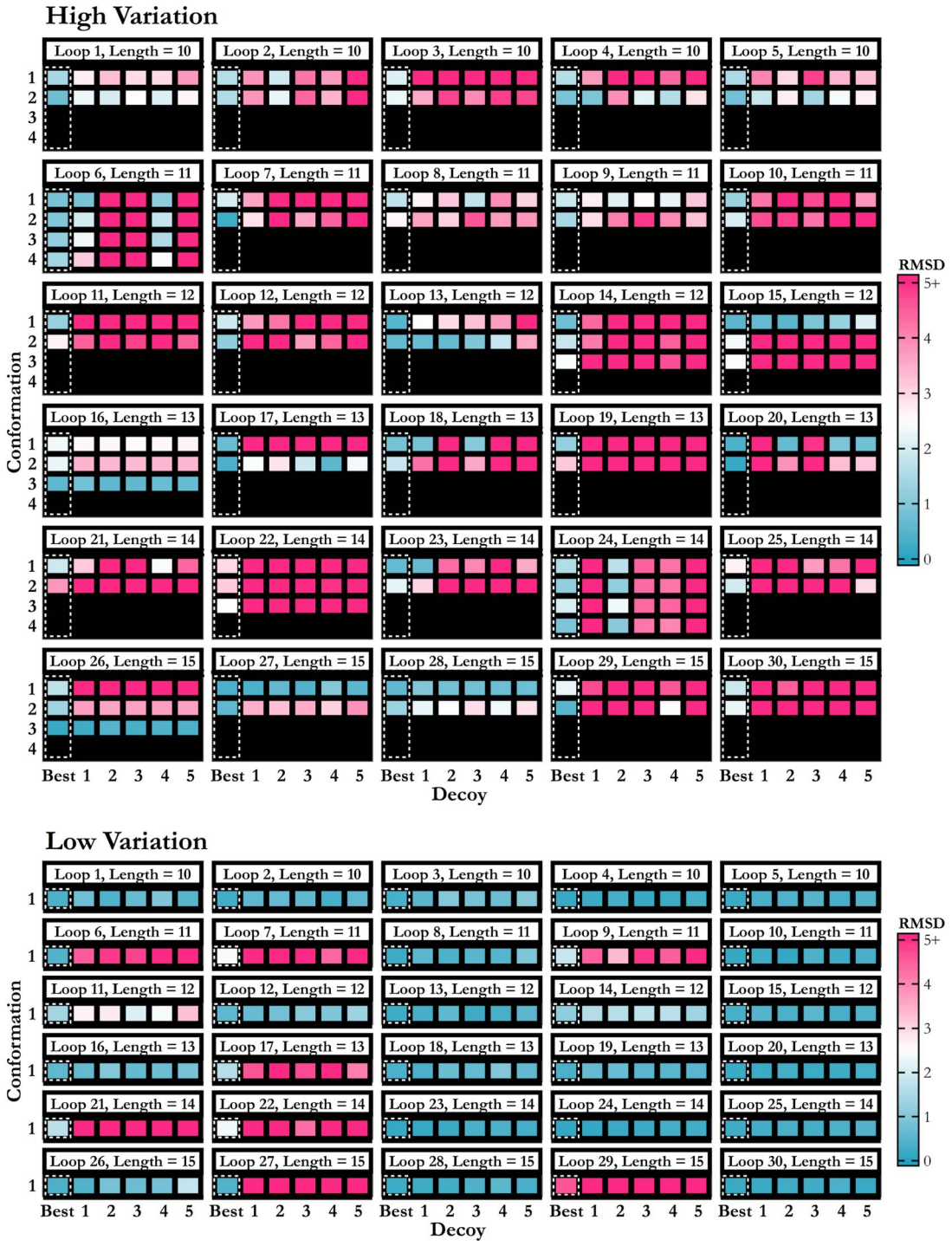

Figure 6.9: Sphinx prediction results for the high and low variation target sets. For each target (each box headed by a white header bar), the different rows correspond to each of the conformations observed for each loop, and the coloured cells in each row represent the the RMSD of a decoy (either the best of the top 500, outlined with a dashed white box, or the five top-ranked decoys). Where more than one structure exists for a conformation, the value indicated is the average RMSD across all structures. Loops with multiple conformations are predicted more poorly than those with only one.

the top five decoys, and choosing the RMSD to the nearest conformation, the average RMSDs for high and low variation loops are 2.39 Å and 1.76 Å respectively.

For highly variable loops, on occasion the decoys selected by the ranking system are close to one of the conformations (for example loop set 26, where all the decoys in the top five are close in structure to conformation 3), however the majority of decoys selected are not close to any experimentally observed conformation. This implies that it would be very difficult to use Sphinx to predict a small ensemble of structures that represent the conformational flexibility of a loop.

We examined the RMSD distributions at various stages of the algorithm to see if we could pinpoint where the difference in accuracy arises. Figure 6.10 shows the RMSD distribution of the decoy set generated by Sphinx at three different stages: after all decoys have been generated, after the selection of the top 500 decoys by the knowledge-based scoring function, and after the top 5 decoys have been selected by dDFIRE. RMSDs shown are the RMSDs to the nearest experimentally-observed conformation. At all stages, there are more near-native decoys in the sets for the low variation target loops, however the difference between the two increases as the selection of decoys becomes smaller. This indicates that it may be slightly easier to generate near-native decoys for loops with only one conformation, but the main difference is in the ranking stage. The scoring functions struggle to identify the good quality conformations for loops that have multiple conformations.

To see whether this observation is unique to Sphinx, we tested three other loop modelling algorithms on the same two target sets — FREAD (Deane and Blundell, 2001; Choi and Deane, 2010), Rosetta (Stein and Kortemme, 2013) and LEAP (Liang *et al.*, 2014). Loops were predicted onto the same member of each set as used for Sphinx. FREAD is a knowledge-based method (see Chapter 2), while LEAP and Rosetta are *ab initio* algorithms. For FREAD, we used the same loop database that we used when running Sphinx (the general loop database as described in Chapter 4), and used the default parameters. We generated 500 decoys using Rosetta and used the algorithm's in-built scoring system to rank them. As recommended by the authors, we ran LEAP

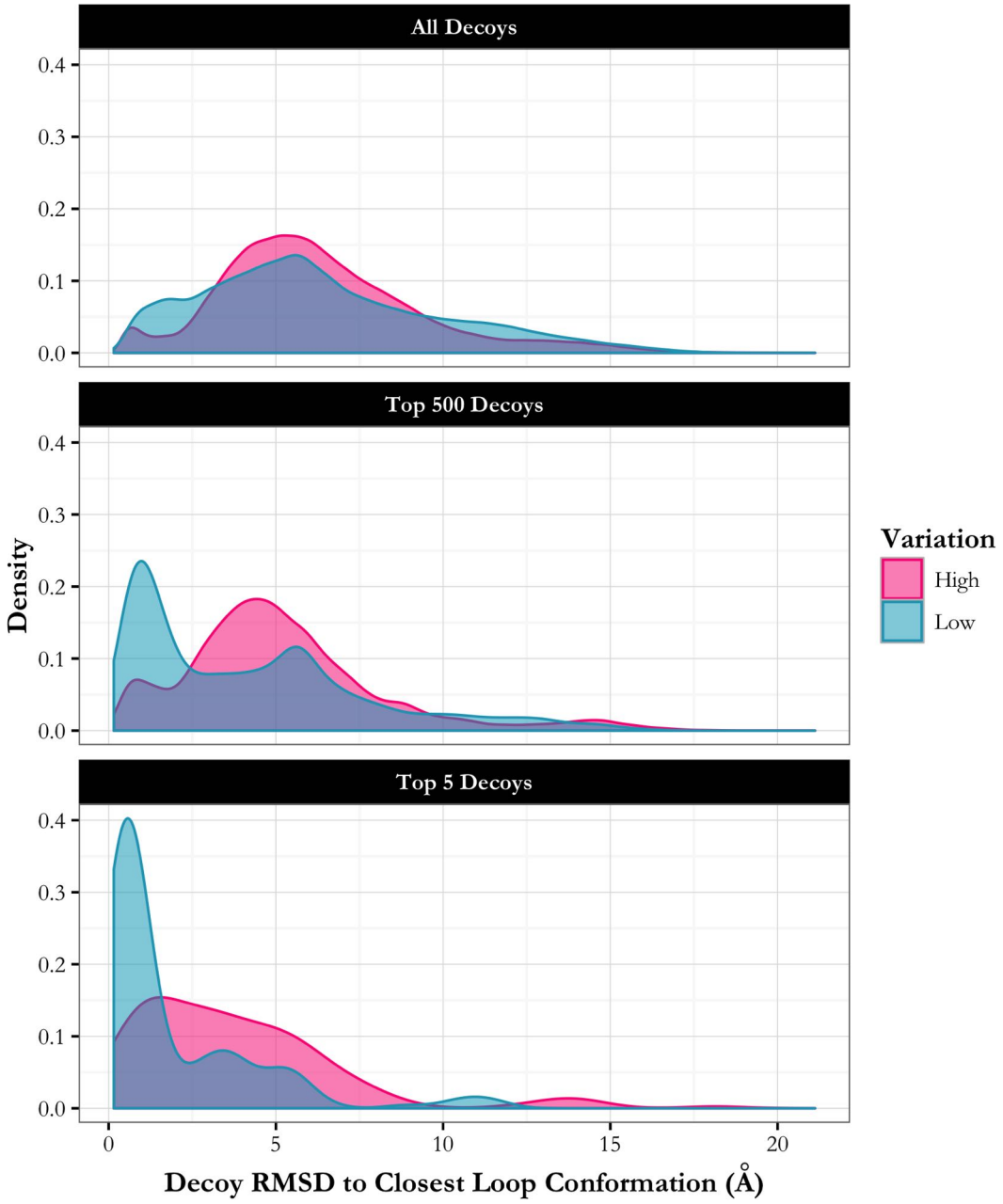

Figure 6.10: RMSD distributions at different stages of the Sphinx algorithm. The stages shown are after all decoys have been generated, after our own knowledge-based scoring function has been used to select the top 500, and after the top 5 decoys are selected by dDFIRE. RMSDs used are for the decoys generated for all targets in the high and low variation sets. As the algorithm progresses, the difference between the RMSD distributions becomes larger, with the decoy sets for low variation loops containing more near-native structures.

ten times to give ten different predictions, which we then ranked using their energies (as calculated by the LEAP algorithm).

The results for FREAD, LEAP and Rosetta are given in Figures 6.11, 6.12 and 6.13 respectively. FREAD produces a similar result to Sphinx. The coverage for the high variation set is lower than for the low variation set (56.7% compared to 86.7%). The average RMSD of the best decoy is 0.78 Å for the high variation set and 0.54 Å for the low variation set. After ranking the fragments that FREAD selects from the database by their anchor RMSD, the average RMSD of the top prediction is 1.00 Å for loops with multiple conformations, and 0.63 Å for loops with only one conformation. The best of the top five decoys has an average RMSD of 0.80 Å and 0.56 Å for the high and low variation sets, respectively. The differences in accuracy are not as large as they were when using Sphinx; this could be due to the low coverage of the high variation set skewing the results.

The results achieved by LEAP also show a similar trend, if not to the same extent. The best decoy produced for each target has an average RMSD of 2.09 Å for loops with only one conformation, and 2.30 Å for those that display conformational variability (again, considering the RMSD to the nearest native conformation). Decoys were made with sub-Ångström accuracy for nine targets in the low variation set, but only four in the high variation set. When ranked according to their calculated energies, the best decoy had an average RMSD of 3.00 Å for the low variation set and 3.66 Å for the high variation set, and the best of the top five had an average RMSD of 2.24 Å and 2.53 Å for the low and high variation sets respectively.

The results from Rosetta also show the same pattern. The best decoy produced had an average RMSD of 0.87 Å for the low variation set and 1.27 Å for the high variation set. For 15 targets in the high variation set, a decoy with an RMSD below 1 Å was generated, whereas decoys with sub-Ångström accuracy were created for 22 of the low variation targets. After ranking, however, the results for the two sets become more similar: the top prediction has an average RMSD of 5.14 Å and 5.07 Å, and an average best-of-top-five RMSD of 2.61 Å and 3.02 Å, for the low and high variation sets respectively. This could be because the Rosetta scoring function is not very accurate; the accuracy of Rosetta is worse than LEAP and Sphinx even though the best decoys in the set are generally close

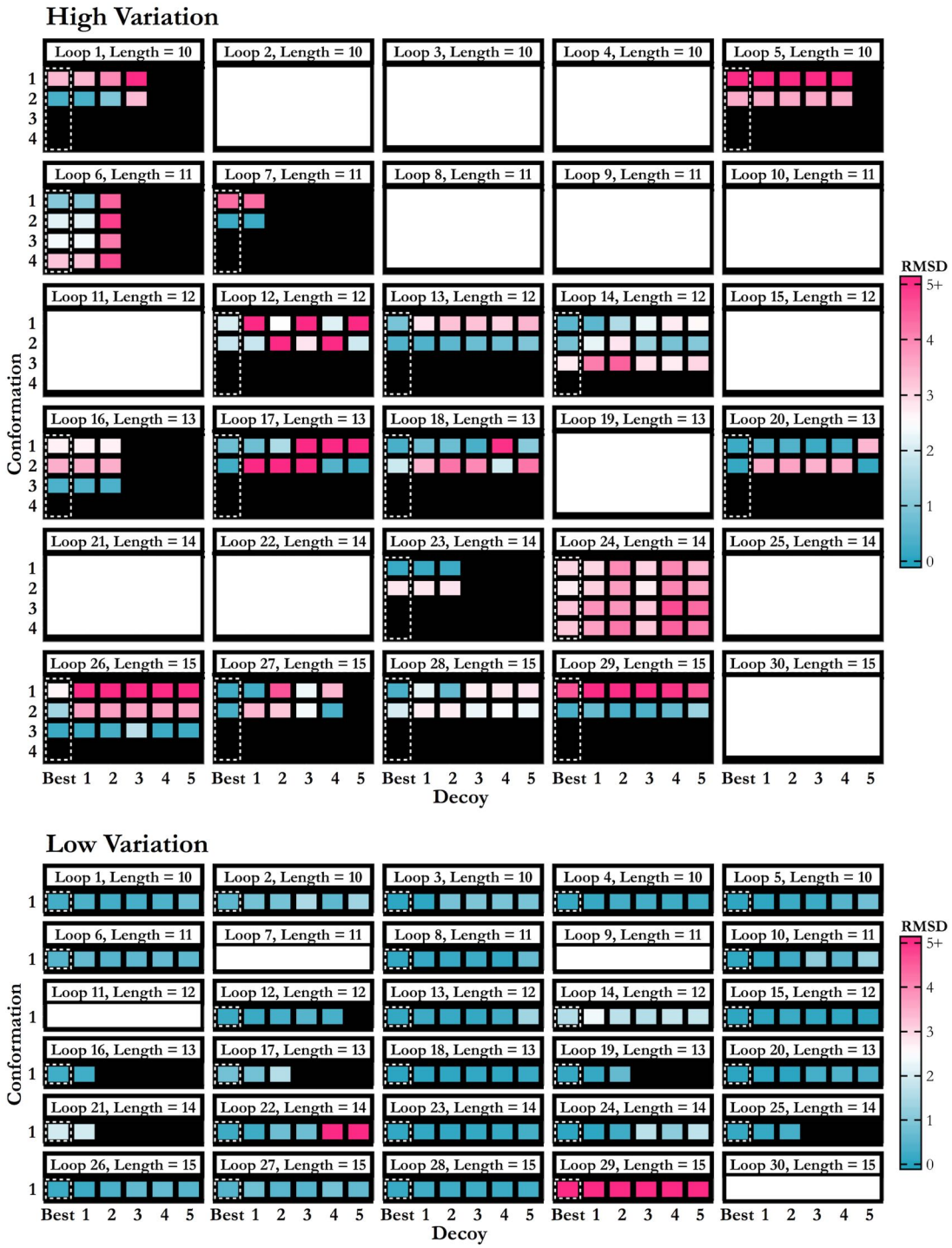

Figure 6.11: FREAD prediction results for the high and low variation target sets. For each target (each box headed by a white header bar), the different rows correspond to each of the conformations observed for each loop, and the coloured cells in each row represent the the RMSD of a decoy (either the best of the top 500, outlined with a dashed white box, or the five top-ranked decoys). Where more than one structure exists for a conformation, the value indicated is the average RMSD across all structures. White boxes indicate where FREAD was unable to make a prediction. Loops with multiple conformations are predicted more poorly than those with only one.

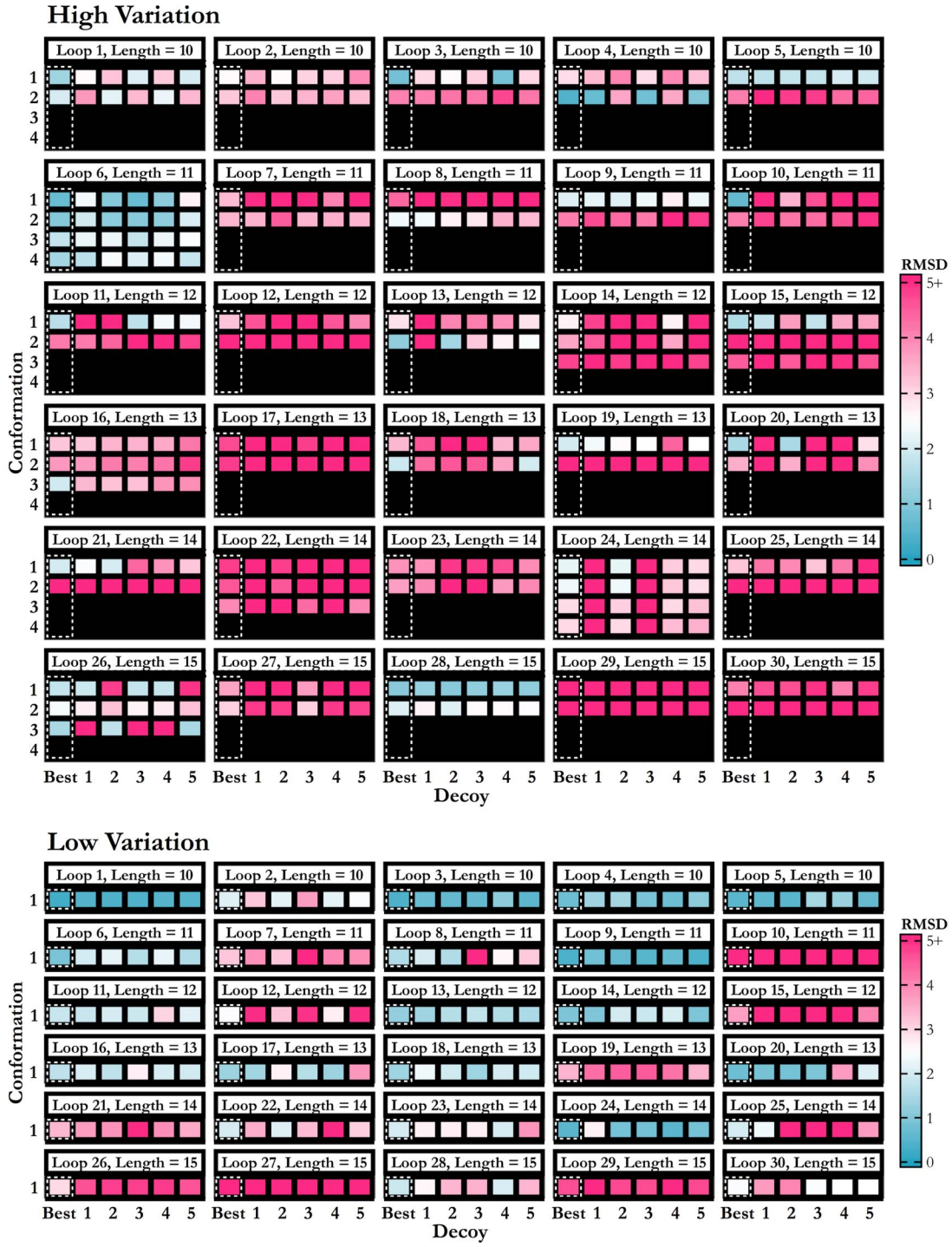

Figure 6.12: LEAP prediction results for the high and low variation target sets. For each target (each box headed by a white header bar), the different rows correspond to each of the conformations observed for each loop, and the coloured cells in each row represent the the RMSD of a decoy (either the best of the top 500, outlined with a dashed white box, or the five top-ranked decoys). Where more than one structure exists for a conformation, the value indicated is the average RMSD across all structures. As is the case for Sphinx and FREAD, loops with multiple conformations are predicted more poorly than those with only one.

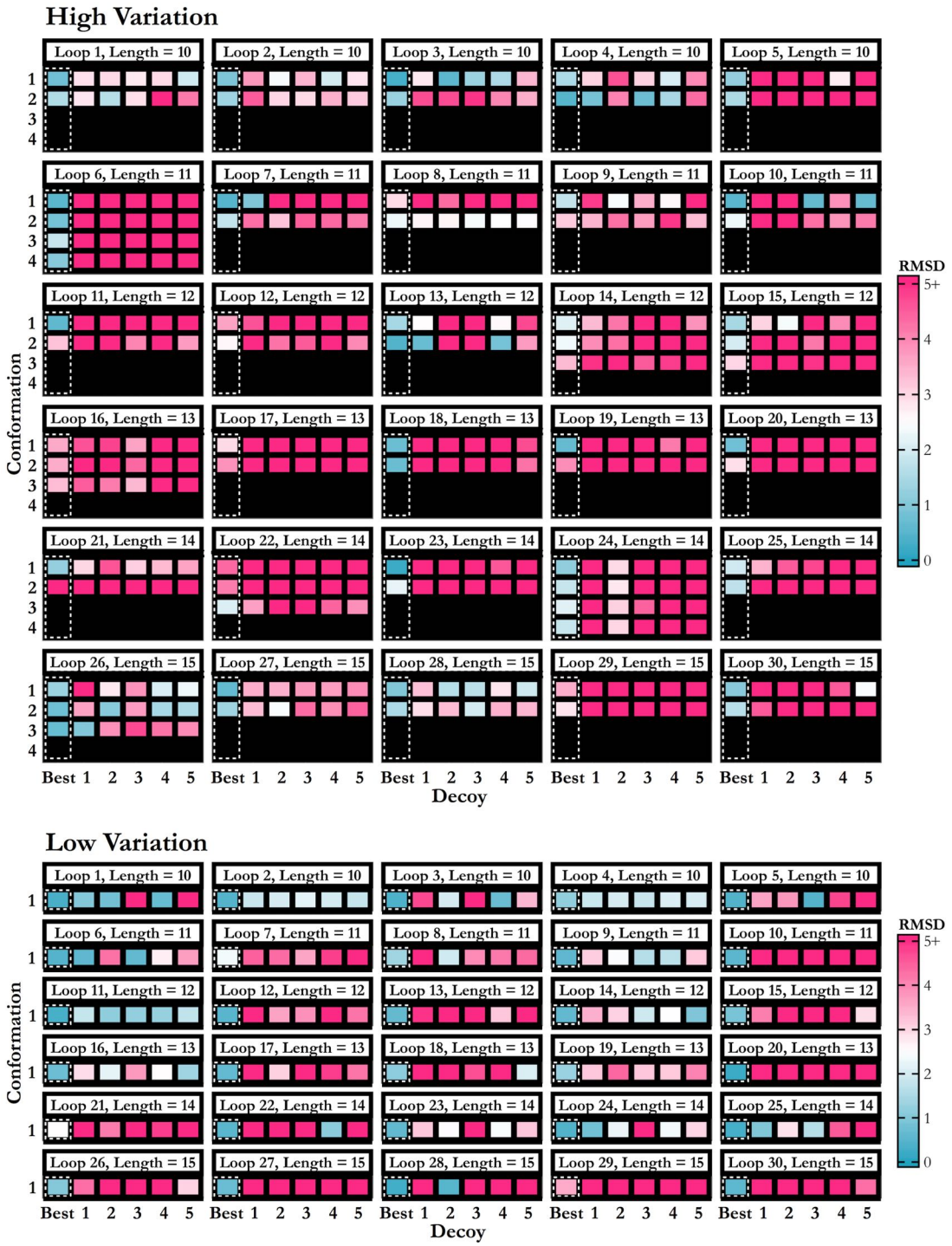

Figure 6.13: Rosetta prediction results for the high and low variation target sets. For each target (each box headed by a white header bar), the different rows correspond to each of the conformations observed for each loop, and the coloured cells in each row represent the the RMSD of a decoy (either the best of the top 500, outlined with a dashed white box, or the five top-ranked decoys). Where more than one structure exists for a conformation, the value indicated is the average RMSD across all structures. Like previous results have shown, decoys generated for loops with multiple conformations tend to be less accurate than those generated for single-conformation loops. However, results after ranking are approximately the same, possibly due to the overall inaccuracy of the scoring function.

to one of the native conformations. During the undertaking of the research presented in Chapter 5, we noticed that when a near-native decoy is present in the decoy set, in the majority of cases this decoy is ranked highly; however this does not occur here (*i.e.* for loop 10 of the low variation set, where the best decoy had an RMSD of 0.57 Å, the decoy ranked top by the Rosetta scoring function has an RMSD of 9.86 Å and the best of the top five decoys has an RMSD of 6.13 Å).

To see how the decoys generated for each target set are spread throughout the conformational space, we calculated the RMSDs between all pairs of decoys for each target (Figure 6.14). This should indicate how structurally alike the decoys are to one another. For each method, the decoys generated for the targets with only one conformation show less structural variability than those targets with multiple conformations, however like the prediction results the trend is more extreme for the decoys obtained using Sphinx. This trend could mean that for low variation targets, loop prediction algorithms are able to explore the conformational space more efficiently.

6.3.2 Scoring Differences

Prediction accuracy may be decreased for highly variable loops because of inaccuracies in the scoring functions. This is demonstrated by Figure 6.10, where the difference in RMSD distributions for the high and low variation sets becomes greater after each ranking step. We therefore decided to look at the final ranking stage in more detail. Figure 6.15 shows the spread in scores returned by dDFIRE for each loop in the two target sets used in the Sphinx prediction investigation (see Section 6.3.1). For loops with a single conformation, the spread in scores is much greater, and there is clearer separation between the 'good' and 'bad' decoys (this is most clearly seen for the target 3P7YA_117_129, where two 'clusters' of decoys have formed, separated by their scores). On the other hand, for high variation loops, the spread is smaller, and the good decoys are not clearly separated from the bad ones. This may be due to the energetic landscape of variable loops; as said in the introduction to this Chapter, sometimes several local energy minima may exist, and other conformations of the protein may not be as energetically

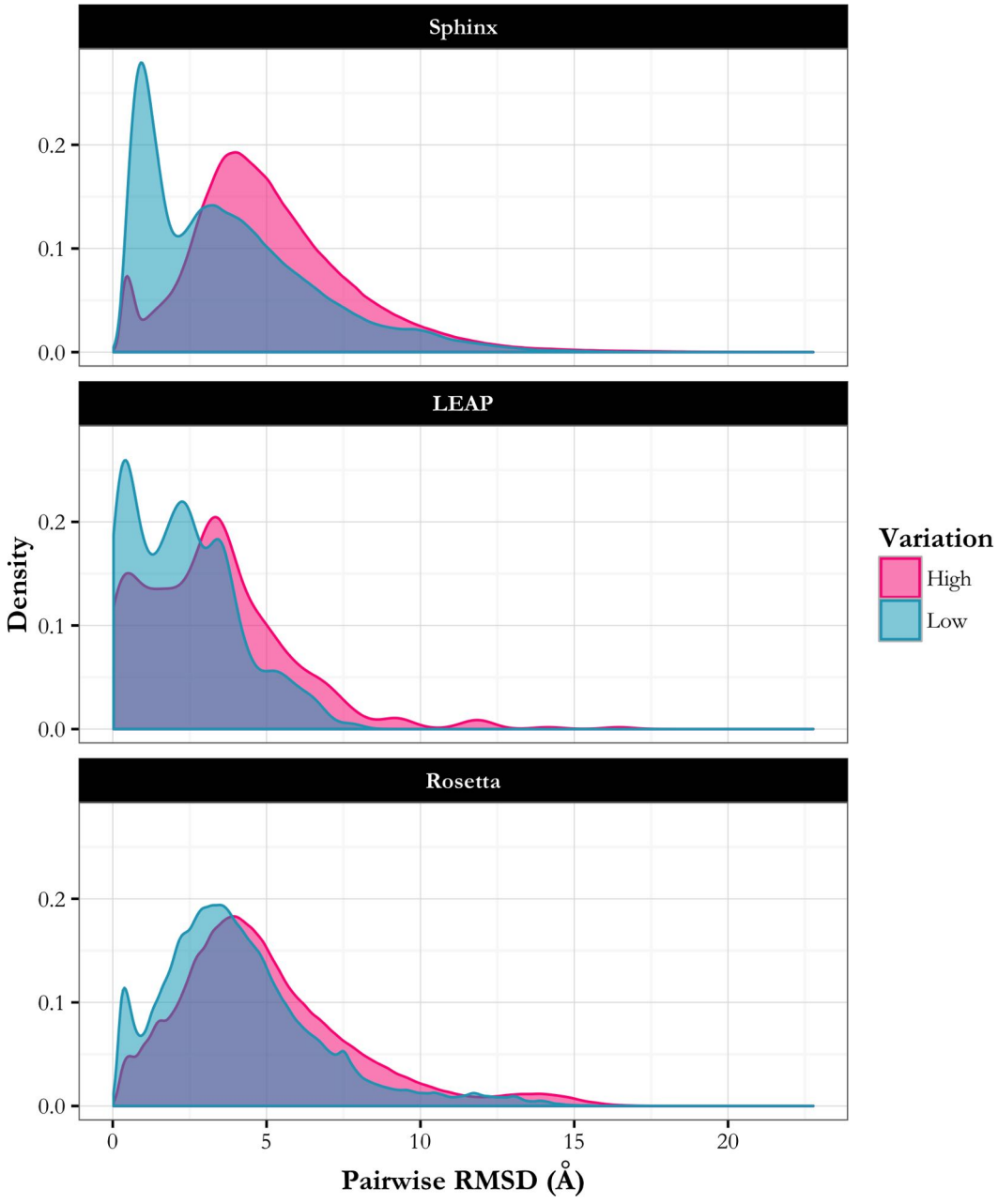

Figure 6.14: Pairwise RMSD distributions for the decoys generated by Sphinx (the top 500 decoys for each target), LEAP (ten decoys for each target) and Rosetta (500 decoys for each target). The distributions for the low variation target set are more skewed towards low RMSDs, indicating that the decoys generated for the low variation targets cover less of the conformational space.

unfavourable as a consequence. If all the conformations, good and bad, have similar energies, it is more likely that a ranking system will fail to select the lowest energy ones.

6.3.3 'Inflexible Loop' Target Set

In 2008, the flexibility of proteins was explored by using molecular dynamics simulations of a large set of proteins that cover a range of different folds (Benson and Daggett, 2008). During this study, a set of 21 'highly inflexible' loops were identified, based on the fact that they moved only minimally during the simulations (the Cα atoms moved less than 0.5 Å). We used Sphinx to predict the structures of these loops (except one, for which the residue numbers given by the authors must be incorrect as they do not correspond to a loop in the protein structure). The results achieved are shown in Figure 6.16. In all but three cases, a decoy with sub-Ångström RMSD was generated; the average RMSD of the best decoy in the top 500 was 0.78 Å. After ranking, the RMSD of the top prediction is also low at 1.15 Å. This is consistent with the results achieved previously; loops with low flexibility are generally predicted with high accuracy.

6.4 Differences Between Loops Displaying High and Low Structural Variation

There may be fundamental differences between loops of multiple and single conformations that causes the disparity in prediction accuracy — for example, if the loops that display high variation are unusual in some aspects of their structure, it would be more difficult to generate and accurately score near-native decoys. To examine this hypothesis, we compared several features of the loops: the quality of the experimental structure, amino acid propensities, bond lengths, bond angles, ϕ/ψ distributions, ω angles, anchor geometry, the presence of contacts (both within the protein and crystal contacts), and the range of scores given by the ranking procedure. Throughout this section, unless

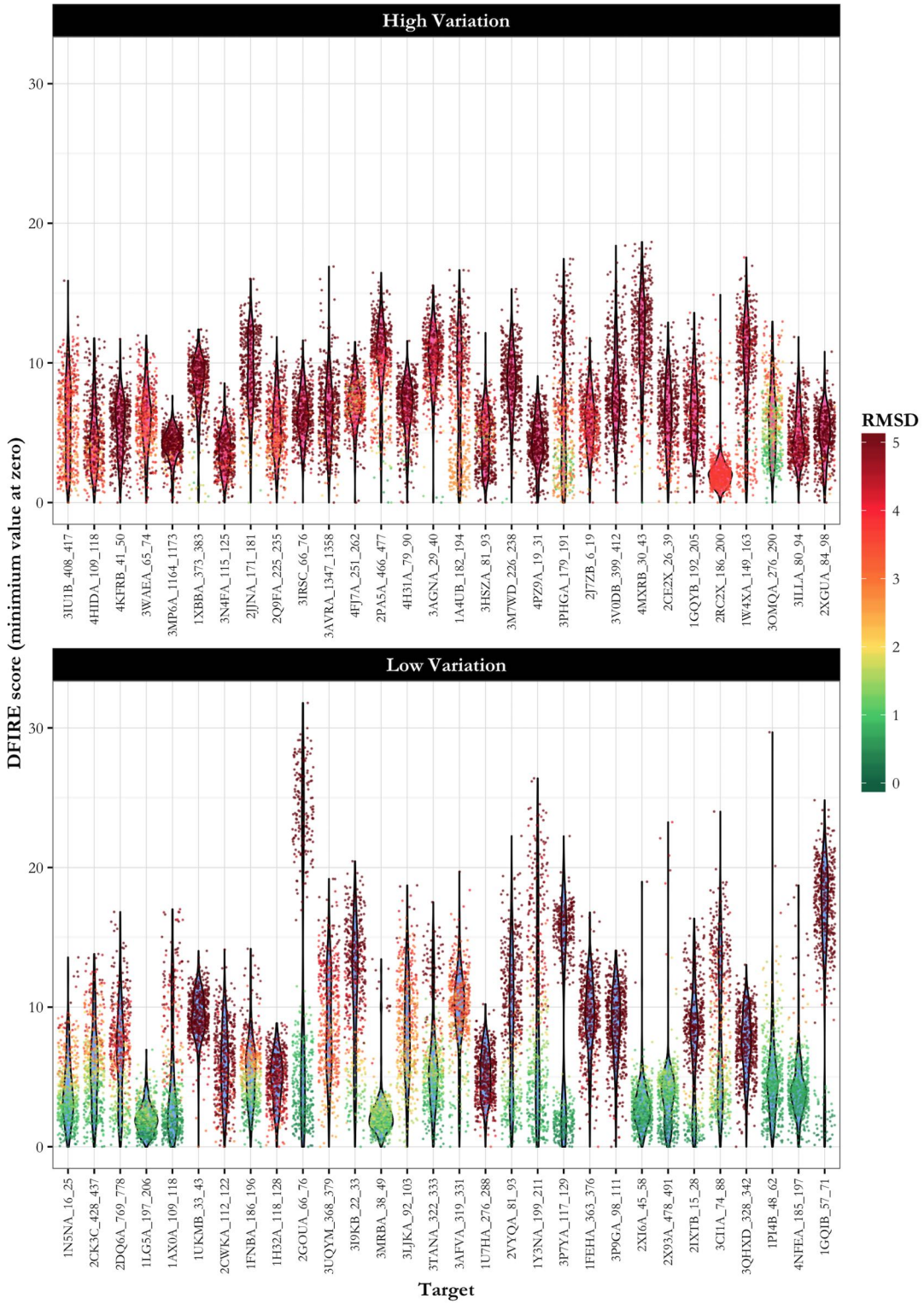

Figure 6.15: dDFIRE scoring for the high and low variation target sets. Violin plots show the distribution of scores for each target. The overlaid points each represent a decoy and are positioned according to their score and coloured by their RMSD to their nearest native conformation. Low variation loops seem to produce a larger spread of scores, and separating the good and bad decoys therefore seems to be easier.

Figure 6.16: Results achieved using Sphinx to predict the structures of the loops in the 'inflexible loop' set, the 20 loops identified by Benson and Daggett (2008). Prediction accuracy is good, with an average top RMSD of 1.15 Å.

otherwise stated, we look at all loops that were found in the PDB with either low (<0.3 Å) or high (>2 Å) variation, not just those loops whose prediction was previously described.

6.4.1 Quality of the Experimentally-Determined Structure

Two measures commonly used to quantify the quality of a structure determined by X-ray crystallography are the B-factor (or temperature factor) and the R-free value. The B-factor is a value assigned to each atom which describes how much the electron density for that atom is spread out. The B-factor is itself often used as a measure of flexibility

— a more disperse electron density implies greater movement. However, it can also be considered as a measure of certainty in the atomic position; the higher the B-factor, the greater the likely error in the reported coordinates.

For each loop found that shows a maximum RMSD between its conformations of over 2 Å (high variation loops), as well as those showing less than 0.3 Å conformational variability (low variation loops), we calculated the normalised B-factor (Parthasarathy and Murthy, 1997) for each backbone atom (for all conformations):

$$B_{\text{normalised}} = \frac{B - \mu}{\sigma}$$

where μ and σ are the mean and standard deviation of the B-factors for all atoms in the structure. By normalising the B-factors, we take into account the quality of the rest of the experimental structure, which will therefore highlight the loops that have more uncertainty relative to their environment. The values calculated for the two sets of loops, plus those for only the loop structures in our high and low variation target sets, are shown in Figure 6.17A. Loops that demonstrate high conformational variability do tend to have higher B-factors than loops of a single conformation, and the targets we predicted using Sphinx are representative of the larger high/low variation sets. However, the difference is not large and therefore we feel that the contrast in prediction accuracy for the two sets cannot solely be attributed to this.

Unlike the B-factor, which gives an indication of local accuracy, the R-free value is a global measure of the quality of the structure (see Chapter 1). R-free values range between 0 and 1, with 0 being the best possible. We looked at the R-free values for each of the sets of loops (high/low variation, all loops/predicted loops) and compared them (Figure 6.17B). Again, structures featuring highly variable loops tend to be of slightly lower quality, but not enough to account for the large difference in prediction accuracy — in both cases, the R-free value is around or slightly above 0.2, which is the value expected from an X-ray crystallography experiment (PDB-101, 2016).

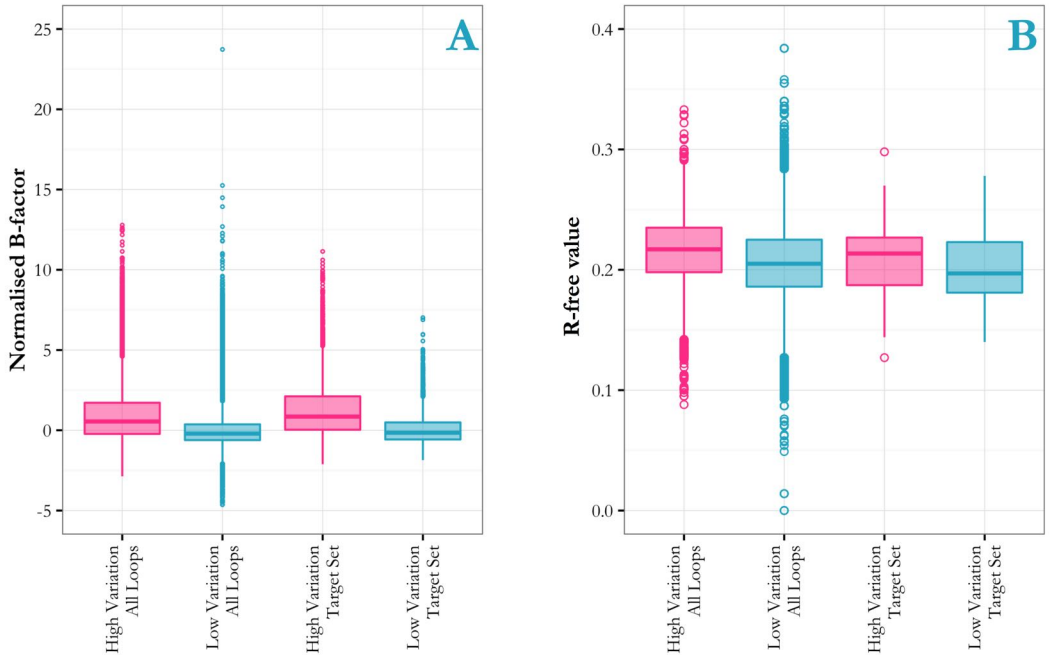

Figure 6.17: Structure quality measures for both high and low variation loops. Four groups of loops are shown: high and low variation sets (showing above 2 Å and below 0.3 Å variation in RMSD respectively), for both all loop structures found in the PDB and the set of target loops described in Section 6.3.1. A — Normalised B-factors for all backbone atoms (N, Cα, C and O). B- R-free values for the structures containing the loops in question.

6.4.2 Amino Acid Composition

The twenty types of amino acid are not present in proteins in equal quantities. Often the relative amounts of the different types depends on the local structure; for example glycine residues are found more frequently in loops than in other regions of the protein (Costantini *et al.*, 2006). To check whether there is any difference in the amino acid compositions of loops in the high and low variation sets, we counted how often each amino acid is present in each of the high and low variation sets and calculated the percentage of residues that are of each type. For comparison, we obtained data from a non-redundant set of loops from the PDB (the loops from a set of proteins with below 90% sequence identity). The results are shown in Figure 6.18. The differences between the high and low variation sets are not large, and neither show large deviations from the non-redundant loop set. A possible slight trend is the increased occurrence of small residues (*i.e.* alanine, glycine and serine) and decreased occurrence of large aromatic residues (*i.e.* phenylalanine, tryptophan and tyrosine) in loops that display

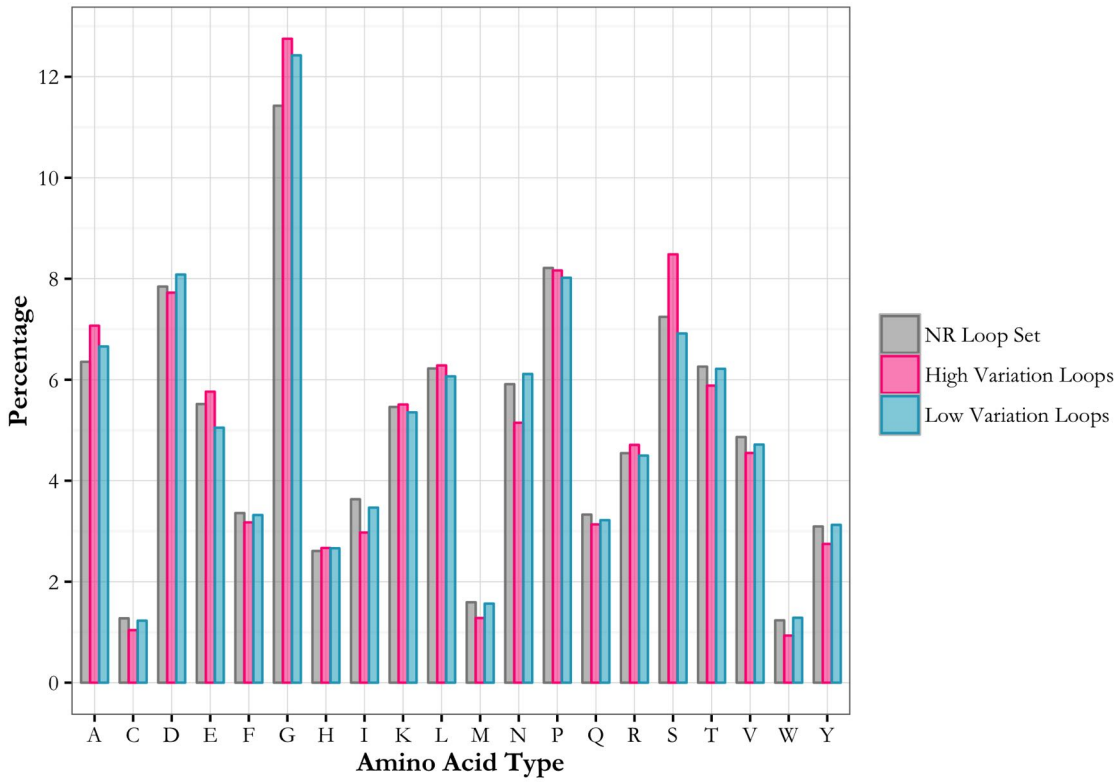

Figure 6.18: The relative occurrences of each amino acid type for the high and low variation loop sets, as well as a set of loops taken from non-redundant protein structures (with less than 90% sequence identity).

multiple conformations — this would make sense, as smaller sidechains may facilitate greater freedom of movement.

6.4.3 Bond Lengths and Angles

In their 1991 paper, Engh and Huber presented mean values and standard deviations for the lengths and angles of each type of covalent bond found in a protein structure, as derived from X-ray crystallography data (Engh and Huber, 1991). These values are used by both MECHANO and Sphinx in their loop building procedures (see Chapters 3 and 4), and therefore, if a native structure displays any deviation from these values, the loop prediction algorithm is less likely to produce the correct conformation.

The atoms of the backbone are designated into six types depending on which residue type they are found in: the peptide nitrogen atom of all residues except proline (NH1); the peptide nitrogen of proline (N); the Cα of all residues except glycine (CH1E); the Cα of glycine (CH2G); the carbonyl carbon (C) and the carbonyl oxygen (O). There

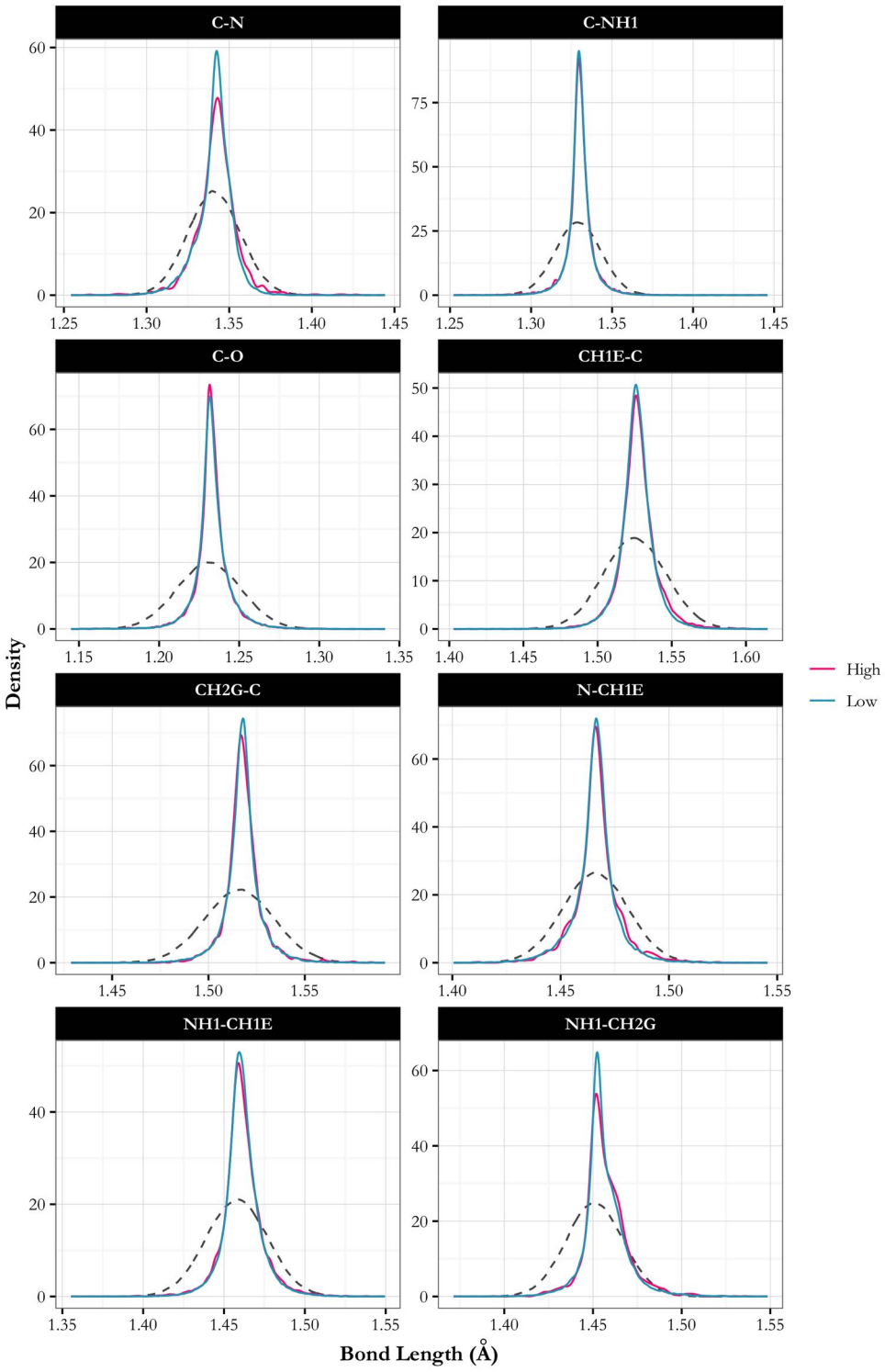

Figure 6.19: Bond length comparison for the high and low variation loop sets. The six atom types are: the peptide nitrogen atom of all residues except proline (NH1); the peptide nitrogen of proline (N); the Cα of all residues except glycine (CH1E); the Cα of glycine (CH2G); the carbonyl carbon (C) and the carbonyl oxygen (O). Dashed lines represent the Gaussian distributions created using the means and standard deviations defined by Engh and Huber (1991). The distributions are very similar for the high and low variation loops.

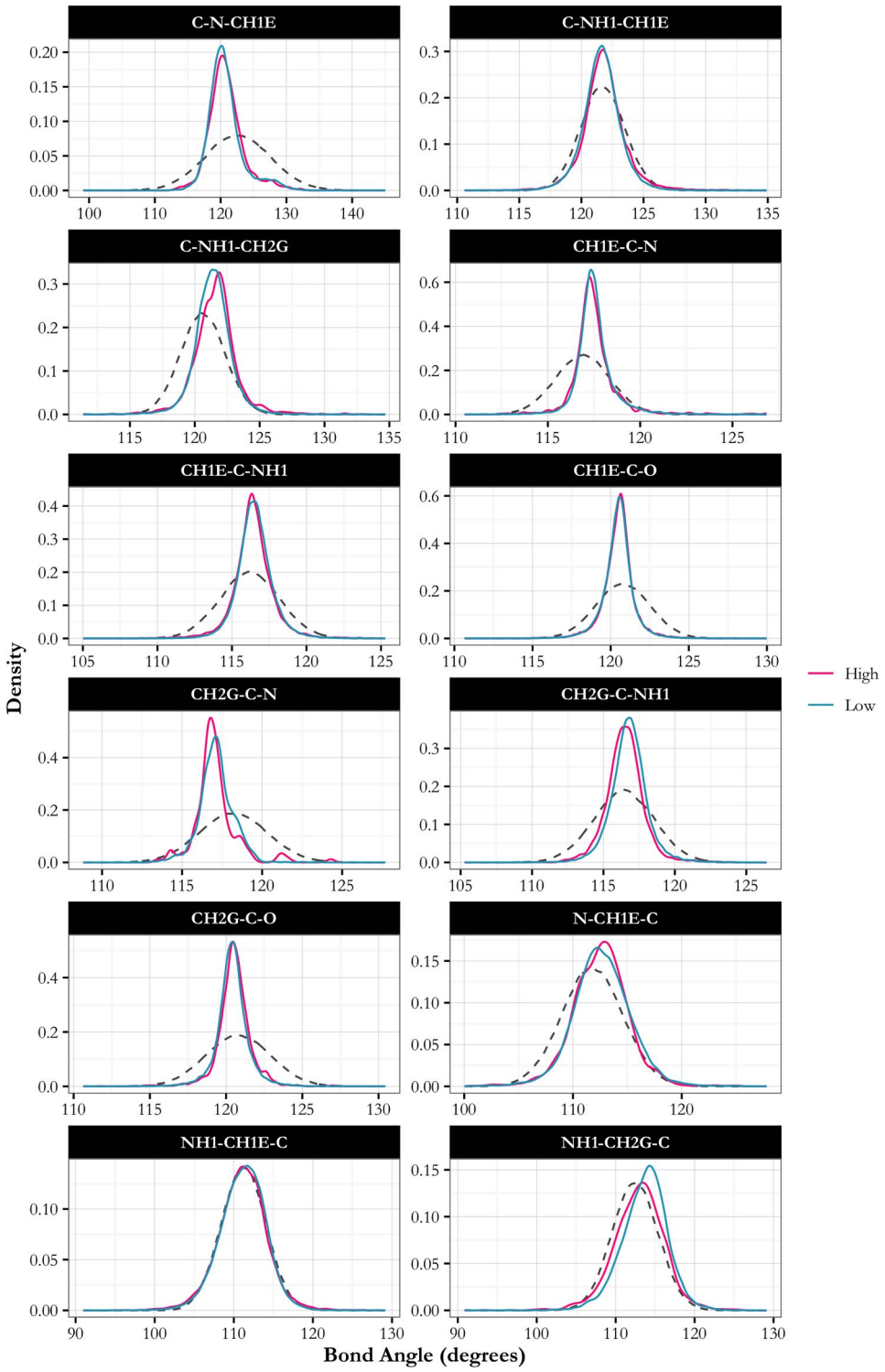

Figure 6.20: Bond angle comparison for the high and low variation loop sets. The six atom types are: the peptide nitrogen atom of all residues except proline (NH1); the peptide nitrogen of proline (N); the Cα of all residues except glycine (CH1E); the Cα of glycine (CH2G); the carbonyl carbon (C) and the carbonyl oxygen (O). Dashed lines represent the Gaussian distributions created using the means and standard deviations defined by Engh and Huber (1991). The distributions are very similar for the high and low variation loops.

are therefore eight types of bond, and twelve types of bond angle. For each structure in the high and low variation categories, we calculated the bond lengths and angles, and compared their distributions to the Gaussian distributions generated by using the mean and standard deviation values given by Engh and Huber (Figures 6.19 and 6.20). It can be seen that the high and low variation sets have very similar bond lengths and angles. In addition, the peak of the distribution is generally located at the mean value given in the literature, although the peaks for the high/low variation sets are both sharper than the Gaussian distributions.

6.4.4 Dihedral Angles

Another factor that could affect prediction accuracy is the dihedral angles found within the loops: the ψ/ψ angle pairs, which describe the orientation of the backbone on either side of the Cα atom; and the ω angle, which describes the rotation around the C-N peptide bond. If loops that can adopt multiple conformations have different distributions of these angles to other loops, then the ability to make near-native predictions will be hindered, since during loop building, these angles are sampled from Ramachandran distributions extracted from experimentally determined structures (most of which show little variation in structure).

To examine this, we firstly calculated all ϕ/ψ angle pairs for the loops in each of the high and low variation sets (including one structure from each observed conformation). We then converted these data into probability densities by partitioning the data points into 5° by 5° bins, and dividing the number of points in each bin by the total number of points. We then subtracted the density of the high variation loop set from the low variation set so that any differences in the two distributions could be observed. We split the residue types into three groups: glycine, proline and all others (since glycine and proline have very distinctive Ramachandran distributions).

The resulting Ramachandran plots are shown in Figure 6.21. For the most part, the two sets of loops produce approximately similar ϕ/ψ distributions. However, there are some differences. For example, for all amino acid types except for glycine and proline, there is a greater density of points for high variation loops at a ϕ/ψ pair of

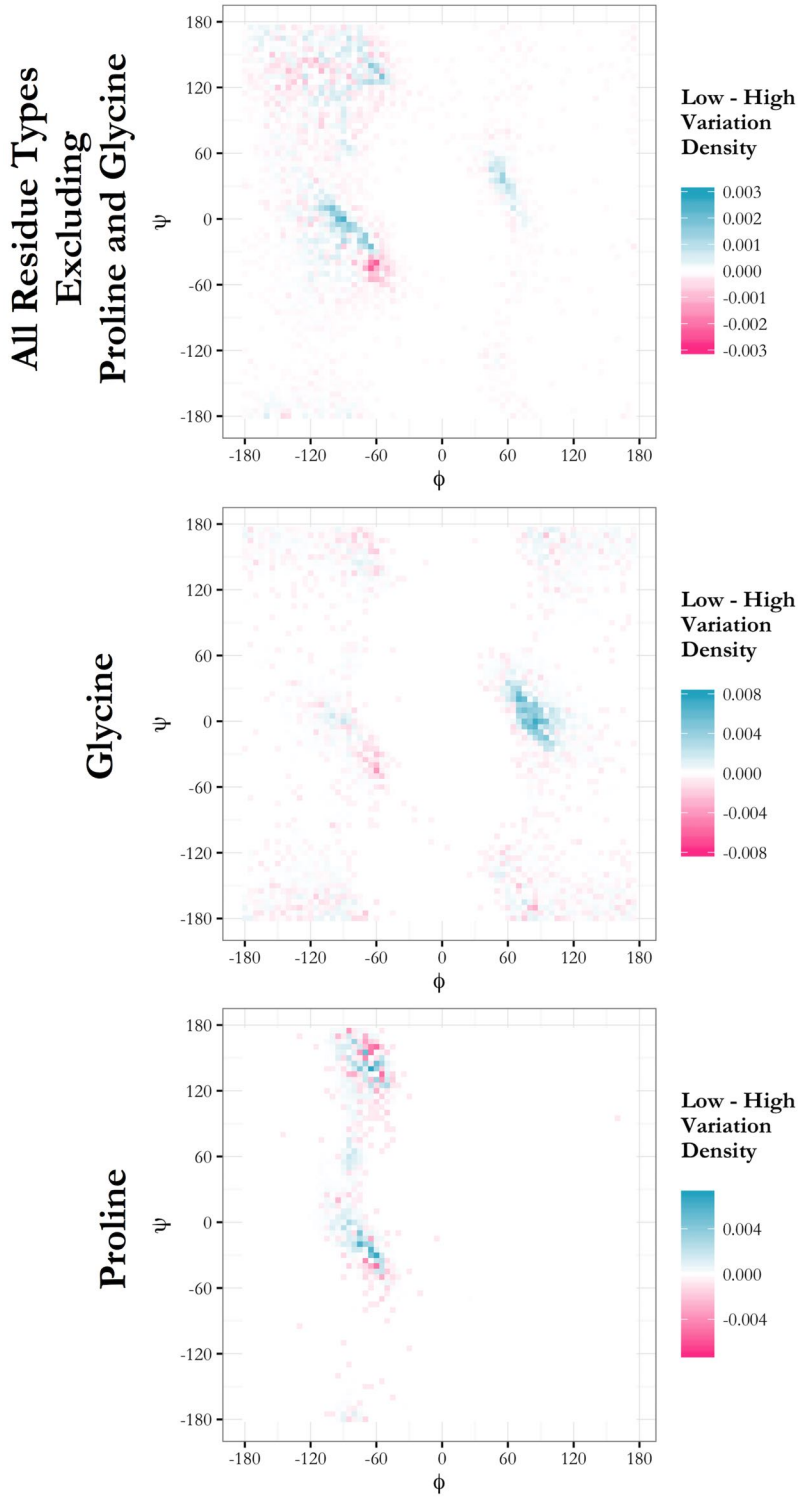

Figure 6.21: Differences in Ramachandran distributions for high and low variation loops. Regions in blue are those with more density for low variation loops than for high variation loops, while pink indicates the opposite. The two distributions are similar.

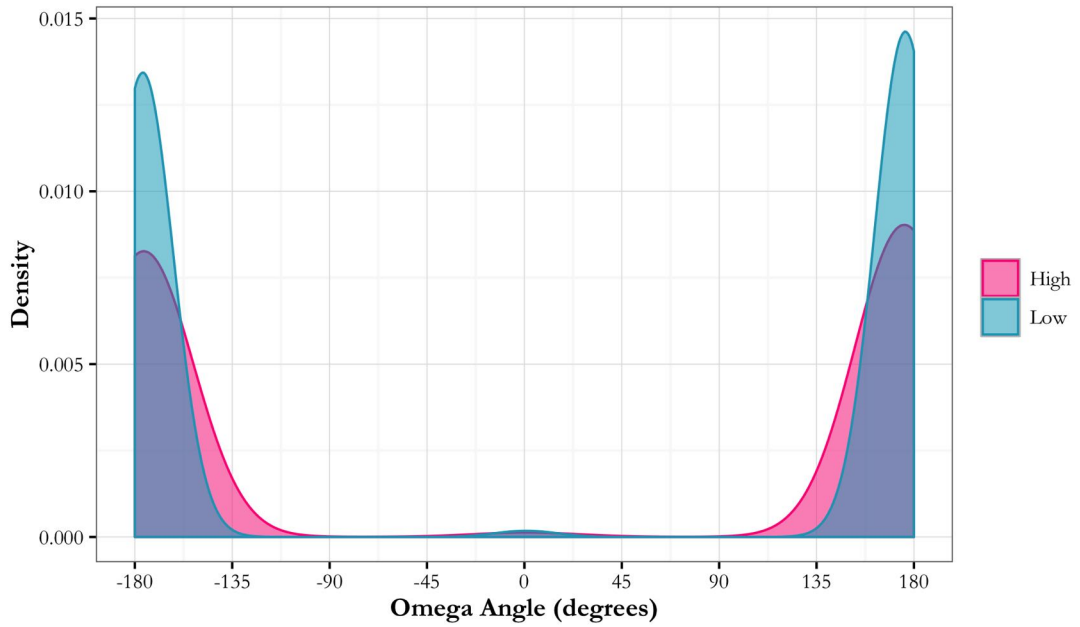

Figure 6.22: Omega angle distributions for high and low variation loops. The peptide bond in loops that only have one known conformation is more strictly planar than those in loops that can adopt multiple conformations.

approximately (-60,-50), and for glycine residues, positive ϕ angles seem to be more prevalent in loops with a single conformation.

Due to the planarity of the peptide bond, in the majority of cases the ω angle has a value of roughly 180° (or 0° for the more rare *cis* conformation). We calculated the ω angles for all residues of all loops in the high and low variation loop sets, and compared their distributions (Figure 6.22). From our results, it appears that the peptide bonds of loops that can adopt multiple conformations may be slightly non-planar — the peak at 180° is broader than that of the low variation loops.

6.4.5 Anchor Geometry

Loop flexibility may also be affected by the geometry of the anchor residues. Intuitively, anchor residues that are far apart should restrict the number of conformations available to a loop, more so than loops whose anchors are close together (Choi *et al.*, 2013). For each loop in our high and low variation sets, we calculated the distances between the Cα atoms of the anchor residues, the same measurements used by FREAD to identify suitable fragments from a database of structures (see Chapter 2). For the four anchor

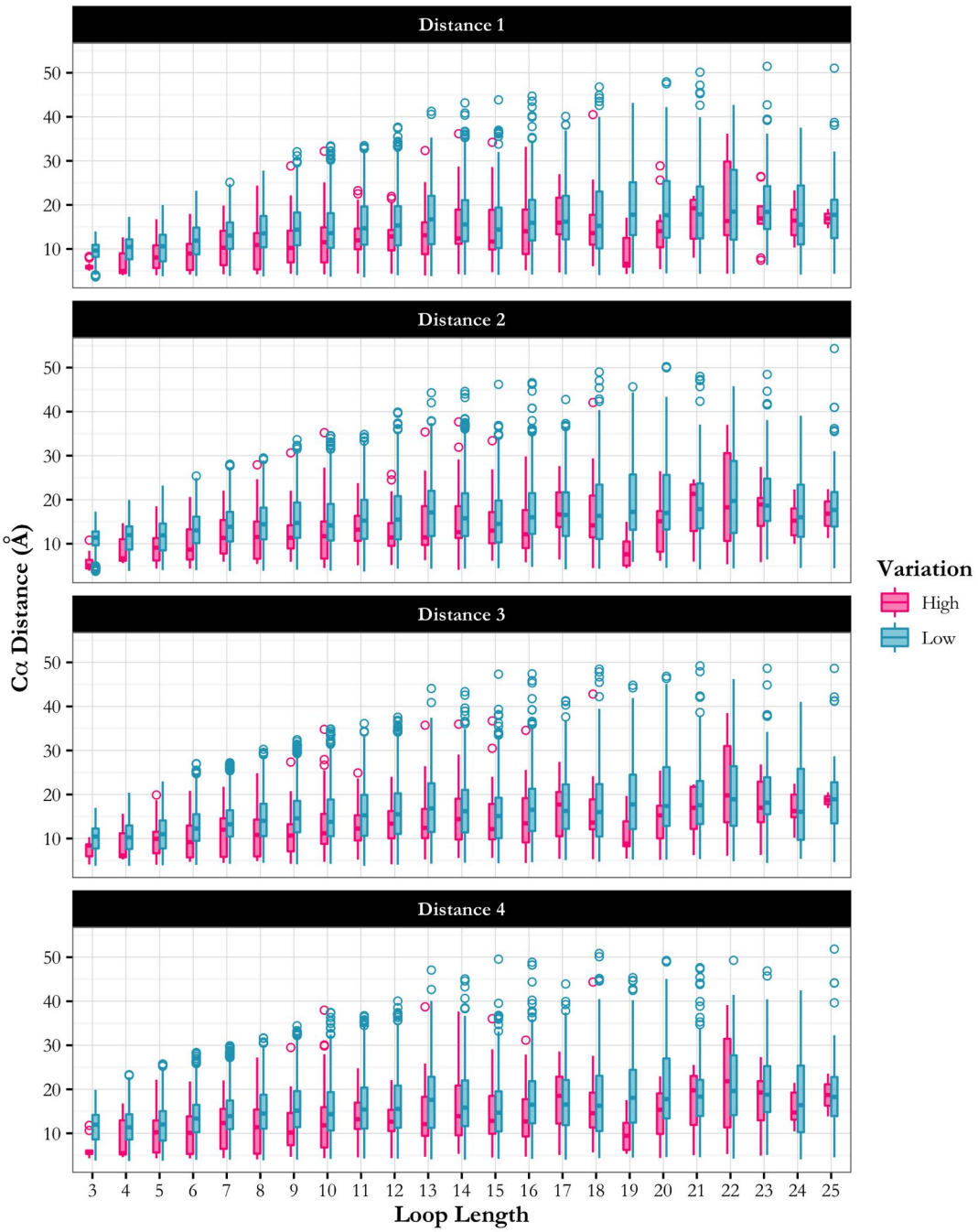

Figure 6.23: Anchor Cα distances for loops in the high and low variation sets. The distances 1-4 are the distances between the Cα atoms of N2-C1, N2-C2, N1-C1 and N1-C2, respectively, where the order of residues is N1-N2-loop-C1-C2.

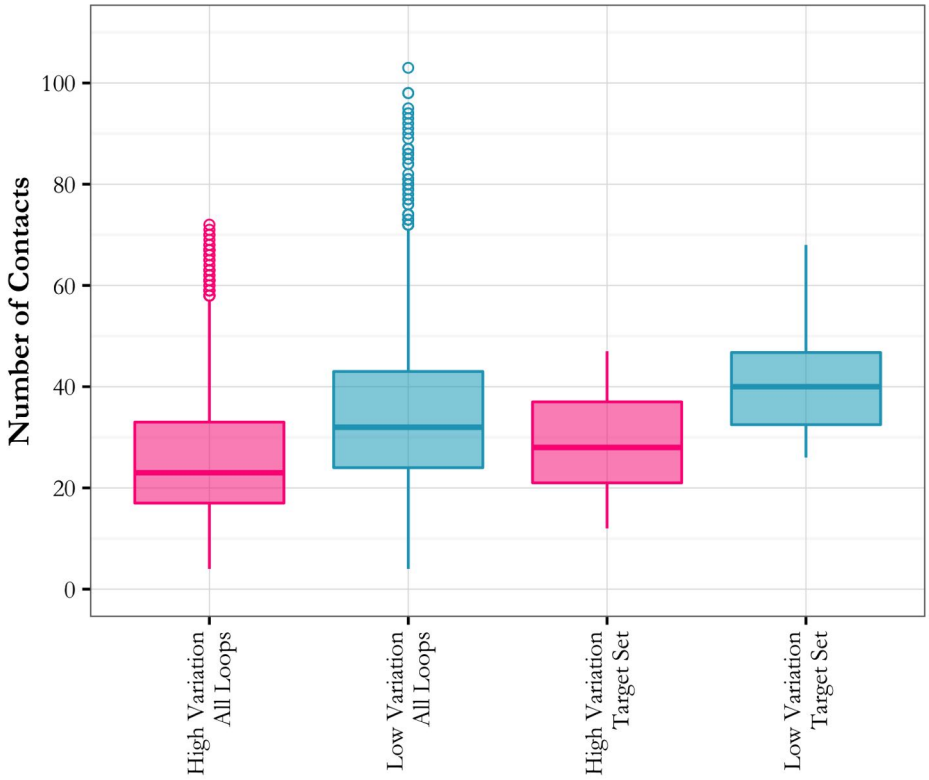

Figure 6.24: The number of residues in contact with the loop, for the high and low variation loop sets (both all loops and those used in the Sphinx prediction analysis). High variation loops tend to have fewer contacts than those of low variation.

residues N1, N2, C1, C2 (two N-anchor and two C-anchor residues, where the order of residues is N1-N2-loop-C1-C2), the distances 1-4 are the distances between the Cα atoms of N2-C1, N2-C2, N1-C1 and N1-C2, respectively. As the number of residues in a loop obviously places limits on how large the anchor separation can be, we split the loops into groups by their length. The results are shown in Figure 6.23. Overall our hypothesis seems to be true; loops with only one known conformation have larger anchor separation than the loops with multiple conformations.

6.4.6 Contacts

We have investigated the number of residues in each protein that are in close proximity to the loop residues in each set. We generated two new sets of high and low variation loops, selected in order to produce the same length distribution in each case (since loop length limits the number of contacts that can be formed). Residues were considered to

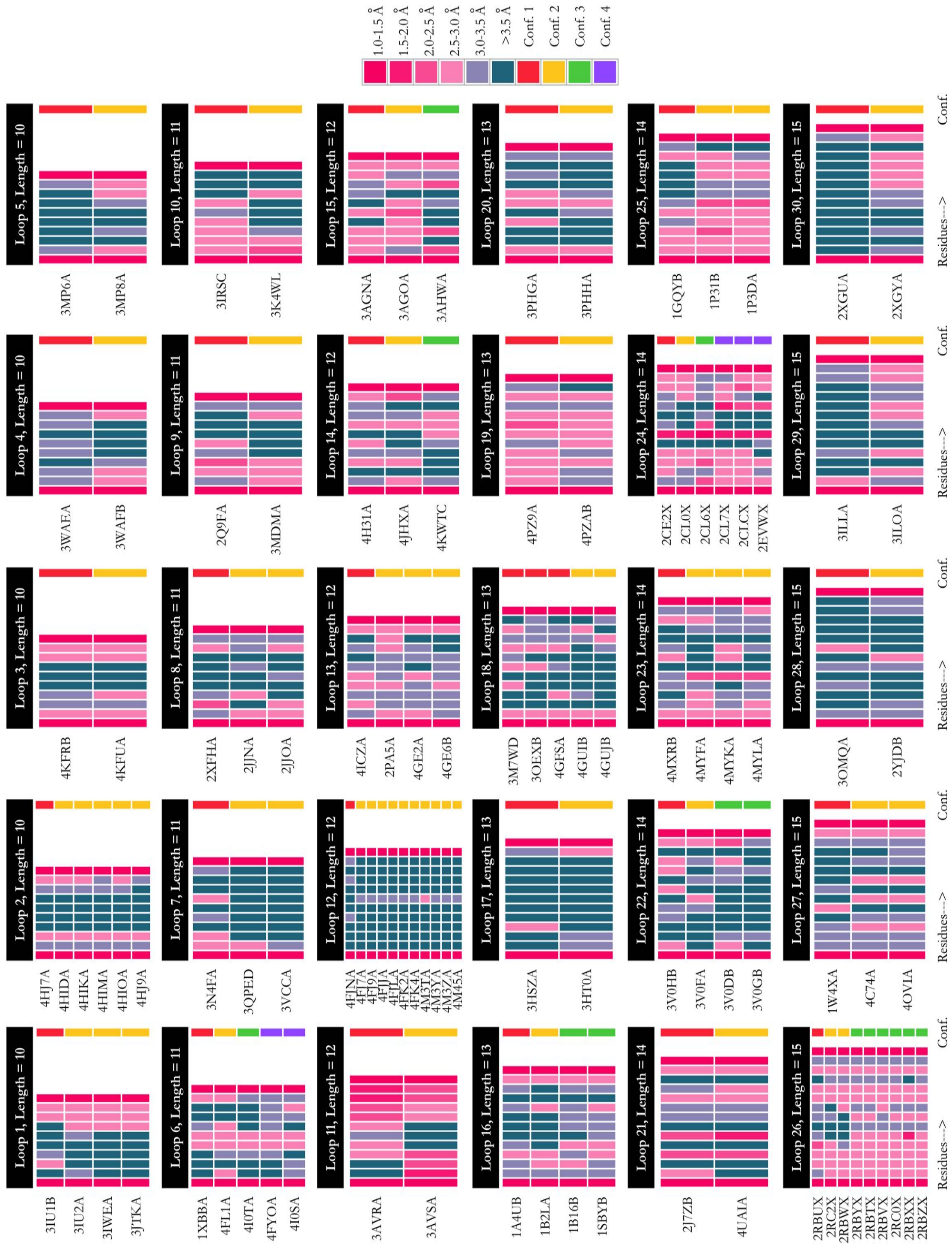

Figure 6.25: Contact profiles for the high variation target set. Each set of profiles with a black heading represents a different protein loop. The different rows within that set correspond to the different copies of that loop present in the PDB, as indicated by their PDB identification codes. Each residue is coloured by its proximity to other residues in the protein. The final column in each profile set indicates the conformation each loop structure belongs to. Different conformations of the same loop have different contact profiles.

Figure 6.26: Contact profiles for the low variation target set. Each set of profiles with a black heading represents a different protein loop. The different rows within that set correspond to the different copies of that loop present in the PDB, as indicated by their PDB identification codes. Each residue is coloured by its proximity to other residues in the protein. The final column in each profile set indicates the conformation each loop structure belongs to. Since the structures in each set have the same conformation, their contact profiles are very similar.

be in contact if any of their atoms were within 6 Å of each other. We also carried out this analysis for the loops in the target sets predicted by Sphinx (Section 6.3.1).

Our results are shown in Figure 6.24. Loops with one observed conformation tend to have a greater number of residues in their vicinity than the loops that show high conformational variability; this trend is also observed in the two loop target sets. This could affect the ability of the scoring functions to select the best decoys. For example, our knowledge-based scoring functions only considers atoms that are up to 8 Å away from the target loop — this result indicates that high variation loops have fewer neighbouring atoms within this range, and therefore less data is being used in the score calculation, which may lead to greater inaccuracy.

We have also produced contact 'profiles' for each loop, shown in Figures 6.25 and 6.26. As expected, for low variation loops, the contact profile is the same or very similar for each version of a loop (since they all have the same conformation). In most cases, loops with multiple conformations have a region with fewer contacts in the centre of the loop, away from the anchors. Occasionally, where there is a close contact (atoms with a distance between them of 1 to 2 Å), these contacts tend to be conserved across all conformations (*i.e.* loop 24, where the seventh residue forms a contact with another residue for all four conformations). For loops of only one conformation, other residues in the protein tend to be found closer to the loop — these neighbouring residues may stabilise the loop structure, and make other conformations unfavourable.

We have also looked at which loop conformations form contacts to other proteins within the crystal. Using PyMOL (Schrödinger, 2010), we identified the residues of each loop that are within 3.5 Å of the protein's symmetry mates. The resulting contact profiles are shown in Figures 6.27 and 6.28. All but four of the high variation loops have at least one conformation that is within 3.5 Å of a symmetry mate; this is true for 17 of the 30 loops in the low variation set. This is one possible reason for the difference in prediction accuracy for the two target sets; none of the prediction methods use any information about the crystal environment when generating their predictions, and therefore they may have pieces of information missing that are required to produce an accurate model. Where multiple conformations are observed, the profiles also differ

Figure 6.27: Crystal contact profiles for the high variation target set. Each set of profiles with a black heading represents a different protein loop. The different rows within that set correspond to the different copies of that loop present in the PDB, as indicated by their PDB identification codes. Each residue is coloured by its proximity to other residues in the protein. The final column in each profile set indicates the conformation each loop structure belongs to. Crystal contacts do not occur as often as contacts to the rest of the protein, but different conformations of the same loop still display different contact profiles, which may offer an insight into why multiple conformations are observed.

Figure 6.28: Crystal contact profiles for the low variation target set. Each set of profiles with a black heading represents a different protein loop. The different rows within that set correspond to the different copies of that loop present in the PDB, as indicated by their PDB identification codes. Each residue is coloured by its proximity to other residues in the protein. The final column in each profile set indicates the conformation each loop structure belongs to. Fewer loops in this set form crystal contacts than the loops in the high variation set, and the profiles are similar for each structure in a set.

between conformations, with some having more contacts than others, or forming contacts in different places. In some circumstances, one conformation of a loop may not be in contact with a symmetry mate at all, while another does (*e.g.* loop 1, where each residue of the first conformation is more than 3.5 Å away from other symmetry mates, but the second forms three contacts to the crystal). This may be one of the reasons that some loops are observed in different conformations; if the space group varies between different crystal structures (*i.e.* the proteins in the crystal are arranged in a different pattern), then this may cause loop structures to change.

6.5 Conclusions

In this chapter we have described our investigation into the conformational flexibility of loops. We have identified a large number of loops for which multiple high-resolution X-ray structures exist in the PDB, and found that in the majority of cases there is very little variation between them. Normally the RMSD between different structures of the same loop is approximately 0.1-0.2 Å. This would indicate that most loops may be relatively static, contrary to the popular view that they display large amounts of flexibility.

On the other hand, structures determined using NMR spectroscopy showed much more variation, with most structures having an RMSD between them of approximately 1 Å, and many more examples existing of loops with large conformational flexibility. We must therefore consider the limitations of each experimental method to decide which set of data gives the most realistic illustration of flexibility. X-ray crystallography, by requiring protein molecules to be in an ordered lattice, may force loops into one conformation and thereby give a false impression of rigidity. However, NMR spectroscopy, whilst requiring the proteins to be in solution (a more natural environment), may overestimate the flexibility of loops — the models in each PDB file are possible structures given the constraints found experimentally, and may therefore not represent true conformations. Instead, there could have been a lack on constraints identified for

that region, leading to ambiguity in the loop position. In practice, the true flexibility is probably somewhere between the two, but this is difficult to demonstrate.

One caveat to consider when interpreting these results is that it cannot definitively be proven that loops are truly inflexible using our methodology. Even though we look at loops with multiple structures available, it may be the case that a loop is able to adopt different conformations, but only one has been observed so far experimentally. As a consequence, some of the loops in our 'low variation' set may actually be conformationally flexible.

When modelling a flexible loop, instead of producing just one prediction, it would be more accurate to produce an ensemble of structures. However, we have shown that the accuracy of prediction is lower for variable loops than it is for conformationally stable ones, making this challenging. We have found that this trend appears with several loop prediction algorithms, of knowledge-based, *ab initio* and hybrid types (Sphinx, FREAD, LEAP and Rosetta). Differences can be observed at all stages of loop modelling, but mainly occur during the ranking steps. We have examined several properties of the loops to try and identify any differences between loops of single and multiple conformations that may cause this discrepancy. While some differences are seen in the local geometry of the loop structures (*i.e.* ϕ, ψ and ω angles), the main differences are in the contacts they form and the distance between their anchors. Loops with only one observed conformation form more contacts to the surrounding protein and have anchors that are further apart than flexible loops, and also tend to have fewer crystal contacts. Crystal contacts are important to consider; proteins arranged differently in the crystal may sometimes be the reason that different conformations arise, making the structure unrepresentative of the true native state of the protein.

It may be possible to devise a series of rules that can be used to predict the flexibility of a given loop. By running Sphinx, an ensemble of decoys is returned, and the distribution of pairwise RMSDs can be produced — we have shown that the distributions of these values are different for loops of high and low variation, and as such this property may be used predict which category a loop belongs to. This test can be used even if the target structure has not been solved experimentally. If a native conformation of the loop is

known, the Sphinx prediction can be compared to this structure, and the accuracy used to estimate flexibility — *i.e.* if a loop is poorly predicted, it is more likely to be flexible. By considering other properties of the loop in addition to this, for example the distance between the anchor residues, the flexibility of a loop may be predicted. This information would be useful for applications such as docking and computational antibody design.

Pre-Chapter Image: The binding site of an antibody, with its six CDR loops, shown from above.

7 Conclusions and Future Directions

Contents

7.1 Conclusions

7.1.1 Chapter 2

In Chapter 2, we reported the results of our research into the behaviour of the knowledge-based loop prediction algorithm FREAD (Deane and Blundell, 2001; Choi and Deane, 2010) when modelling the H3 loop of antibodies. We tested multiple fragment databases from which loop conformations could be selected, and found that a database of antibody fragments results in the best prediction accuracy. A database including 'H3-like' fragments, however, did not improve model quality, as these fragments were seldom judged to be suitable conformations by the algorithm.

Investigations into the parameters used by FREAD (the sequence and anchor geometry similarity cutoffs) yielded new values which improve prediction accuracy. However, coverage was reduced: the proportion of targets for which a prediction was made fell from 80% to 44%. The average accuracy of prediction was also improved by using a statistical potential to rank the decoys.

We tested FREAD on the set of targets from the second Antibody Modelling Assessment (AMA-II), using the antibody fragment database. When using the default parameters, and considering the average RMSD of the best model (of the five submitted), FREAD was ranked sixth out of the seven methods. However, its predictions seemed to be more consistent; the average RMSD calculated over all submitted models was better than all but two of the other methods. No significant improvement was observed by using the new parameter values on this test set.

7.1.2 Chapter 3

Chapter 3 described the development of a new *ab initio* algorithm, MECHANO, specifically designed to model H3 loops. We found that results are improved by the addition of clash checks, the ability to 'switch' the direction of loop building, a modified closure algorithm that enforces decoys to be feasible in terms of ϕ/ψ space, and resampled dihedral data. This ϕ/ψ resampling procedure, which maintains the characteristics of H3 dihedral distributions while supplementing the limited amount of data available with that of general loops, is a novel method that has not been used in loop prediction before.

We compared MECHANO's accuracy to that of several other freely-available loop modelling algorithms. MECHANO's performance was better than all of the *ab initio* methods; the only algorithm to produce higher-quality predictions was FREAD. However, the ranking method we used for this test, dDFIRE (Zhang *et al.*, 2004), fails to rank the decoys correctly, achieving an average RMSD of 3.57 Å while the best possible result given the decoy sets generated is 1.82 Å. We also tested MECHANO on the AMA-II target set; the average RMSD of the best decoy generated was of a similar accuracy to FREAD, but again performance was let down by the ranking method.

7.1.3 Chapter 4

In Chapter 4, we presented our new loop modelling algorithm, Sphinx, which combines knowledge-based and *ab initio* methodologies in a novel way. At this stage, we focussed only on decoy generation. The algorithm is based on FREAD and MECHANO, the algorithms investigated in Chapters 2 and 3 of this thesis. By searching a database of H3 loops for fragments shorter than the target loop, and using their structural information alongside *ab initio* techniques to build loop conformations, we are able to generate high-quality decoy sets. This methodology also allows us to use the information contained within loops of different lengths, which may still have similar conformations. More near-native conformations can be generated using Sphinx than could be made using FREAD or MECHANO separately, and unlike FREAD, Sphinx could produce a prediction for every target.

We compared Sphinx to RosettaAntibody, one of the leading *ab initio* H3 prediction algorithms, and showed that our hybrid approach produces better decoys. In addition, in a non-native environment, we showed that Sphinx's ability to produce near-native conformations is only minimally affected, and results were similar to those achieved using the knowledge-based method H3Loopred. The general version of Sphinx, which uses fragments and dihedral data from all protein types, also produces excellent results.

7.1.4 Chapter 5

The aspect of a loop modelling algorithm that often limits prediction accuracy is the ranking step. We therefore tested multiple ways of performing this step on the decoy sets generated for the general loop set (Chapter 4), and reported the results in Chapter 5. We investigated various clustering algorithms, the performance of ten freely-available scoring functions (two physics-based and eight knowledge-based), and the application of a minimisation protocol; the best approach we found was a fast minimisation step using Rosetta (Mandell *et al.*, 2009) followed by scoring using the SOAP-Loop statistical potential (Dong *et al.*, 2013). Using this ranking system, Sphinx is able to produce high-quality predictions, more accurate than those achieved by a straightforward *ab initio* algorithm.

When applied to the H3 decoy sets generated by Sphinx, our ranking method again produces good results — Sphinx achieves better accuracy than RosettaAntibody, as well as most of the algorithms tested during the AMA-II. All these results show that loops of a different length to the target can still be of use when carrying out structure prediction.

7.1.5 Chapter 6

Chapter 6 described our investigations into loop flexibility, in the context of structure prediction. We identified a large number of loops for which there are multiple structures in the PDB, and showed that, in general, the variation between them (in terms of backbone RMSD) is small. However, there is a large difference between structures solved by X-ray crystallography and those determined using NMR spectroscopy — in the latter case, loops appear to be more flexible.

There are examples of loops that are able to adopt different conformations. The accuracy of prediction for these loops is low compared to loops that display little structural variation. This is true when using Sphinx, FREAD, LEAP and Rosetta. We investigated aspects of the loop structures that may account for this distinction; the main differences are in the contacts formed (including crystal contacts) and the geometry of the anchor residues.

7.2 Future Directions

There are many questions arising from this work that offer opportunities for future research. It is interesting that 'H3-like' loops, identified from protein types other than antibodies, do not improve prediction accuracy and in fact are rarely chosen by our database search algorithm as candidate structures. Either these loops are not actually H3-like, or some aspect of the algorithm is stopping them from being selected — for example, the loops may be similar in structure but not in sequence, giving low ESS scores and meaning they do not get selected. Alternatively, since our tests are performed

on known structures, the H3-like loops may be similar in structure to H3 loops, but not to those already observed.

By incorporating 'switching' into our loop prediction algorithms, we have seen that some target loops display a clear preference in which direction they are built (*i.e.* whether residues are added sequentially to the N- or C-anchor). Most H3 loops show a preference for the C-N direction. We initially hypothesised that this was due to the presence of the C-terminal kink in many H3 structures, however this proved not to be the case. More thought is therefore required to solve this puzzle — perhaps it is a result of properties of the fragments themselves; we would not have picked up on this since we considered the decoy sets as a whole.

We have proposed that Sphinx should be used only after initially running a knowledge-based method such as FREAD, since FREAD also produces high-accuracy predictions, but is less computationally expensive. However, FREAD can produce poor results when the target loop has a novel structure. It would be helpful, therefore, to formulate rules that indicate when to ignore a FREAD prediction and run Sphinx. In other words, we wish to have a measure of confidence in the FREAD prediction. One option available to address this is the application of more restrictive sequence similarity and anchor geometry parameters. This should ensure that the only predictions FREAD produces are highly accurate, leaving Sphinx to predict those structures which FREAD could not.

The excellent performance of our hybrid loop prediction algorithm, Sphinx, when predicting the structures of both H3 and general loops, suggests that the same methodology could be successfully applied to other loop types. For example, we could develop a membrane protein version, or versions for the five canonical CDR loops. In particular, applying Sphinx to the latter case should be successful, since a study into CDR loops has showed that different length CDRs can still belong to the same canonical class (Nowak *et al.*, 2016). Unlike other loop prediction algorithms, Sphinx would be able to use these different length loops to make its predictions. Further to this, the basic idea behind Sphinx (*i.e.* the integration of a knowledge-based and an *ab initio* algorithm) could be expanded to use different algorithm combinations.

A recurring problem through all of our work has been that of accurate decoy selection. In particular, ranking was shown to be poor when predicting in a non-native environment (*i.e.* onto a model, not a crystal structure). Since one of the key applications for Sphinx, the *in silico* design of antibodies, requires accurate loop prediction onto models, this is an important area that needs improvement. An H3-specific energy function would be a possible way of increasing ranking accuracy, however it is likely that the limited number of H3 structures available is not sufficient to provide the data for this. Perhaps a hybrid method could also work here, using a combination of physics and knowledge-based approaches. Minimising or running molecular dynamics simulations on antibody structures using a physics-based energy function could yield more structures from which to acquire data — this data could then be used in a knowledge-based framework.

In addition to these issues that in general relate to the performance of loop prediction, there are many possibilities for future work arising from our flexibility study. An interesting outcome of our research is the observed flexibility differences between loop structures solved using X-ray crystallography and those solved using NMR spectroscopy: NMR structures display more conformational variation. To gain further insights into which approach gives a more realistic impression, we plan to compare structures that have been solved using both techniques.

It would be helpful in many situations to know whether a loop is flexible or not — for example, one may wish to know if a CDR is likely to change its conformation upon binding. Using the results described in Chapter 6, it may be possible to formulate a set of rules that give a prediction of structural variability; for example using the distance between anchor points or the number of contacts that a loop forms. An extension of this would be the prediction of mutations that decrease/increase flexibility, or the manner in which a loop is able to move.

In accordance with the majority of the work presented in this thesis, a specific case to investigate in terms of flexibility would be antibodies. It has been seen in specific cases that germline antibodies are more flexible than mature ones; it would be interesting to look at this on a larger scale by investigating the structures of each type present in

the PDB (assuming enough data is available). If the decrease of flexibility is a cause of increased affinity, this may give helpful guidance about which mutations to make during the process of antibody design.

Appendices

A
Target Sets

Contents

The following pages contain tables that detail each of the target sets used for the work presented in this thesis. Target sets are given in the order that they appear in the main text.

Table A.1: Details of the 'Half-SAbDab' target set of H3 loops used to test FREAD (Chapter 2).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
3zlqD	3	AGR	3cfbB	6	GQGRPY	1a6tD	8	RDDYYFDF
4bfbD	3	AGR	2eh7H	6	EYDEAY	3qnxB	8	RGDVYNRQ
2q8aH	3	SHF	1sm3H	6	VGQFAY	1kemH	8	WGSYAMDY
1cftB	3	KDY	3s62H	6	GADVAY	1jpsH	8	DTAAYFDY
1i3gH	3	WGY	1fl3H	6	GQGRPY	1g7jB	8	ERDYRLDY
2q8bH	3	SHF	4hxbH	6	LRAVDY	1g7mB	8	ERDYRLDY
1cfqB	3	KDY	1sjvA	6	HYFRSY	2jixH	8	ERLGIGDY
1hh6B	3	KDY	3cfcH	6	GQGRPY	1fdlH	8	ERDYRLDY
1cfnB	3	KDY	3d85B	6	NWDVAY	1iqdB	8	PDPDAFDI
1h8nA	3	WGY	3k1kC	6	NVGFEY	1a7rH	8	ERDYRLDY
3sgdJ	4	GTTY	4d9rH	6	EGGVNN	1g7lB	8	ERDYRLDY
3mnwB	4	GWLH	3r08H	7	FDWDKNY	2hkfH	8	YGNHPFAY
2xxmB	4	SGSS	1yn1H	7	GYSSMDY	1wejH	8	YDYGNFDY
3ggwD	4	PMDY	1t66D	7	YGYHGAY	1a7qH	8	ERDYRLDY
3u0wH	4	QMGY	2c1oB	7	DDYDGAF	1iehA	8	YSGGALDA
2vxtH	4	GLRF	1mrcH	7	LRGYFDY	1uwgH	8	FYGNFFDY
1tqbB	4	GTDY	4fabH	7	SYYGMDY	3zkmH	8	AYRNAMDY
1emtH	4	SSAY	3s36H	7	VTDAFDI	4ffvD	8	FRDVFFDV
4jo2H	4	NFDL	1mrfH	7	LRGYFDY	1qbmH	8	YDYGNFDY
3mnzB	4	GWLH	2iffH	7	GNYDFDG	3k74B	8	DWATGLAK
3qg7H	4	EGDY	1axtH	7	YFYSFSY	1vfaB	8	ERDYRLDY
3c5sB	4	PMDY	1kc5H	7	GGTGFDY	1f4yH	8	HFYAVLDY
2xv6D	4	SGSS	1mreH	7	LRGYFDY	1mamH	8	DPYGPAAY
3t2nI	4	GFAY	3g5xB	7	AGRGFPY	1ahwB	8	DNSYYFDY
3ujtI	4	GFAY	1yqvH	7	GNYDFDG	3o6kH	8	DYDYGVDY
2ddqH	4	YGGY	1jhkH	7	WGGNSAY	1vfbB	8	ERDYRLDY
2r4sH	4	GFGY	3uzeA	7	GWEGFAY	1uz6F	8	ETGTRFDY
3mnvD	4	GWLH	1nj9H	7	GGGLFAF	1f4wH	8	HFYAVLDY
2dtgC	4	EWAY	1flrH	7	SYYGMDY	2wucH	8	WAWPAFDY
1ndmB	5	WEMDY	2xknD	7	DSGGFAY	4at6O	8	GGSGLLAY
1dl7H	5	DYGPY	2r29H	7	DYEGFAY	1a7pH	8	ERDYRLDY
1pskH	5	KSFDY	1ynkH	7	GYSSMDY	1a7oH	8	ERDYRLDY
1ktrH	5	QSGAY	4i9wG	7	GNYVFDY	2vxqH	8	LDGYTLDI
1ineH	5	HRFVH	3lexH	7	GNYLPAY	1kipB	8	ERDYRLDY
1dqqB	5	WGGDV	1ghfH	7	VEAGFDY	3j2xB	8	VDGYAMDY
2dqhH	5	WDGDY	1kcvH	7	GGTGFPY	1xcqB	9	FLLRQYFDV
1nakI	5	DDFDI	3g5vB	7	AGRGFPY	3hb3C	9	HEYYYAMDY
1ggbH	5	EGYIY	3r06F	7	FDWDKNY	2aepH	9	VDYGTNYDY
1ic5H	5	WAGDY	1zeaH	7	RSWYFDV	2itdA	9	ERGDGYFAV
1ngyB	5	RDMDY	2nr6F	7	WVRQMDY	3vg0H	9	HDGHETMDY
1ngxH	5	RDSDY	3g5zB	7	AGRGFPY	3f7vA	9	ERGDGYFAV
1xgtB	5	WGGDV	2d7tH	7	GWWGMDV	1fskI	9	GARDTWFAY
3a67H	5	WDGDY	2j88H	7	YYGSVDY	3ls4H	9	GTTIVAGDV
1aj7H	5	YYGIY	1kcuH	7	GGTGFDY	4g3yH	9	NYYGSTYDY
1xgqB	5	WGGDV	1mrdH	7	LRGYFDY	2itcA	9	ERGDGYFAV
1gpoH	5	WHGDY	3u0tD	7	LYSLPVY	2fd6H	9	WGPHWYFDV
1j1pH	5	WDGDY	3ietD	7	GKVRNAY	1m7iB	9	GGAVGAMDY
2zkhH	5	WSFLY	1kcrH	7	GGTGFTY	2ih3A	9	ERGDGYFAV
1uacH	5	WDGDY	3leyH	7	GNYLPAY	2ck0H	9	GRDGEAMDY
1cr9H	5	DLHDY	3upcE	7	SYGAFDY	2hh0H	9	GAYYIKEDF
1ngwH	5	RDMDY	1tetH	7	RSWYFDV	2g2rB	9	YDYAADRVY
1cu4H	5	DLHDY	3qo1B	8	RGDVYNRQ	1k4dA	9	ERGDGYFAV
3fctD	5	RDMDY	3v52H	8	LTNGYLDV	4gq9H	9	GVTRGYFDV
1xgpB	5	WGGDV	3o6mH	8	DYDYGVDY	2dweA	9	ERGDGYFAV
3cxdH	5	QMGDY	1fl6B	8	DGSYAMDY	4hjjH	9	RGIRSAMDY
1e4wH	5	SPSDY	1b4jH	8	GFLPWFAD	2fbjH	9	LHYYGYNAY
1n7mL	5	RDSDY	3mj9H	8	HFYTYFDV	4hc1M	9	ASDSDGFAY
1p4bH	5	GLFDY	1fgnH	8	DNSYYFDY	3or6A	9	ERGDGYFAV
1dqjB	5	WGGDV	1i7zB	8	YDSPLGDY	3j2zH	9	GVTRGYFDV
2dqiH	5	WDGDY	3j30D	8	VDGYAMDY	3gnmH	9	SFIGYNFDF
1ic7H	5	WAGDY	1mimH	8	DYGYYFDF	3c1fH	9	GRLSLGFDY
2dqgH	5	WDGDY	3v4uH	8	LTNGYLDV	1mfeH	9	GGHGYYGDY
1j1oH	5	WDGDY	4am0H	8	GGSHAMDY	3ks0H	9	LFGSYYFDY
1xgrB	5	WGGDV	2vq1F	8	IPGFGFDY	4dvbH	9	GELTYAMDY
3dsfH	5	QMGDY	3zknC	8	AYRNAMDY	1igtB	9	HGGYYAMDY
1jn1H	5	EKDLP	3o61H	8	DYDYGVDY	3b9kD	9	NRESYYFDY
2yssB	5	WDGDY	3s35H	8	HDGTNFDY	2h8pA	9	ERGDGYFAV
2vh5H	6	GRFFDY	4alaH	8	GGSHAMDY	4ffzZ	9	WFFPWYFDV

Table A.1: cont.

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
2dwdA	9	ERGDGYFAV	3esuF	10	SGLLRYAMDY	1aifB	10	RPLFYYAVDY
2dd8H	9	DTVMGGMDV	1f11B	10	DYGSTYGFAY	1mj8H	10	KHYFYDGVVY
3or7A	9	ERGDGYFAV	2vdkH	10	PLYDYYAMDY	1ztxH	10	SASYGDYADY
1eo8H	9	PTGRLWFSY	3l95B	10	GSGFRWVMDY	1egjH	10	GDGIHGGFAY
2w0fA	9	ERGDGYFAV	1qkzH	10	DPLEYYGMDY	3h0tB	10	GIVFSYAMDV
1s5hB	9	ERGDGYFAV	4h20H	10	RGLTGALFAY	3pp3I	10	NVFDGYWLVY
1nfdH	9	AGRFDHFDY	2uylY	10	EETVRASFGN	3dv4B	10	DGYYVDAMDY
4g5zH	9	DLRTGPFDY	3mxwH	10	DWERGDFFDY	2vdoH	10	PLYDYYAMDY
2ih1A	9	ERGDGYFAV	1ce1H	10	EGHTAAPFDY	43c9B	10	YGYGGDRFSY
2hg5A	9	ERGDGYFAV	1a31H	10	AGGYYTGGDY	1qygH	10	VPQLGRGFAY
1s5iH	9	YDDYTWFTY	3nidE	10	PLYDYYAMDY	1q72H	10	VPQLGRGFAY
3vi4H	9	ALALYAMDY	3w12C	10	DPYGSKPMDY	2vdlH	10	PLYDYYAMDY
4jr9H	9	SWYNRAMDY	2vl5C	10	LKPGGTWFAY	1rukH	10	LLWYDGGAGS
3fb7A	9	ERGDGYFAV	1igiH	10	SSGNKWAMDY	3o45H	10	LYGFTYGFAY
1c5dB	9	EDGWNYFDY	1c1eH	10	GTTIVRAMDY	4g6kH	10	NRYDPPWFVD
3v0vH	9	QGRGYTLDY	1rumH	10	LLWYDGGAGS	2gfbJ	10	GDYYGSRGAY
4hwbH	9	MPNWGSFDY	1q0yH	10	WVLDYYGMDY	3gjfM	10	ELLPYYGMDV
3ls5H	9	GTTIVAGDV	2zjsH	10	SPYYYGNWDY	2agjH	10	TSGWDIEFEY
2g75C	9	DTVMGGMDV	1qnzH	10	HYGSAYAMDY	1ad0D	10	DRGLRFYFDY
3c1eH	9	GRLSLGFDY	1a4kH	10	AERLRRTFDY	2vdqH	10	PLYDYYAMDY
2gsgD	9	GRGYVWFAY	3eotH	10	GFGDGGYFDV	3ze0E	10	PLYDYYAMDY
2arjH	9	LIGSWYFDF	4jznC	10	GPLSRGYYDY	1dbaH	10	GDYVNWYFDV
3hplA	9	ERGDGYFAV	4g6mH	10	NRYDPPWFVD	2xzcH	10	LTQSSHNDAN
2p7tA	9	ERGDGYFAV	3esvG	10	SGLLRYAMDY	4h88H	10	SGYGYYAMEY
3ogcA	9	ERGDGYFAV	2d03H	10	NRGYSYAMDS	1jguH	10	GDAYSRYFDV
1r3lB	9	ERGDGYFAV	1s3kH	10	GTRDGSWFAY	2o5yH	10	GTTIVWAMDY
1r3kB	9	ERGDGYFAV	2vdnH	10	PLYDYYAMDY	1q0xH	10	WVLDYYGMDY
1um5H	9	SSMSDPGAN	3s96A	10	GGDNYLWFAY	32c2B	10	GYYGSSHSPV
3ehbC	9	HEYYYAMDY	4jzoD	10	GPLSRGYYDY	4dvrH	10	DPWELNAFNV
1jv5B	9	QYGNLWFAY	3pp4H	10	NVFDGYWLVY	2cmrH	10	DNPTLLGSDY
1mhhD	9	FLLRQYFDV	3jwdH	10	DPWELNAFNV	1rulH	10	LLWYDGGAGS
2a0lD	9	GYDYFAMDY	1lk3H	10	GVPGNNWFPY	2dlfH	10	IYYHYPWFAY
1mfcH	9	GGHGYYGDY	1mhpX	10	GFGDGGYFDV	2ajsH	10	YDYYGNTGDY
3fb5A	9	ERGDGYFAV	4dgiH	10	SGYGYYAMEY	3et9F	10	SGLLRYAMDY
3v0wH	9	QGRGYTLDY	3ld8C	10	RDYGTYYFDY	3hi6H	11	SYDFWSNAFDI
3ntcH	9	GIPGYAMDY	3o2wH	10	GTTIVRAFDY	1ohqB	11	ALEPLSEPLGF
1yuhH	9	YAYCRPMDY	4f33D	10	GGYDGRGFDY	1a6uH	11	YDYYGSSYFDY
1f90H	9	YDDYTWFTY	4etqH	10	SNYRYDYFDV	2fjgB	11	FVFFLPYAMDY
4ffyH	9	WFFPWYFDV	3u30F	10	TWLLRWVMDL	1q9kB	11	DHDGYYERFSY
1orqB	9	GVDYFAMDY	1beyH	10	EGHTAAPFDY	3dvnB	11	EYYRWYTAIDY
1wt5A	9	SGGPYFFDY	3lohA	10	DPYGSKPMDY	1ai1H	11	ENHMYETYFDV
2qjdB	9	VEGNTIKFT	3ve0A	10	QLYGNSFFDY	3bdyH	11	WGGDGFYAMDY
1pz5B	9	GGAVGAMDY	1ru9H	10	LLWYDGGAGS	12e8H	11	GHDYDRGRFPY
1yntD	9	SSTWYYFDY	2ajyH	10	YDYYGNTGDY	2gkiB	11	GAYKRGYAMDY
2w60A	9	ATTATELAY	25c8H	10	DPPYYGHGDY	3phoB	11	DIGYGNSPFAY
4dw2H	9	GELTYAMDY	1kb5H	10	SRTDLYYFDY	1fveD	11	WGGDGFYAMDY
3u9uC	9	IPDDHYFDY	1lo0Y	10	GDSFLVWFTF	1oarH	11	MWYYGTYYFDY
2xqyG	9	LGYVYGFDY	1ucbH	10	GLDDGAWFAY	2v17H	11	DNGAARATFAY
1bbdH	9	YYSYYDMDY	3iflH	10	RTTTADYFAY	2y06H	11	YDYYGGSYFDY
2adfH	9	DNPYYALDY	1vpoH	10	AYYGYVGLVH	3okoB	11	DHDGYYERFAY
3cfiI	9	WLGGRDWYD	3nigE	10	PLYDYYAMDY	3oknB	11	DHDGYYERFAY
1eapB	10	SYYGSSYVDY	4g6aC	10	ITTTTYAMDY	1ad9B	11	EKTTYYYAMDY
1yc7A	10	QCGVRSIREY	1dlfH	10	IYYHYPWFAY	1p7kB	11	GGYRPYYAMDY
3difD	10	GNYDRAWFAY	4aehH	10	YDGDYGSFDY	2r23B	11	DHDGYYERFSY
1rmfH	10	GGWLLLSFDY	1ncwH	10	LLWYDGGAGS	1i8mB	11	GGYRPYYAMDY
2v7hH	10	RGYWAYDFDY	4gagH	10	ITTTTYAMDY	2r1yB	11	DHDGYYERFSY
1bgxH	10	YYYGYWYFDV	3ze2H	10	PLYDYYAMDY	1oauJ	11	MWYYGTYYFDY
1ay1H	10	YYYGYWYFDV	1l7iH	10	NLGPSFYFDY	1ngqH	11	YDYYGSSYFDY
1wz1H	10	IYYHYPWFAY	2bfvH	10	LNYAVYGMDY	1xf2B	11	GGYRPYYAMDY
1lo3Y	10	GDSFLVWFTF	1bfoF	10	EGHTAAPFDY	3iy3B	11	KGYVEGGGLDY
2ak1H	10	YDYYGNTGDY	4hixH	10	YDHYSGSSDY	1qoka	11	GTPTGPYYFDY
4hgwB	10	DHDGYYERFS	3mc1H	10	EGSNNNALAY	1yy9D	11	ALTYYDYEFAY
3dusB	10	DGYYVDAMDY	1jnhH	10	GRSLYYTMDY	2adiB	11	HEDGNWNYFDY
4f3fB	10	GGYDGRGFDY	2o5zH	10	GTTIVWAMDY	3b9vA	11	WGGDGFYAMDY
2ajxH	10	YDYYGNTGDY	1lo2H	10	GDSFLVWFTF	1ifhH	11	RERYDENGFAY
2ajzH	10	YDYYGNTGDY	3o41A	10	LYGFTYGFAY	3eolE	11	TLGLVLDAMDY
3r1gH	10	GPFSPWVMDY	2xwtA	10	LDWNYNPLRY	3v7aF	11	SGTTAYFAMDY
3hmxH	10	RRPGQGYFDF	1bfvH	10	LNYAVYGMDY	1ehlH	11	RSGYKYYALDY

Table A.1: cont.

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
1ob1E	11	NYYRFDGGMDF	3g04B	12	LEPGYSSTWSVN	3fo2H	13	HWGGYYIPYGMDH
3nh7K	11	ERWHVRGYFDH	2p46B	12	GGYELRDRTYGQ	3k81A	13	NLQTGTLSGARLY
3j42K	11	WGNYPHYAMDY	1it9H	12	NRDYSNNWYFDV	1xiwH	13	SGYYGDSDWYFDV
1oaxJ	11	MWYYGTYYFDY	3c08H	12	RDYDYDGRYFDY	1r0aH	13	SAITSVTDSAMDH
1d5iH	11	GHSYYFYDGDY	3rvxD	12	TGVYRYPERAPY	3klhD	13	SAITSVTDSAMDH
3be1H	11	WGGDGFYAMDY	2p4aD	12	GGDALVATRYGR	1q9wB	13	DIYSFGSRDGMDY
3cfeH	11	SDYGNYGRGDY	3p0vI	12	ESRVSFEAAMDY	2zchH	13	ADYGFNSGEAMDY
1n4xH	11	DRHDYGEIFTY	3macH	12	YFDTYNNYGFAN	2zc1H	13	ADYGFNSGEAMDY
1fptH	11	DFYDYDVGFDY	1kfaH	12	EGLLLDYFTMDY	2x1oA	13	GAGWAGTMTDYNY
1yy8B	11	ALTYYDYEFAY	4evnC	12	HMGYQVRETMDV	1kyoJ	13	SEYYSVTGYAMDY
3auvF	11	FVFFLPYAMDY	3ejyE	12	LYYGYGYWYFDV	3i02B	13	DMRRFDDGDAMDY
1sjxA	11	EDRHRIGTVGY	2xa8H	12	GSHYFGHWHFAV	1nmaH	13	SGGSYRYDGGFDY
1igcH	11	WGNYPYYAMDY	3nabH	12	ELYQGYMDTFDS	3cx5U	13	SEYYSVTGYAMDY
3iy0H	11	RGFAGAASFAY	1dn0D	12	PPHDTSGHYWNY	2gsiH	13	GVITMMGYQAMDY
4hfwH	11	GVVGATNEIDF	3v0aC	12	GYRTDTRIPGGS	3gizH	13	DIQYGNYYYGMDV
1tziB	11	SSYAYYAAMDY	2a6jH	12	SVYYGGSYYFDY	2zckH	13	ADYGFNSGEAMDY
3phqB	11	DIGYGNSPFAY	1ol0A	12	EPRIPRPPSFDY	2gjzH	13	HWGGYYIPYGMDH
4jqiH	11	SRQFWYSGLDY	1il1A	12	ISTTRDYYALDY	1yefH	14	WGFIPVREDYVMDY
1hinH	11	RERYDENGFAY	2a6kB	12	SVYYGGSYYFDY	2aabH	14	SNYVGYHVRWYFDV
1a6vJ	11	YDYYGSSYFDY	4kucF	12	WVYGNFDSALDY	4dgvH	14	DISLVRDAFIYFDF
3l7eH	11	MGSDYDVWFDY	4fqyH	12	HGNYYYYSGMDV	1adqH	14	TRSYVVAAEYYFHY
1nc2D	11	RGSYPYNYFDV	2p47B	12	GGYELRDRTYGQ	2qhrH	14	HIYYGSSHYYAMDY
1acyH	11	ENHMYETYFDV	3rvvD	12	TGVYRYPERAPY	3oayH	14	KGSDRLSDNDPFDA
1oayJ	11	MWYYGTYYFDY	1dsfH	12	SPIYYDYAPFTY	3oazM	14	KGSDRLSDNDPFDA
1e6oH	11	PVVRLGYNFDY	3cviH	12	KPYYGNFAWFAY	1cz8H	14	YPYYYGTSHWYFDV
1oaqH	11	MWYYGTYYFDY	3p0gB	12	KDYGAVLYEYDY	2j5lC	14	DGPTDSSGYGGFGY
3rajH	11	TLITTGYAMDY	1c12B	12	SLTWLLRRKRSY	2qqnH	14	GELPYYRMSKVMDV
1ncbH	11	GEDNFGSLSDY	3rvtD	12	TGVYRYPERAPY	1gigH	14	DFYDYDVFYYAMDY
3bpcB	11	DHDGYYERFSY	2ht2E	12	LYYGYGYWYFDV	1wcbH	14	DKGSYGNYEAWFAY
4a6yB	11	YDYYGGSYFDY	2p44B	12	GGYELRDRTYGQ	3qhfH	14	TLRVSGDYVRDFDL
3u9pH	11	CRRDSSYVVDA	3c09H	12	RDYDYAGRYFDY	1y01D	14	LDFDVYNHYYVLDY
1keqH	11	RSGYKYYALDY	2vxvH	12	VGSGWYPYTFDI	3gkwH	14	QGLWFDSDGRGFDF
1nl0H	11	ASIAAARVLDY	2ht3E	12	LYYGYGYWYFDV	1op5H	14	KGSDRLSDNDPFDA
1ejoH	11	RAFDSDVGFAS	4fqvH	12	HGNYYYYSGMDV	3n9gH	14	DWGSNYVWGSYPKY
2mcpH	11	NYYGSTWYFDV	3naaH	12	QLYQGYMDTFDS	1yehH	14	WGFIPVREDYVMDY
1cl7H	11	DRHDYGEIFTY	1otsE	12	LYYGYGYWYFDV	1zlsH	14	KGSDRLSDNDPFDA
2r2eB	11	DHDGYYERFSY	3ncjH	12	QLYQGYMDTFDS	4hfuH	14	VGGEWGSGRYYLDH
1nc4B	11	RGSYPYNYFDV	2fecI	12	LYYGYGYWYFDV	3v6zA	14	EGAYSGSSSYPMDY
3b2vH	11	VSIFGVGTFDY	2r9hC	12	LYYGYGYWYFDV	3iywH	14	DWGSNYVWGSYPKY
1fvcD	11	WGGDGFYAMDY	3ngbJ	12	GKNCDYNWDFEH	4jn2H	14	AAYYSYYNYDGFAY
1oazJ	11	MWYYGTYYFDY	3rhwF	13	DGDYYRYGRYFDY	3v6fE	14	EGAYSGSSSYPMDY
3nz8H	11	GYGKNVYAMDY	1a14H	13	SGGSYRYDGGFDY	1zlvM	14	KGSDRLSDNDPFDA
2fjfH	11	FVFFLPYAMDY	1r24D	13	GGTGTRSLYYFDY	3cfjF	14	LDFDVYNHYYVLDY
2r2hB	11	DHDGYYERFSY	3kr3H	13	DNGDYVGEKGFDI	4jn1H	14	AAYYSYYNYDGFAY
3oklB	11	DHDGYYERFAY	1kxvD	13	SLRYGLPGCPIIP	1yegH	14	WGFIPVREDYVMDY
3ulvH	12	ELYQGYMDTFDS	3g6dH	13	GLGAFHWDMQPDY	1wc7B	14	DKGSYGNYEAWFAY
4eneC	12	LYYGYGYWYFDV	4f91d	13	EDDYDGTPDAMDY	2bseF	14	RSGGFSSNRELYDG
3nn8C	12	RGGSSHYYAMDY	2ipuH	13	RAHTTVLGDWFAY	1dqdH	14	PPGYYGSGPYAMDY
3cvhH	12	KPYYGNFAWFAY	3o2dH	13	EKDNYATGAWFAY	3ujiH	14	LGFEGDYSGSFFDY
3gbmH	12	HMGYQVRETMDV	4jamH	13	LPRGQLVNAYFRN	1dqlH	14	GNPPYSSGWGGGDY
1igmH	12	HRVSYVLTGFDS	1fbiH	13	LYYYGTSYGVLDY	1oakH	14	DPADYGNYDYALDY
3ttlH	12	RTVTMGRYAMDY	1nmcB	13	SGGSYRYDGGFDY	1seqH	14	GAMFGNDFKYPMDR
2htlC	12	LYYGYGYWYFDV	3k7uA	13	RPQGDYVTAHYDY	3rkdH	14	IKSVITTGDYALDY
1j05B	12	FGYYVSDYAMAY	1bz7B	13	GGTGTRSLYYFDY	1kn4H	14	WGFIPVREDYVMDY
6fabH	12	SEYYGGSYKFDY	3cxhJ	13	SEYYSVTGYAMDY	1zlwM	14	KGSDRLSDNDPFDA
1ospH	12	SRDYYGSSGFAF	3riaJ	13	DGDYYRYGRYFDY	1y18H	14	LDFDVYNHYYVLDY
3nacH	12	QLYQGYMDTFDS	3rifH	13	DGDYYRYGRYFDY	1yeiH	14	WGFIPVREDYVMDY
3p0yH	12	ESRVSFEAAMDY	2j6eI	13	LGPDDYTLDGMDV	1yeeH	14	WGFIPVREDYVLDY
3ejzC	12	LYYGYGYWYFDV	3sdyH	13	EPPLFYSSWSLDN	2qscH	14	DRYYETSGSNAFDV
2feeJ	12	LYYGYGYWYFDV	2gk0H	13	HWGGYYIPYGMDH	1yecH	14	WGFIPVREDYVMDY
4kylH	12	GMWVSTGYALDF	3fo0H	13	HWGGYYIPYGMDH	3ln9A	15	GTRKNVWTRQHPFDY
3eobH	12	GIYFYGTTYFDY	2ntfB	13	GSLYYNNYGWFGY	3vg9C	15	EGNYYDGGSVRYFDY
4dtgH	12	LGGYDEGDAMDS	3efdH	13	SWEAYWRWSAMDY	3mluH	15	LYLFEGAQSSNAFDL
3na9H	12	ELYQGYMDTFDS	3eysH	13	RAHTTVLGDWFAY	2b1hH	15	LGGDYEDSGADAFDF
1qlrB	12	PPHDTSGHYWNY	3mckH	13	SSGSIVIATGFAY	3hi1H	15	GPVPAVFYGDYRLDP
2p45B	12	GGYELRDRTYGQ	1n6qH	13	SAITSVTDSAMDH	3bkcH	15	RGFYGRKYEVNHFDY
3fmgH	12	NHGSPDFHVMDY	4hs8H	13	SSIYYGKDYVLDY	2b1aH	15	LGGDYEDSGADAFDF
1deeD	12	VKFYDPTAPNDY	3ivkA	13	QGYRRRSGRGFDY	3hzkB	15	DINPGSDGYYDALDY

Table A.1: cont.

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
2vyrI	15	PWYPFMASKGSEFDY	1opgH	16	HPFYRYDGGNYYAMDH	2fx8J	18	EGTTGWGWLGKPIGAFAH
3ujjH	15	LGFWGANRGGGGMDV	3qehE	16	DLEMRDGNNHGSHLEF	3grwH	18	TYGIYDLYVDYTEYVMDY
4aq1D	15	SDKYNFDTSHAGYGY	1etzB	16	RTFSYYYGSSFYYFDN	2ny7H	18	VGPYSWDDSPQDNYYMDV
3k80B	15	DLYGLGSHMENEYDS	1u0qB	16	RMPYSGDYRSSGTYDY	2x1pB	18	DLHRPYGPGTQRSDEYDS
3ijyD	15	DISPSYGVYYEGFAY	4jdvA	16	GKYCTARDYYNWDFQH	4jm4H	18	HRHHDVFMLVPIAGWFDV
3vgaC	15	EGNYYDGGSVRYFDY	3u7wH	16	GKYCTARDYYNWDFEH	3ru8H	18	VGPYSWDDSPQDNYYMDV
3effD	15	QPSYHMYSWWVALDY	4fqkE	16	DVQYSGSYLGAYYFDY	4dk6A	18	DRFPTMEVVTIMTNEYDY
1faiH	15	SFYGGSDLAVYYFDS	3eyfB	16	DGKCGGGRCYSGLLDY	3r0mA	18	KWRPLRYSDYPSNSDYYD
3mlrH	15	LYLFEGAQSSNAFDL	4gxvI	17	ELLMDYYDHIGYSPGPT	1rz8D	19	VYEGEADEGEYDNNGFLKH
3qxuD	15	DDNYVTASWRSGPDY	3sm5I	17	GGLEPRSVDYYYYGMDV	2ny2D	19	VYEGEADEGEYDNNGFLKH
1op9A	15	TEVAGWPLDIGIYDY	3u6rH	17	VVIPNAIRHTMGYYFDY	3mlxI	19	DFGEYHYDGRGFQCEGFDL
3pjsD	15	QPSYHMYSWWVALDY	4hk3J	17	GGLEPRSVDYYYYGMDV	2nxzD	19	VYEGEADEGEYDNNGFLKH
3ebaA	15	TEVAGWPLDIGIYDY	4hkxA	17	AGLEPRSVDYYFYGLDV	3skjI	19	YDSGYDYVAVAGPAEYFQH
4idlA	15	DGRRFDGARWREYES	3sobH	17	GSGHVNAVKNYGYVMDY	2ny3D	19	VYEGEADEGEYDNNGFLKH
2qqlH	15	EDFRNRRRLWYVMDY	1mvfB	17	SSRWMDYSALTAKAYNS	1g9nH	19	VYEGEADEGEYRNNGFLKH
4f37F	15	RGFYGRKYEVNHFDY	1f58H	17	EEAMPYGNQAYYYAMDC	4ersH	19	EKDHYDILTGYNYYYGLDV
3mltH	15	LYLFEGAQSSNAFDL	4hpoH	17	LGGRYYHDSSGYYYLDY	1rzjH	19	VYEGEADEGEYDNNGFLKH
3qxtA	15	DDNYVTASWRSGPDY	3npsB	17	TFHIRRYRSGYYDKMDH	3sn6N	19	CPAPFTRDCFDVTSTTYAY
3hzmB	15	DINPGSDGYYDALDY	4hkbE	17	AGLEPRSVDYYFYGLDV	1gc1H	19	VYEGEADEGEYDNNGFLKH
2qqkH	15	EDFRNRRRLWYVMDY	1ikfH	17	HTLYDTLYGNYPVWFAD	1yylR	19	VYEGEADEGEYDNNGFLKH
3so3C	16	DLGIAARRFVSGAFDI	1ri8A	17	GWSSLGSCGTNRNRYNY	2ny5H	19	VYEGEADEGEYDNNGFLKH
4jkpH	16	GKYCTARDYYNWDFEH	2x1qA	18	SSRVFYTEVLQTTTGYDY	2ny1D	19	VYEGEADEGEYDNNGFLKH
3eyoD	16	DGKCGGGRCYSGLLDY	3juyE	18	VGEWGWDDSPYDNYYMDV	2ny4D	19	VYEGEADEGEYDNNGFLKH
4fzeH	16	TVYYDILTGLYTNFHY	4fqlH	18	DRLESSAMDILEGGTFDI	3idxH	19	DIGLKGEHYDILTAYGPDY
3lqaH	16	GGNYYNLWTGYYPLAY	4jm2A	18	HRHHDVFMLVPIAGWFDV	4ej1D	19	DGSQYRSTYSFRDKPDYGS
2xqbH	16	DPAAWPLQQSLAWFDP	4dkaA	18	DRFPTMEVVTIMTNEYDY			
3lmjH	16	GGNYYNLWTGYYPLAY	1n0xH	18	VGPYSWDDSPQDNYYMDV			

Table A.2: Details of the H3 target set from the second Antibody Modelling Assessment, used to test FREAD (Chapter 2), MECHANO (Chapter 3) and Sphinx (Chapter 5).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
4kq3H	8	DSEYYFDH	4kmtH	10	YDGIYGELDF	4m61B	11	GRLRRGGYFDY
4kq4H	8	EVRRSMDY	4m43H	10	RNYDGSWFAY	4m6oH	14	VYSSGWHVSDYFDY
4m6mH	8	WYYKPFDV	4kuzH	11	NDGYRGYAMDY			
4mauH	8	YDGYSFDY	4m7kH	11	SGYYGNSGFAY			

Table A.3: Details of the H3 target set used to test MECHANO (Chapter 3).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
2vxtH	4	GLRF	1b2wH	8	GFLPWFAD	4dagH	14	VLSRASGMPDAFDI
1tqbB	4	GTDY	1ztxH	10	SASYGDYADY	1dqlH	14	GNPPYSSGWGGGDY
1emtH	4	SSAY	1lo4H	10	GTSFVRYFDV	3eyfB	16	DGKCGGGRCYSGLLDY
1fj1B	4	GLDS	2xzcH	10	LTQSSHNDAN	4jdvA	16	GKYCTARDYYNWDFQH
2r4rH	4	GFGY	1s3kH	10	GTRDGSWFAY	4hhaB	16	EGLSRDNSGFTGLIDY
2eh7H	6	EYDEAY	3s96A	10	GGDNYLWFAY	3so3C	16	DLGIAARRFVSGAFDI
4f9pC	6	SDYGAY	3fmgH	12	NHGSPDFHVMDY	2hfgH	16	RVCYNRLGVCAGGMDY
1z3gH	6	GWDVAY	1jfqH	12	SEYYGGSYKFDY	4jm4H	18	HRHHDVFMLVPIAGWFDV
3d85B	6	NWDVAY	4dtgH	12	LGGYDEGDAMDS	4nzrH	18	HRHHDVFMLVPIAGWFDV
1bafH	6	GWPLAY	1hezB	12	VKFYDPTAPNDY	4eowH	18	RPPRFYDKTEYWEDGFDV
3s35H	8	HDGTNFDY	4ky1H	12	GMWVSTGYALDF	2fx7H	18	EGTTGWGWLGKPIGAFAH
1uweH	8	FYGNFFDY	1y18B	14	LDFDVYNHYYVLDY	4fqlH	18	DRLESSAMDILEGGTFDI
1iqdB	8	PDPDAFDI	1yeeH	14	WGFIPVREDYVLDY			
1plgH	8	GGKFAMDY	2dquH	14	VSHYDGSRDWYFDV			

Table A.4: Details of the first preliminary H3 target set used to develop Sphinx (Chapter 4).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
1lo4H	10	GTSFVRYFDV	1ztxH	10	SASYGDYADY	3s96A	10	GGDNYLWFAY
1s3kH	10	GTRDGSWFAY	2xzcH	10	LTQSSHNDAN			

Table A.5: Details of the second preliminary H3 target set used to develop Sphinx (Chapter 4).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
1lo4H	10	GTSFVRYFDV	1dqlH	14	GNPPYSSGWGGGDY	3eyfB	16	DGKCGGGRCYSGLLDY
1s3kH	10	GTRDGSWFAY	1yeeH	14	WGFIPVREDYVLDY	3so3C	16	DLGIAARRFVSGAFDI
1ztxH	10	SASYGDYADY	2dquH	14	VSHYDGSRDWYFDV	4hhaB	16	EGLSRDNSGFTGLIDY
2xzcH	10	LTQSSHNDAN	4dagH	14	VLSRASGMPDAFDI	4jdvA	16	GKYCTARDYYNWDFQH
3s96A	10	GGDNYLWFAY	2hfgH	16	RVCYNRLGVCAGGMDY			

Table A.6: Details of the H3 training set used to develop Sphinx (Chapter 4).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
4hxbH	6	LRAVDY	1blnB	10	YYRYEAWFAS	4dgyH	14	DISLVRDAFIYFDF
1za6B	6	SLNMAY	4g6mH	10	NRYDPPWFVD	1fnsH	14	DPADYGNYDYALDY
1z3gH	6	GWDVAY	1jn6B	10	GRSLYYTMDY	2hrpH	16	SGGIERYDGTYYVMDY
3cfbB	6	GQGRPY	1eapB	10	SYYGSSYVDY	3qehA	16	DLEMRDGNNHGSHLEF
2uziH	6	GRFFDY	3wihH	10	FSNYVYPFDY	4fqjH	16	DVQYSGSYLGAYYFDY
1igyB	6	EGEVPY	2vl5A	10	LKPGGTWFAY	4olzH	16	GKYCTARDYYNWDFEH
3d85B	6	NWDVAY	1jrhH	12	RAPFYGNHAMDY	1etzB	16	RTFSYYYGSSFYYFDN
1f3dH	6	DYDGVY	2g5bB	12	DNGSDYRWYFDV	4grwE	16	VRTGWGLNAPDYAMDY
1bafH	6	GWPLAY	1kfaH	12	EGLLLDYFTMDY	4olvH	16	GKYCTARDYYNWDFEH
1sm3H	6	VGQFAY	3bgfB	12	EGIPQLLRTLDY	3lmjH	16	GGNYYNLWTGYYPLAY
1mimH	8	DYGYYFDF	3rvvD	12	TGVYRYPERAPY	2xqbH	16	DPAAWPLQQSLAWFDP
2pcpB	8	STWDDFDY	2vxvH	12	VGSGWYPYTFDI	3qycA	16	LPDLCPGDNCTYPDAS
1b4jH	8	GFLPWFAD	1mnuH	12	IYFDYADFIMDY	2x1pA	18	DLHRPYGPGTQRSDEYDS
1a6tB	8	RDDYYFDF	4rgmC	12	IYYGNNGGVMDY	4b5eA	18	REGDVGLVSYKRSSNYPY
3hr5B	8	DNWDAMDY	3eo9H	12	GIYFYGTTYFDY	3r0mA	18	KWRPLRYSDYPSNSDYYD
4o4yH	8	GLYNDYTV	3i9gH	12	GGFYGSTIWFDF	3grwH	18	TYGIYDLYVDYTEYVMDY
1a7qH	8	ERDYRLDY	2qqnH	14	GELPYYRMSKVMDV	1zvyA	18	TRKYVPVRFALDQSSYDY
3uo1H	8	LTNGYLDV	1yejH	14	WGFIPVREDYVMDY	4dkaA	18	DRFPTMEVVTIMTNEYDY
4lkxA	8	MGYDGLAY	4nbxB	14	APYGANWYRDEYAY	4i13B	18	SRISYRVWNTIPYNKLTL
1kelH	8	WGSYAMDY	2dquH	14	VSHYDGSRDWYFDV	2fx7H	18	EGTTGWGWLGKPIGAFAH
1rurH	10	AGGYYTGGDY	3rkdD	14	IKSVITTGDYALDY	1n0xH	18	VGPYSWDDSPQDNYYMDV
4ojfH	10	YDHYSGSSDY	2brrH	14	DYYGSTYPYYAMDY	2x1qA	18	SSRVFYTEVLQTTTGYDY
1ucbH	10	GLDDGAWFAY	1kxqE	14	GNSVRLASWEGYFY			
4gagH	10	ITTTTYAMDY	4grwH	16	VRTGWGLNAPDYAMDY			

Table A.7: Details of the H3 target set used to test Sphinx (Chapter 4).

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
2xknB	7	DSGGFAY	1ce1H	10	EGHTAAPFDY	3tnmA	13	GTRLRTLRNAFDI
2r29H	7	DYEGFAY	1cloH	10	DRGLRFYFDY	4lvhB	13	HGSPGYTLYAWDY
1yqvH	7	GNYDFDG	4h20H	10	RGLTGALFAY	1xiwD	13	SGYYGDSDWYFDV
1nj9B	7	GGGLFAF	4g7vH	10	SYSWSYAIDY	3ifoA	13	RPITPVLVDAMDY
3g5yB	7	AGRGFPY	3kykH	10	EGHNDWYFDL	1qfuH	13	FLQITTIIYGMDY
1kcrH	7	GGTGFTY	1q72H	10	VPQLGRGFAY	4edxB	13	GGYYYGTSYYFDY
2q76B	7	GDGFYVY	4ebqH	10	SNYRYDYFDV	2zchH	13	ADYGFNSGEAMDY
1mcoH	7	VPLVVNP	1lo0H	10	GDSFLVWFTF	3kr3H	13	DNGDYVGEKGFDI
3fzuC	7	LTTGSDS	3s96A	10	GGDNYLWFAY	3kdmB	13	DPGLWDYYYGMDV
3ietB	7	GKVRNAY	1qkzH	10	DPLEYYGMDY	3o2dH	13	EKDNYATGAWFAY
2hkfH	8	YGNHPFAY	4hfwB	11	GVVGATNEIDF	4k7pH	13	VVYYSASYWYFDV
1cicB	8	GLAFYFDH	3pgfH	11	TPWWYWSGLDY	3ojdB	15	ERDYGSRYRGYALDF
3e8uH	8	SHRFGLDY	1ehlH	11	RSGYKYYALDY	4m5yH	15	HVRSGYPDTAYYFDK
1mamH	8	DPYGPAAY	1ad9B	11	EKTTYYYAMDY	3gi9H	15	SPHYYGTTYNYPMDY
1uwgH	8	FYGNFFDY	4ioiB	11	WGGDGFYAMDY	3ujjH	15	LGFWGANRGGGGMDV
3o6kH	8	DYDYGVDY	1ejoH	11	RAFDSDVGFAS	3hzmB	15	DINPGSDGYYDALDY
1uweH	8	FYGNFFDY	1n4xH	11	DRHDYGEIFTY	3mlrH	15	LYLFEGAQSSNAFDL
1xgyH	8	TVSYVMDY	3vfgH	11	RGGHYGYALDY	1yedB	15	WGFTTVRENYYAMDY
4lqfH	8	QWGGAMDY	1ob1B	11	NYYRFDGGMDF	3ijhB	15	DISPSYGVYYEGFAY
2g60H	8	AAAAGADY	4cmhB	11	GDYYGSNSLDY	4hcrH	15	EGSSSSGDYYYGMDV
4dvbA	9	GELTYAMDY	1sbsH	12	GAYYRYDYAMDY	3u6rA	17	VVIPNAIRHTMGYYFDY
1um5H	9	SSMSDPGAN	1dsfH	12	SPIYYDYAPFTY	3npsB	17	TFHIRRYRSGYYDKMDH
4g3yH	9	NYYGSTYDY	4lmqE	12	DQGLNYGSLFDY	4hpoH	17	LGGRYYHDSSGYYYLDY
1nlbH	9	FLLRQYFDV	1j05B	12	FGYYVSDYAMAY	1dfbH	17	GRDYYDSGGYFTVAFDI
3clfH	9	GRLSLGFDY	4kaqH	12	STYYGGDWYFNV	3sobH	17	GSGHVNAVKNYGYVMDY
3v0wH	9	QGRGYTLDY	2r56H	12	AKRVGATGYFDL	1ikfH	17	HTLYDTLYGNYPVWFAD
1i9rH	9	SDGRNDMDS	1h0dB	12	LGDYGYAYTMDY	4hkxA	17	AGLEPRSVDYYFYGLDV
1fskC	9	GARDTWFAY	2r8sH	12	RAAGMSTYGFDY	2fb4H	17	DGGHGFCSSASCFGPDY
1mfaH	9	GGHGYYGDY	3ma9H	12	YFDTYNNYGFAN	3pnwB	17	TVRGSKKPYFSGWAMDY
2fatH	9	WGPHWYFDV	3g04B	12	LEPGYSSTWSVN	3mlwH	17	LHYSDRSGSYFNDVFHM

Table A.8: Details of the RosettAntibody benchmark set used to compare Sphinx to other methods (Chapters 4 and 5). Targets that have crystal contacts are highlighted with an asterisk.

PDB	Length	Sequence	PDB	Length	Sequence	PDB	Length	Sequence
2ddqH	4	YGGY	2aepH	9	VDYGTNYDY	2h1pH	11	RDSSASLYFDY
1dqqD	5	WGGDV	2fbjH*	9	LHYYGYNAY	1fptH*	11	DFYDYDVGFDY
1z3gH	6	GWDVAY	1igtB	9	HGGYYAMDY	2adgB*	11	HEDGNWNYFDY
1tetH	7	RSWYFDV	2fd6H	9	WGPHWYFDV	1igmH*	12	HRVSYVLTGFDS
1bqlH	7	GNYDFDG	2adfH	9	DNPYYALDY	2fjhH	12	WGHSTSPWAMDY
1cgsH	7	GYSSMDY	2jelH	9	VMGEQYFDV	2g5bH	12	DNGSDYRWYFDV
1mlbB	7	GDGNYGY	1yntD	9	SSTWYYFDY	2h2pC	12	LYYGYGYWYFDV
2c1pH	7	DDYDGAF	1dbaH	10	GDYVNWYFDV	1fbiH	13	LYYYGTSYGVLDY
2bdnH	8	GVFGFFDY	2b2xH	10	GFGDGGYFDV	1bj1H	14	YPHYYGSSHWYFDV
1fgnH*	8	DNSYYFDY	1clzH	10	GLDDGAWFAY	1wc7H	14	DKGSYGNYEAWFAY
1jptH	8	DTAAYFDY	2cjuH*	10	GYGYGAWFAY	1zanH*	14	DGGYSSSTLYAMDA
1a6tD	8	RDDYYFDF	1forH	10	SGNYPYAMDY	2aj3B*	14	YHRHFIRGPLSFDY
1kemH*	8	WGSYAMDY	1kb5H	10	SRTDLYYFDY	2dquH*	14	VSHYDGSRDWYFDV
1qblH	8	YDYGNFDY	2ajuH	10	YDYYGNTGDY	1f58H*	17	EEAMPYGNQAYYYAMDC
1vfaB	8	ERDYRLDY	1ztxH	10	SASYGDYADY	1hzhH*	18	VGPYSWDDSPQDNYYMDV
1iqdB	8	PDPDAFDI	1mcpH*	11	NYYGSTWYFDV	1g9mH*	19	VYEGEADEGEYRNNGFLKH
1k4cA	9	ERGDGYFAV	1ncaH	11	GEDNFGSLSDY	2b4cH	22	DFGPDWEDGDSYDGSGRGFFDF
1jhlH	9	DDNYGAMDY	2fjgH	11	FVFFLPYAMDY			

Table A.9: Details of the set of protein loops used to develop the general version of Sphinx (Chapters 4 and 5). Targets that have crystal contacts are highlighted with an asterisk.

PDB	Length	Start	End	Sequence	PDB	Length	Start	End	Sequence
1smrA*	6	113	118	MLAQFD	1vhqA	12	148	159	FDFPLRLTIGTD
4j3vA	6	313	318	AGVDDI	4gwiA*	12	81	92	HSVTHTNFEDNA
3u1lA	6	78	83	KGMCCL	3amnA	12	121	132	REKYQTEASIGD
2jc5A	6	150	155	WNIAHQ	4ag1A*	12	56	67	GAHNITEEEDTW
4l6dA	6	270	275	TSGVES	4mjkA	12	145	156	WKIHRIGSKESR
1trbA*	6	289	294	DHIYRQ	3w8kA*	14	200	213	YAYGYSKEDKPIRV
1q35A*	6	237	242	AKHSPN	4r3fA*	14	66	79	GDPTGTGHGGESIY
1vheA*	6	267	272	ASMVSH	1gp0A	14	1593	1606	KDGSPCPSKSGLSY
1s9rA	6	348	353	FDGTNY	2gdmA*	14	44	57	FSFLKGTSEVPQNN
1ofdA	6	218	223	AQPMRL	4p6bA	14	277	290	LLSGQTPEGETLLS
4x00A	8	175	182	LCFHRQPD	2q4wA	14	422	435	WDDRTSVVIPEEGE
3pvkA	8	176	183	NAKYSGSL	1kv9A	14	288	301	VTPGDSWDFTATQQ
1tkeA*	8	47	54	ACDLIEND	4wksC*	14	715	728	AFSQSSDPRSPHYR
3gveA	8	59	66	NGDLIQGN	3f41A*	14	153	166	FDYYRPVAETKPGL
1f46A	8	118	125	DDQRRMMT	2axqA*	14	321	334	LGLFSDAKITPRGN
2carA*	8	103	110	LLAGFEDK	2r16A*	16	857	872	GIITERRYLSSVPSNF
2ynaA	8	241	248	QFTEFVGT	4fe9A*	16	326	341	PQANWGNDFGFVDAIS
2bw4A	8	221	228	NGAVGALT	1ei5A*	16	139	154	RLKTTHFEPGSHYSYC
2z9wA	8	115	122	VPYNEAID	2ddbA*	16	18	33	RSVKPTARNMLQMKWN
1n08A*	8	85	92	NPYYKNKL	1rgzA*	16	112	127	YTAGGLPLQVPDEVTD
3khyA	10	165	174	KHNIRKYGAH	4j94A	16	133	148	GRPSPTDGFVGMAPDV
2cfcA	10	208	217	ARIPQKEIGT	1qcxA*	16	12	27	HGVTGGGSASPVYPTT
4ci7A*	10	480	489	YSEFSPCNFV	3r6aA*	16	26	41	LETPVAMRFEIPQTGV
2oemA	10	271	280	VTPSEFYGVA	1eokA*	16	127	142	IEHSGAKPNPIPTFPG
2driA	10	89	98	DRQATKGEVV	4lihA	16	453	468	NCYDEGGDMNFPFGGY
3kweA*	10	130	139	GAGCVVMMHV	4x84A*	18	86	103	GADESNERLELIKGGGAA
1pg4A	10	450	459	DNEGHPQEGA	2y8tA	18	127	144	QQPDRPPYRNNFLEDVPT
1nhsA*	10	362	371	YLMDFNPDKQ	1rp0A*	18	71	88	QSVSPGGGAWLGGQLFSA
4gekA	10	116	125	GDIRDIAIEN	4p3fA*	18	96	113	NRHKFTGKKVTEDLLTDN
3dciA	10	117	126	FIYKPREAVP	4cc2A*	18	1521	1538	VYTFKARNPNELSVSANQ
2ddxA*	12	155	166	VASYCPTPGGQL	2ocgA	18	78	95	DPRGYGHSRPPDRDFPAD
2nuwA	12	7	18	IITPFDKQGKVN	2octA*	18	32	49	NKKFPVFKAVSFKSQVVA
1eokA	12	170	181	TKYFGTTAPNNK	3t8wA	18	133	150	NNPGKENGPVKISSKVND
4i3gA	12	198	209	AANNQENNRFSV	4jicA*	18	268	285	EGATGGPRDFKQGDKPFD
4g29A*	12	294	305	ESMAGKVFVTSP	1v4aA	18	212	229	QPTQDGFVYRVDMRLRPF

Table A.10: Details of the set of protein loops that display high structural variation used to investigate loop flexibility (Chapter 6).

Loop	Length	Start	End	Sequence	Conformation	PDBs
1	10	408	417	MNHPTHKSLK	1	3IU1B
					2	3IU2A 3IWEA 3JTKA
2	10	109	118	HGDSAKRYNM	1	4HJ7A
					2	4HIDA 4HIKA 4HIMA 4HIOA 4HJ9A
3	10	41	50	HNLLRSGSEY	1	4KFRB
					2	4KFUA
4	10	65	74	QEEGSRSPAD	1	3WAEA
					2	3WAFB
5	10	1164	1173	PEPDELGNSG	1	3MP6A
					2	3MP8A
6	11	373	383	EDKELGSGNFG	1	1XBBA
					2	4FL1A
					3	4I0TA
					4	4FYOA 4I0SA
7	11	115	125	VSGAHLETGAS	1	3N4FA
					2	3QPED 3VCCA
8	11	171	181	DIQMDDPTDPA	1	2XFHA
					2	2JJNA 2JJOA
9	11	225	235	SRNTLAKFLPG	1	2Q9FA
					2	3MDMA
10	11	66	76	GRNSSVLGSVS	1	3IRSC
					2	3K4WL
11	12	1347	1358	ESNSRKTYIVHC	1	3AVRA
					2	3AVSA
12	12	251	262	KVIENMYGSREI	1	4FJNA
					2	4FJ7A 4FJ9A 4FJJA 4FJLA 4FK2A 4FK4A 4M3TA 4M3YA
13	12	466	477	LSWPDYGVPSSA	1	4ICZA
					2	2PA5A 4GE2A 4GE6B
14	12	79	90	GPSGSQIGDKES	1	4H31A
					2	4JHXA
					3	4KWTC
15	12	29	40	DVANGDRPDNYP	1	3AGNA
					2	3AGOA
					3	3AHWA
16	13	182	194	GITRTPLVHTFNS	1	1A4UB
					2	1B2LA
					3	1B16B 1SBYB
17	13	81	93	GRVRKAERWGPRT	1	3HSZA
					2	3HT0A
18	13	226	238	GAVKKASAPGQIS	1	3M7WD 3OEXB 4GFSA
					2	4GUIB 4GUJB
19	13	19	31	GSRMQGQLDTELS	1	4PZ9A
					2	4PZAB
20	13	179	191	ATSASLHNELPLN	1	3PHGA
					2	3PHHA
21	14	6	19	SYRCPCRFFESHVA	1	2J7ZB
					2	4UAIA
22	14	399	412	TDFEVGDVFQGVET	1	3V0HB
					2	3V0FB
					3	3V0DB 3V0GB
23	14	30	43	PYDEGCVRNGGRAG	1	4MXRB
					2	4MYFA 4MYKA 4MYLA
24	14	26	39	NHFVDECDPTIEDS	1	2CE2X
					2	2CL0X
					3	2CL6X
					4	2CL7X 2CLCX 2EVWX
25	14	192	205	TNMEPDHMDTYEGD	1	1GQYB
					2	1P31B 1P3DA
26	15	186	200	GYEGPGGAANNVFTN	1	2RBUX
					2	2RC2X 2RBWX
					3	2RBYX 2RBTX 2RBVX 2RC0X 2RBXX 2RBZX
27	15	149	163	ASGQLSVPQLPNFPG	1	1W4XA
					2	4C74A 4OVIA
28	15	276	290	EPPHVLISRPSAPFT	1	3OMQA
					2	2YJDB
29	15	80	94	QGRVRKAERWGPRTL	1	3ILLA
					2	3ILOA
30	15	84	98	HPPPPADAAGPVPLT	1	2XGUA
					2	2XGYA

Table A.11: Details of the set of protein loops that display low structural variation used to investigate loop flexibility (Chapter 6).

Loop	Length	Start	End	Sequence	Conformation	PDBs
1	10	16	25	TIAKPVEVVD	1	1N5NA 1S17A
2	10	428	437	LKQGQYSPMA	1	2CK3C 2JDIC
3	10	769	778	LQHRSFTMSN	1	2DQ6A 2DQMA 2ZXGA
4	10	197	206	SLPTPPLLES	1	1LG5A 1LGDA
5	10	109	118	LGIFNNSKQD	1	1AX0A 1AX1A 1AX2A 1AXYA 1AXZA
6	11	33	43	HKGSRLASIHS	1	1UKMB 1V7PB
7	11	112	122	GLEVSDAICNV	1	2CWKA 2DXDB 2DXEB 2DXFB 2DYAB
8	11	186	196	FEKHDDYKFNG	1	1FNBA 1FNDA
9	11	118	128	RMGAPEWDYIG	1	1H32A 1H33A
10	11	66	76	KGYAWTPGIYT	1	2GOUA 2GQ8A 2GQ9A 2GQAA
11	12	368	379	GITDPWYNPGDV	1	3UQYM 3USEM
12	12	22	33	RYLDILPRVRSD	1	3I9KB 3I9LB
13	12	38	49	SDAASQRMEPRA	1	3MRBA 3MRCA 3MRDA
14	12	92	103	RDLSSTPLIVDG	1	3LJKA 3M5PA 3Q7IA 3Q88A
15	12	322	333	VVEENYHDAPIV	1	3TANA 3TAPA 3TAQA 3TARA 4DQSA
16	13	319	331	GGPVDTFHAMRSS	1	3AFVA 3MM2A 3VXIA 3VXJA
17	13	276	288	RGETEGRQSDSQV	1	1U7HA 1X7DA
18	13	81	93	IASTRHGDDVDDK	1	2VYQA 2VZLA
19	13	199	211	GARSTGSNTWMDF	1	1Y3NA 1Y3QA 3A09A
20	13	117	129	QAPVSASALNANT	1	3P7YA 3P80A 3P81A
21	14	363	376	DRPQMEKDGLRDID	1	1FEHA 3C8YA
22	14	98	111	KTGKHVDANGKIYL	1	3P9GA 3P9HA
23	14	45	58	GTFDKGTKTGGPFG	1	2XI6A 2XIFA 2XIHA 2XJ6A
24	14	478	491	GIEPPVVRSEKDFD	1	2X93A 2X94A 2X95A
25	14	15	28	NNDTLTSTTGGSGI	1	2IXTB 3D43B
26	15	74	88	ATPADDERHSFKFKQ	1	3CI1A 3CI3A
27	15	328	342	TNIFILAESLGSVES	1	3QHXD 3QI6A
28	15	48	62	IAKKQPVTQQTLFEL	1	1PI4B 1PI5B 2FFYB
29	15	185	197	LWTGGKDTCGGDSGG	1	4NFEA 4NFFA
30	15	57	71	LNKPIVARDEKLKDY	1	1GQIB 1GQJB 1GQKB 1GQLA

Table A.12: Details of the 'inflexible loop' set from Benson and Daggett (2008) used in Chapter 6.

PDB	Length	Start	End	Sequence	PDB	Length	Start	End	Sequence
1opsA	6	17	22	LTPAMM	2hnpA	7	257	263	RMGLIQT
1risA	6	81	86	IRDNVR	1chuA	7	271	277	RPDGTRF
1dywA	6	131	136	GKHVVF	1e1qA	7	64	70	LNLEPDN
1fkbA	7	64	70	AQMSVGQ	1ubqA	7	16	22	EVEPSDT
1gprA	7	14	20	VFVSPIT	1opsA	10	34	43	FAEMSQLVGK
1e1qA	7	44	50	LRNVQAE	1opsA	10	50	59	AKGQTLMPNM
1gprA	7	151	157	NREQEDI	1fqnA	10	29	38	SPVDIDTHTA
1whiA	7	11	17	ADNSGAR	1ge8A	11	54	64	PSSIFSKYEVV
1vmoA	7	26	32	CHSGYAN	3fibA	15	215	229	FGHLSPTGTTEFWLG
1opsA	7	26	32	VTNPIGI	1whiA	16	88	103	RDDKSPRGTRIFGPVA

References

Abhinandan, K. R. and Martin, A. C. R. (2010). Analysis and prediction of VH/VL packing in antibodies. *Protein Eng. Des. Sel.*, **23**, 689–697.

Adams, J. A. (2003). Activation Loop Phosphorylation and Catalysis in Protein Kinases: Is There Functional Evidence for the Autoinhibitor Model? *Biochemistry*, **42**, 601–607.

Aggarwal, R. S. (2014). What's fueling the biotech engine-2012 to 2013. *Nat. Biotechnol.*, **32**, 32–9.

Almagro, J. C., Beavers, M. P., Hernandez-Guzman, F., Maier, J., Shaulsky, J., Butenhof, K., Labute, P., Thorsteinson, N., Kelly, K., Teplyakov, A., Luo, J., Sweet, R., and Gilliland, G. L. (2011). Antibody modeling assessment. *Proteins*, **79**, 3050–66.

Alzari, P. M., Lascombe, M. B., and Poljak, R. J. (1988). Three-Dimensional Structure of Antibodies. *Annu. Rev. Immunol.*, **6**, 555–580.

Amemiya, T., Koike, R., Kidera, A., and Ota, M. (2012). PSCDB: a database for protein structural change upon ligand binding. *Nucleic Acids Res.*, **40**, D554–D558.

Babor, M. and Kortemme, T. (2009). Multi-constraint Computational Design Suggests that Native Sequences of Germline Antibody H3 Loops are Nearly Optimal for Conformational Flexibility. *Proteins*, **75**, 846–858.

Baker, D. H. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, **294**, 93–96.

Beck, A., Wurch, T., Bailly, C., and Corvaia, N. (2010). Strategies and challenges for the next generation of therapeutic antibodies. *Nat. Rev. Immunol.*, **10**, 345–352.

Benson, N. C. and Daggett, V. (2008). Dynameomics : Large-scale assessment of native protein flexibility. *Protein Sci.*, **17**, 2038–2050.

Berenger, F., Shrestha, R., Zhou, Y., Simoncini, D., and Zhang, K. Y. J. (2012). Durandal: Fast exact clustering of protein decoys. *J. Comput. Chem.*, **33**, 471–474.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Bernstein, L. S., Ramineni, S., Hague, C., Cladman, W., Chidiac, P., Levey, A. I., and Hepler, J. R. (2004). RGS2 Binds Directly and Selectively to the M1 Muscarinic Acetylcholine Receptor Third Intracellular Loop to Modulate G q/11 Alpha Signaling. *J. Biol. Chem.*, **279**, 21248–21256.

Berrondo, M., Kaufmann, S., and Berrondo, M. (2014). Automated Aufbau of antibody structures from given sequences using Macromoltek's SmrtMolAntibody. *Proteins*, **82**, 1636–45.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, **4**, 187–217.

Brünger, A. T. (1997). Free R value: cross-validation in crystallography. *Methods Enzymol.*, **277**, 366–96.

Brunning, A. (2014). A brief guide to the twenty common amino acids. http://www.compoundchem.com/2014/09/16/aminoacids.

Bujotzek, A., Dunbar, J., Lipsmeier, F., Schäfer, W., Antes, I., Deane, C. M., and Georges, G. (2015). Prediction of VH–VL domain orientation for antibody variable domain modeling. *Proteins*, **83**, 681–695.

Burnet, F. (1957). A modification of Jerne's theory of antibody production using the concept of clonal selection. *Aust. J. Sci.*, **20**, 67–69.

Canutescu, A. A. and Dunbrack Jr., R. L. (2003). Cyclic coordinate descent : A robotics algorithm for protein loop closure. *Protein Sci.*, **12**, 963–972.

Canutescu, A. A., Shelenkov, A. A., and Dunbrack Jr., R. L. (2003). A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.*, **12**, 2001–2014.

Cao, R., Wang, Z., and Cheng, J. (2014). Designing and evaluating the MULTICOM protein local and global model quality prediction methods in the CASP10 experiment. *BMC Struct. Biol.*, **14**, 13.

Choi, Y. and Deane, C. M. (2010). FREAD revisited: Accurate loop structure prediction using a database search algorithm. *Proteins*, **78**, 1431–40.

Choi, Y. and Deane, C. M. (2011). Predicting antibody complementarity determining region structures without classification. *Mol. Biosyst.*, **7**, 3327–34.

Choi, Y., Agarwal, S., and Deane, C. M. (2013). How long is a piece of loop? *PeerJ*, **1**, 1–15.

Chothia, C. and Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J.*, **5**, 823–826.

Chothia, C. and Lesk, A. M. (1987). Canonical Structures for the Hypervariable Regions of Immunoglobulins. *J. Mol. Biol.*, **196**, 901–17.

Chys, P. and Chacon, P. (2013). Random Coordinate Descent with Spinor-matrices and Geometric Filters for Efficient Loop Closure. *J. Chem. Theory Comput.*, **9**, 1821–1829.

Clark, L. A., Boriack-Sjodin, P. A., Eldredge, J., Fitch, C., Friedman, B., Hanf, K. J. M., Jarpe, M., Liparoto, S. F., Li, Y., Lugovskoy, A., Miller, S., Rushe, M., Sherman, W., Simon, K., and Van Vlijmen, H. (2006). Affinity enhancement of an in vivo matured therapeutic antibody using structure-based computational design. *Protein Sci.*, **15**, 949–60.

Costantini, S., Colonna, G., and Facchiano, A. M. (2006). Amino acid propensities for secondary structures are influenced by the protein structural class. *Biochem. Bioph. Res. Co.*, **342**, 441–451.

Crommelin, D. J., Storm, G., Verrijk, R., de Leede, L., Jiskoot, W., and Hennink, W. E. (2003). Shifting paradigms: biopharmaceuticals versus low molecular weight drugs. *Int. J. Pharm.*, **266**, 3–16.

de Bakker, P. I. W., DePristo, M. A., Burke, D. F., and Blundell, T. L. (2003). Ab Initio Construction of Polypeptide Fragments : Accuracy of Loop Decoy Discrimination by an All-Atom Statistical Potential and the AMBER Force Field With the Generalized Born Solvation Model. *Proteins*, **51**, 21–40.

Deane, C. M. and Blundell, T. L. (2001). CODA: A combined algorithm for predicting the structurally variable regions of protein models. *Protein Sci.*, **10**, 599–612.

DePristo, M. a., De Bakker, P. I. W., Lovell, S. C., and Blundell, T. L. (2003). Ab initio construction of polypeptide fragments: efficient generation of accurate, representative ensembles. *Proteins*, **51**, 41–55.

Derreumaux, P. and Schlick, T. (1998). The Loop Opening/Closing Motion of the Enzyme Triosephosphate Isomerase. *Biophys. J.*, **74**, 72–81.

Diskin, R., Scheid, J. F., Marcovecchio, P. M., West Jr, A. P., Klein, F., Gao, H., Gnanapragasam, P. N. P., Abadir, A., Seaman, M., Nussenzweig, M. C., and Bjorkman, P. J. (2011). Increasing the Potency and Breadth of an HIV Antibody by Using Structure-Based Rational Design. *Science*, **334**, 1289–1294.

Dong, G. Q., Fan, H., Schneidman-Duhovny, D., Webb, B., and Sali, A. (2013). Optimized atomic statistical potentials: assessment of protein interfaces and loops. *Bioinformatics*, **29**, 3158–66.

Dunbar, J., Fuchs, A., Shi, J., and Deane, C. M. (2013). ABangle: Characterising the VH-VL orientation in antibodies. *Protein Eng. Des. Sel.*, **26**, 611–620.

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. (2014). SAbDab: the structural antibody database. *Nucleic Acids Res.*, **42**, D1140–6.

Echols, N., Milburn, D., and Gerstein, M. (2003). MolMovDB: analysis and visualization of conformational change and structural flexibility. *Nucleic Acids Res.*, **31**, 478–482.

EMBL-EBI (2016). Uniprot: Current release statistics. http://www.ebi.ac.uk/uniprot/TrEMBLstats.

Engh, R. A. and Huber, R. (1991). Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallogr. Sect. A*, **47**, 392–400.

Fasnacht, M., Butenhof, K., Goupil-Lamy, A., Hernandez-Guzman, F., Huang, H., and Yan, L. (2014). Automated antibody structure prediction using Accelrys tools: results and best practices. *Proteins*, **82**, 1583–98.

Feeney, A. J., Tang, A., and Ogwaro, K. M. (2000). B-cell repertoire formation: role of the recombination signal sequence in non-random V segment utilization. *Immunol. Rev.*, **175**, 59–69.

Finlay, W. J. J. and Almagro, J. C. (2012). Natural and man-made V-gene repertoires for antibody discovery. *Front. Immunol.*, **3**, 1–18.

Fiser, A. (2009). Comparative protein structure modelling. In D. J. Rigden, editor, *From Protein Structure to Function with Bioinformatics*, chapter 3. Springer.

Fiser, A. and Sali, A. (2003). ModLoop: Automated modeling of loops in protein structures. *Bioinformatics*, **19**, 2500–2501.

Fiser, A., Kinh, R., Do, G., and Sali, A. (2000). Modeling of loops in protein structures. *Protein Sci.*, **9**, 1753–1773.

Gavenonis, J., Sheneman, B. A., Siegert, T. R., Eshelman, M. R., and Kritzer, J. A. (2015). Comprehensive analysis of loops at protein-protein interfaces for macrocycle design. *Nat. Chem. Biol.*, **10**, 716–722.

Glanville, J., Zhai, W., Berka, J., Telman, D., Huerta, G., Mehta, G. R., Ni, I., Mei, L., Sundar, P. D., Day, G. M. R., Cox, D., Rajpal, A., and Pons, J. (2009). Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *P. Natl. Acad. Sci. USA*, **106**, 20216–21.

Go, N. and Scheraga, H. A. (1970). Ring Closure and Local Conformational Deformations of Chain Molecules. *Macromolecules*, **3**, 178–187.

Gonzalez, S. F., Degn, S. E., Pitcher, L. A., Woodruff, M., Heesters, B. A., and Carroll, M. C. (2011). Trafficking of B Cell Antigen in Lymph Nodes. *Annu. Rev. Immunol.*, **29**, 215–33.

Gront, D., Kmiecik, S., Blaszczyk, M., Ekonomiuk, D., and Koli??ski, A. (2012). Optimization of protein models. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **2**, 479–493.

Harder, T., Borg, M., Boomsma, W., Røgen, P., and Hamelryck, T. (2012). Fast large-scale clustering of protein structures using gauss integrals. *Bioinformatics*, **28**, 510–515.

Hay, M., Thomas, D. W., Craighead, J. L., Economides, C., and Rosenthal, J. (2014). Clinical development success rates for investigational drugs. *Nat. Biotechnol.*, **32**, 40–51.

Hildebrand, P. W., Goede, A., Bauer, R. A., Gruening, B., Ismer, J., Michalsky, E., and Preissner, R. (2009). SuperLooper - a prediction server for the modeling of loops in globular and membrane proteins. *Nucleic Acids Res.*, **37**, W571–4.

Hill, J. R., Kelm, S., Shi, J., and Deane, C. M. (2011). Environment specific substitution tables improve membrane protein alignment. *Bioinformatics*, **27**, i15–23.

Holtby, D., Li, S. C., and Li, M. (2013). LoopWeaver: loop modeling by the weighted scaling of verified proteins. *J. Comput. Biol.*, **20**, 212–23.

Ilari, A. and Savino, C. (2008). Protein Structure Determination by X-Ray Crystallography. In J. M. Keith, editor, *Bioinformatics Volume I: Data, Sequence Analysis and Evolution*, chapter 3. Humana Press.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, **431**, 931–945.

Jacobson, M. P., Pincus, D. L., Rapp, C. S., Day, T. J. F., Honig, B., Shaw, D. E., and Friesner, R. A. (2004). A hierarchical approach to all-atom protein loop prediction. *Proteins*, **55**, 351–367.

James, L. C. and Tawfik, D. S. (2003). Conformational diversity and protein evolution - A 60-year-old hypothesis revisited. *Trends. Biochem. Sci.*, **28**, 361–368.

Janeway, C. A., Travers, P., Walport, M., and Shlomchik, M. J. (2001). *Immunobiology: the immune system in health and disease*. Garland Science, fifth edition edition.

Jiang, X.-R., Song, A., Bergelson, S., Arroll, T., Parekh, B., May, K., Chung, S., Strouse, R., and Mire-sluis, A. (2011). Advances in the assessment and control of the effector functions of therapeutic antibodies. *Nat. Rev. Drug Discov.*, **10**, 101–110.

Joosten, R. P., te Beek, T. a. H., Krieger, E., Hekkelman, M. L., Hooft, R. W. W., Schneider, R., Sander, C., and Vriend, G. (2011). A series of PDB related databases for everyday needs. *Nucleic Acids Res.*, **39**, D411–9.

Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.*, **7863**, 11225–11236.

Juritz, E. I., Alberti, S. F., and Parisi, G. D. (2011). PCDB: a database of protein conformational diversity. *Nucleic Acids Res.*, **39**, D475–D479.

Kabat, E., Wu, T., Bilofsky, H., Reid-Miller, M., and Perry, H. (1983). *Sequences of Proteins of Immunological Interest*. National Institutes of Health, Bethesda.

Kelm, S., Vangone, A., Choi, Y., Ebejer, J.-P., Shi, J., and Deane, C. M. (2013). Fragment-based modelling of membrane protein loops - successes , failures and prospects for the future. *Proteins*.

Kiyoshi, M., Caaveiro, J. M. M., Miura, E., Nagatoishi, S., Nakakido, M., Soga, S., Shirai, H., Kawabata, S., and Tsumoto, K. (2014). Affinity improvement of a therapeutic antibody by structure-based computational design: Generation of electrostatic interactions in the transition state stabilizes the antibody-antigen complex. *PLoS One*, **9**, 1–9.

Köhler, G. and Milstein, C. (1975). Continuous cultures of fused cells secreting antibody of predefined specificity. *Nature*, **256**, 495–497.

Kosloff, M. and Kolodny, R. (2008). Sequence-similar, structure-dissimilar protein pairs in the PDB. *Proteins*, **71**, 891–902.

Krauss, I. R., Merlino, A., Vergara, A., and Sica, F. (2013). An Overview of Biological Macromolecule Crystallization. *Int. J. Mol. Sci.*, **14**, 11643–11691.

Krawczyk, K., Baker, T., Shi, J., and Deane, C. M. (2013). Antibody i-Patch prediction of the antibody binding site improves rigid local antibody-antigen docking. *Protein Eng. Des. Sel.*, **26**, 621–9.

Krivov, G. G., Shapovalov, M. V., and Dunbrack, R. L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. *Proteins*, **77**, 778–795.

Kryshtafovych, A., Barbato, A., Fidelis, K., Monastyrskyy, B., Schwede, T., and Tramontano, A. (2014). Assessment of the assessment: Evaluation of the model quality estimates in CASP10. *Proteins*, **82**, 112–126.

Kunik, V. and Ofran, Y. (2013). The indistinguishability of epitopes from protein surface is explained by the distinct binding preferences of each of the six antigen-binding loops. *Protein Eng. Des. Sel.*, **26**, 599–609.

Kuroda, D., Shirai, H., Jacobson, M. P., and Nakamura, H. (2012). Computer-aided antibody design. *Protein Eng. Des. Sel.*, **25**, 507–21.

Larsson, P., Wallner, B., Lindahl, E., and Elofsson, A. (2008). Using multiple templates to improve quality of homology models in automated homology modeling. *Protein Sci.*, **17**, 990–1002.

Laskowski, R. A., MacArthur, M. W., Moss, D. S., and Thornton, J. M. (1993). PROCHECK - a program to check the stereochemical quality of protein structures. *J. App. Cryst.*, **26**, 283–291.

Lazaridis, T. and Karplus, M. (2000). Effective energy functions for protein structure prediction. *Curr. Opin. Struc. Biol.*, **10**, 139–145.

Leavy, O. (2010). Therapeutic antibodies: past, present and future. *Nat. Rev. Immunol.*, **10**, 297.

Lee, J., Wu, S., and Zhang, Y. (2009). Ab initio protein structure prediction. In D. J. Rigden, editor, *From Protein Structure to Function with Bioinformatics*, chapter 1. Springer.

Lee, M., Lloyd, P., Zhang, X., Schallhorn, J. M., Sugimoto, K., Leach, A. G., Sapiro, G., and Houk, K. N. (2006). Shapes of antibody binding sites: qualitative and quantitative analyses based on a geomorphic classification scheme. *J. Org. Chem.*, **71**, 5082–92.

Leem, J., Dunbar, J., Georges, G., Shi, J., and Deane, C. M. (2016). ABodyBuilder: automated antibody structure prediction with data-driven accuracy estimation. *mAbs*, **0862**, 0.

Lefranc, M.-P., Pommié, C., Ruiz, M., Giudicelli, V., Foulquier, E., Truong, L., Thouvenin-Contet, V., and Lefranc, G. (2003). IMGT unique numbering fro immunoglobulin and T cell receptor variable domains and Ig superfamily V-like domains. *Dev. Comp. Immunol.*, **27**, 55–77.

Lewis, S. M., Wu, X., Pustilnik, A., Sereno, A., Huang, F., Rick, H. L., Guntas, G., Leaver-Fay, A., Smith, E. M., Ho, C., Hansen-Estruch, C., Chamberlain, A. K., Truhlar, S. M., Conner, E. M., Atwell, S., Kuhlman, B., and Demarest, S. J. (2014). Generation of bispecific IgG antibodies by structure-based design of an orthogonal Fab interface. *Nat. Biotechnol.*, **32**, 191–8.

Li, H. and Zhou, Y. (2005). SCUD: Fast structure clustering of decoys using reference state to remove overall rotation. *J. Comput. Chem.*, **26**, 1189–1192.

Li, S. C. and Ng, Y. K. (2010). Calibur : a tool for clustering large numbers of protein decoys. *BMC Bioinform.*, **11**, 1–12.

Li, S. C., Bu, D., and Li, M. (2012). Clustering 100,000 protein structure decoys in minutes. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **9**, 765–773.

Liang, S., Zhang, C., and Zhou, Y. (2014). LEAP: Highly accurate prediction of protein loop conformations by integrating coarse-grained sampling and optimized energy scores with all-atom refinement of backbone and side chains. *J. Comput. Chem.*, **35**, 335–341.

Lippow, S. M., Wittrup, K. D., and Tidor, B. (2007). Computational design of antibody-affinity improvement beyond in vivo maturation. *Nat. Biotechnol.*, **25**, 1171–6.

MacCallum, R. M., Martin, A. C., and Thornton, J. M. (1996). Antibody-antigen interactions: contact analysis and binding site topography. *J. Mol. Biol.*, **262**, 732–45.

Maier, J. K. X. and Labute, P. (2014). Assessment of fully automated antibody homology modeling protocols in molecular operating environment. *Proteins*, **82**, 1599–610.

Mandal, C., Kingery, B. D., Anchin, J. M., Subramaniam, S., and Linthicum, S. D. (1996). ABGEN: a knowledge-based automated approach for antibody structure modeling. *Nat. Biotechnol.*, **14**, 323–328.

Mandell, D. J., Coutsias, E. A., and Kortemme, T. (2009). Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nat. Methods*, **6**, 551–2.

Marcatili, P., Rosi, A., and Tramontano, A. (2008). PIGS: automatic prediction of antibody structures. *Bioinformatics*, **24**, 1953–4.

Market, E. and Papavasiliou, F. N. (2003). V(D)J recombination and the evolution of the adaptive immune system. *PLoS Biol.*, **1**, 24–27.

Martí-Renom, A. M., Mas, M. J., Aloy, P., Querol, E., Avilés, X. F., and Oliva, B. (1998). Statistical analysis of the loop-geometry on a non-redundant database of proteins. *J. Mol. Model*, **4**, 347–354.

Martin, A. C., Cheetham, J. C., and Rees, A. R. (1989). Modeling antibody hypervariable loops: a combined algorithm. *P. Natl. Acad. Sci. USA*, **86**, 9268–9272.

Martin, A. C., MacArthur, M. W., and Thornton, J. M. (1997). Assessment of comparative modeling in CASP2. *Proteins*, **Suppl. 1**, 14–28.

Martin, A. C. R. and Thornton, J. M. (1996). Structural families of loops in homologous proteins: Automatic classification, modelling and application to antibodies. *J Mol. Biol.*, **263**, 800–815.

McCafferty, J., Griffiths, A. D., Winter, G., and Chiswell, D. J. (1990). Phage antibodies: filamentous phage displaying antibody variable domains. *Nature*, **348**, 552–554.

Messih, M. A., Lepore, R., Marcatili, P., and Tramontano, A. (2014). Improving the accuracy of the structure prediction of the third hypervariable loop of the heavy chains of antibodies. *Bioinformatics*, **30**, 2733–2740.

Messih, M. A., Lepore, R., and Tramontano, A. (2015). LoopIng: A template-based tool for predicting the structure of protein loops. *Bioinformatics*, **31**, 3767–3772.

Miao, Z., Cao, Y., and Jiang, T. (2011). RASP: rapid modeling of protein side chain conformations. *Bioinformatics*, **27**, 3117–3122.

Michalsky, E., Goede, A., and Preissner, R. (2003). Loops In Proteins (LIP) - a comprehensive loop database for homology modelling. *Protein Eng.*, **16**, 979–85.

Monzon, A. M., Juritz, E., Fornasari, S., and Parisi, G. (2013). CoDNaS: a database of conformational diversity in the native state of proteins. *Bioinformatics*, **29**, 2512–2514.

Morea, V., Tramontano, A., Rustici, M., Chothia, C., and Lesk, A. M. (1998). Conformations of the third hypervariable region in the VH domain of immunoglobulins. *J. Mol. Biol.*, **275**, 269–294.

Moult, J., Pedersen, J. T., Judson, R., and Fidelis, K. (1995). A Large-Scale Experiment to Assess Protein Structure Prediction Methods. *Proteins*, **23**, ii–iv.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Genetics, M., Pasteur, I., and Bolognetti, F. C. (2014). Critical assessment of methods of protein structure prediction (CASP). *Proteins*, **82**, 1–6.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 433–53.

North, B., Lehmann, A., and Dunbrack Jr., R. L. (2011). A new clustering of antibody CDR loop conformations. *J. Mol. Biol.*, **406**, 228–56.

Nowak, J., Baker, T., Georges, G., Kelm, S., Klostermann, S., Shi, J., Sridharan, S., and Deane, C. M. (2016). Length-independent structural similarities enrich the antibody CDR canonical class model. *mAbs*, **8**, 751–60.

Odegard, V. H. and Schatz, D. G. (2006). Targeting of somatic hypermutation. *Nat. Rev. Immunol.*, **6**, 573–83.

O'Meara, M. J., Leaver-Fay, A., Tyka, M. D., Stein, A., Houlihan, K., Dimaio, F., Bradley, P., Kortemme, T., Baker, D., Snoeyink, J., and Kuhlman, B. (2015). Combined covalent-electrostatic model of hydrogen bonding improves structure prediction with Rosetta. *J. Chem. Theory Comput.*, **11**, 609–622.

Panchenko, A. R. and Madej, T. (2005). Structural similarity of loops in protein families: toward the understanding of protein evolution. *BMC Evol. Biol.*, **5**, 10.

Parsons, J., Holmes, J. B., Rojas, J. M., Tsai, J., and Strauss, C. E. M. (2005). Practical conversion from torsion space to Cartesian space for in silico protein synthesis. *J. Comput. Chem.*, **26**, 1063–8.

Parthasarathy, S. and Murthy, M. R. N. (1997). Analysis of temperature factor distribution in high-resolution protein structures. *Protein Sci.*, **6**, 2561–2567.

PDB-101 (2016). R-value and R-free. http://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/r-value-and-r-free.

Pedotti, M., Simonelli, L., Livoti, E., and Varani, L. (2011). Computational docking of antibody-antigen complexes, opportunities and pitfalls illustrated by influenza hemagglutinin. *Int. J. Mol. Sci.*, **12**, 226–251.

Petoukhov, M. V., Eady, N. A. J., Brown, K. A., and Svergun, D. I. (2002). Addition of missing loops and domains to protein models by x-ray solution scattering. *Biophys. J.*, **83**, 3113–25.

Pineda, J. R. E. T., Callender, R., and Schwartz, S. D. (2007). Ligand Binding and Protein Dynamics in Lactate Dehydrogenase. *Biophys. J.*, **93**, 1474–1483.

Pokkuluri, P. R., Yang, X., Londer, Y. Y., and Schiffer, M. (2012). Pitfalls in the interpretation of structural changes in mutant proteins from crystal structures. *J Struct. Funct. Genomics*, **13**, 227–232.

Pommerville, J. (2012). *Alcamo's Fundamentals of Microbiology: Body Systems Edition*. Jones and Bartlett.

Ponder, J. W. and Case, D. A. (2003). Force fields for protein simulations. *Adv. Protein Chem.*, **66**, 27–85.

Qiu, L., Gulotta, M., and Callender, R. (2007). Lactate Dehydrogenase Undergoes a Substantial Structural Change to Bind its Substrate. *Biophys. J.*, **93**, 1677–1686.

Regad, L., Martin, J., Nuel, G., and Camproux, A.-C. (2010). Mining protein loops using a structural alphabet and statistical exceptionality. *BMC Bioinform.*, **11**, 75.

Ren, J., Liu, Q., Ellis, J., and Li, J. (2014). Tertiary structure-based prediction of conformational B-cell epitopes through B factors. *Bioinformatics*, **30**, 264–273.

Rohl, C. A., Strauss, C. E. M., Chivian, D., and Baker, D. (2004). Modeling structurally variable regions in homologous proteins with rosetta. *Proteins*, **55**, 656–77.

Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Eng.*, **12**, 85–94.

Rouet, R., Lowe, D., and Christ, D. (2014). Stability engineering of the human antibody repertoire. *FEBS Lett.*, **588**, 269–277.

Samudrala, R. and Moult, J. (1998). An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J. Mol. Biol.*, **275**, 895–916.

Saper, C. B. (2009). A Guide to the Perplexed on the Specificity of Antibodies. *J. Histochem. Cytochem.*, **57**, 1–5.

Saraste, M., Sibbald, P. R., and Wittinghofer, A. (1990). The P-loop - a common motif in ATP- and GTP-binding proteins. *Trends Biochem. Sci.*, **15**, 430–434.

Schaefer, C. and Rost, B. (2012). Predict impact of single amino acid change upon protein structure. *BMC Genom.*, **13 Suppl 4**, S4.

Schrödinger (2010). The PyMOL Molecular Graphics System, Version 1.7.4.

Schroeder Jr, H. W. and Cavacini, L. (2010). Structure and Function of Immunoglobulins. *J. Allergy Clin. Immunol.*, **125**, S41–S52.

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. (2005). The FoldX web server: An online force field. *Nucleic Acids Res.*, **33**, 382–388.

Scott, A. M., Wolchok, J. D., and Old, L. J. (2012). Antibody therapy of cancer. *Nat. Rev. Cancer*, **12**, 278–287.

Sela-Culang, I., Alon, S., Ofran, Y., and Alerts, E. (2012). A Systematic Comparison of Free and Bound Antibodies Reveals Binding-Related Conformational Changes. *J. Immunol.*, **189**, 4890–9.

Sela-Culang, I., Kunik, V., and Ofran, Y. (2013). The structural basis of antibody-antigen recognition. *Front. Immunol.*, **4**, 1–13.

Sethi, D. K., Agarwal, A., Manivel, V., Rao, K. V. S., and Salunke, D. M. (2006). Differential Epitope Positioning within the Germline Antibody Paratope Enhances Promiscuity in the Primary Immune Response. *Immunity*, **24**, 429–438.

Shen, M.-Y. and Sali, A. (2006). Statistical potential for assessment and prediction of protein structures. *Protein Sci.*, **15**, 2507–2524.

Shirai, H., Kidera, A., and Nakamura, H. (1996). Structural classification of CDR-H3 in antibodies. *FEBS Lett.*, **399**, 1–8.

Shirai, H., Ikeda, K., Yamashita, K., Tsuchiya, Y., Sarmiento, J., Liang, S., Morokata, T., Mizuguchi, K., Higo, J., Standley, D. M., and Nakamura, H. (2014). High-resolution modeling of antibody structures by a combination of bioinformatics, expert knowledge, and molecular simulations. *Proteins*, **82**, 1624–1635.

Shortle, D., Simons, K. T., and Baker, D. (1998). Clustering of low-energy conformations near the native structures of small proteins. *P. Natl. Acad. Sci. USA*, **95**, 11158–11162.

Sikic, K., Tomic, S., and Carugo, O. (2010). Systematic comparison of crystal and NMR protein structures deposited in the protein data bank. *Open Biochem. J.*, **4**, 83–95.

Sirin, S., Apgar, J. R., Bennett, E. M., and Keating, A. E. (2015). AB-Bind: Antibody binding mutational database for computational affinity predictions. *Protein Sci.*, **22**, 393–409.

Sivasubramanian, A., Sircar, A., Chaudhury, S., and Gray, J. J. (2009). Toward high-resolution homology modeling of antibody Fv regions and application to antibody-antigen docking. *Proteins*, **74**, 497–514.

Sokal, R. and Michener, C. (1958). A statistical method for evaluating systematic relationships. *Kans. Univ. Sci. Bull.*, pages 1409–1438.

Soto, C. S., Fasnacht, M., Zhu, J., Forrest, L., and Honig, B. (2007). Loop modeling: Sampling, filtering, and scoring. *Proteins*, **70**, 834–843.

Spadiut, O., Tan, T. C., Pisanelli, I., Haltrich, D., and Divne, C. (2010). Importance of the gating segment in the substrate-recognition loop of pyranose 2-oxidase. *FEBS J.*, **277**, 2892–2909.

Stanfield, R. L., Zemla, A., Wilson, I. A., and Rupp, B. (2006). Antibody Elbow Angles are Influenced by their Light Chain Class. *J. Mol. Biol.*, **357**, 1566–1574.

Stein, A. and Kortemme, T. (2013). Improvements to robotics-inspired conformational sampling in Rosetta. *PloS One*, **8**, e63090.

Tai, C.-H., Bai, H., Taylor, T. J., and Lee, B. (2014). Assessment of template-free modeling in CASP10 and ROLL. *Proteins*, **82**, 57–83.

Tainer, J. A., Thayer, M. M., and Cunningham, R. P. (1995). DNA repair proteins. *Curr. Opin. Struc. Biol.*, **5**, 20–26.

Teplyakov, A., Luo, J., Obmolova, G., Malia, T. J., Sweet, R., Stanfield, R. L., Kodangattil, S., Almagro, J. C., and Gilliland, G. L. (2014). Antibody modeling assessment II. Structures and models. *Proteins*, **82**, 1563–82.

Thakkar, S., Nanaware-Kharade, N., Celikel, R., Peterson, E. C., and Varughese, K. I. (2014). Affinity improvement of a therapeutic antibody to methamphetamine and amphetamine through structure-based antibody engineering. *Sci. Rep.*, **4**, 3673.

The Protein Data Bank (2016). PDB current holdings breakdown. http://www.rcsb.org/pdb/statistics/holdings.do.

Tonegawa, S. (1983). Somatic generation of antibody diversity. *Nature*, **302**, 575–581.

Totrov, M. (2012). Loop simulations. In *Homology Modeling: Methods and Protocols (Methods in Molecular Biology)*, chapter 9. Humana Press.

Tsai, J., Taylor, R., Chothia, C., and Gerstein, M. (1999). The Packing Density in Proteins: Standard Radii and Volumes. *J Mol. Biol.*, **290**, 253–266.

Wang, F., Ekiert, D. C., Ahmad, I., Yu, W., Yong, Z., Bazirgan, A. T., Raudsepp, T., Mwangi, W., Criscitiello, M. F., Wilson, I. A., Schultz, P. G., and Smider, V. V. (2013). Reshaping Antibody Diversity. *Cell*, **153**, 1379–1393.

Wang, G. and Dunbrack, R. L. (2003). PISCES: a protein sequence culling server. *Bioinformatics*, **19**, 1589–1591.

Weiner, S. J., Kollman, P. A., Case, D. A., Singh, U. C., Ghio, C., Alagona, G., Profeta, S., and Weinerl, P. (1984). A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins. *J. Am. Chem. Soc.*, **106**, 765–784.

Weitzner, B. D., Kuroda, D., Marze, N., Xu, J., and Gray, J. J. (2014). Blind prediction performance of RosettaAntibody 3.0: grafting, relaxation, kinematic loop modeling, and full CDR optimization. *Proteins*, **82**, 1611–23.

Weitzner, B. D., Dunbrack, R. L., and Gray, J. J. (2015). The Origin of CDR H3 Structural Diversity. *Structure*, **23**, 302–311.

Whitelegg, N. R. J. and Rees, A. R. (2000). WAM: an improved algorithm for modelling antibodies on the WEB. *Protein Eng. Des. Sel.*, **13**, 819–24.

Willis, J. R., Briney, B. S., DeLuca, S. L., Crowe, J. E., and Meiler, J. (2013). Human Germline Antibody Gene Segments Encode Polyspecific Antibodies. *PLoS Comput. Biol.*, **9**.

Wlodawer, A., Minor, W., Dauter, Z., and Jaskolski, M. (2008). Protein crystallography for non-crystallographers, or how to get the best (but not more) from published macromolecular structures. *FEBS J.*, **275**, 1–21.

Xiang, Z., Soto, C. S., and Honig, B. (2002). Evaluating conformational free energies: the colony energy and its application to the problem of loop prediction. *P. Natl. Acad. Sci. USA*, **99**, 7432–7437.

Xu, J. and Berger, B. (2006). Fast and accurate algorithms for protein side-chain packing. *J. Am. Chem. Soc.*, **53**, 533–557.

Yamashita, K., Ikeda, K., Amada, K., Liang, S., Tsuchiya, Y., Nakamura, H., Shirai, H., and Standley, D. M. (2014). Kotai Antibody Builder: Automated high-resolution structural modeling of antibodies. *Bioinformatics*, **30**, 3279–3280.

Yang, Y. and Zhou, Y. (2008). Specific interactions for an initio folding of protein terminal regions with secondary structures (dDFIRE). *Proteins*, **72**, 793–803.

Yu, H. (1999). Extending the size limit of protein nuclear magnetic resonance. *Proc. Natl. Acad. Sci. USA*, **96**, 332–334.

Zhang, C., Liu, S., and Zhou, Y. (2004). Accurate and efficient loop selections by the dfire-based all-atom statistical potential. *Protein Sci.*, **13**, 391–399.

Zhang, J. and Zhang, Y. (2010). A Novel Side-Chain Orientation Dependent Potential Derived from Random-Walk Reference State for Protein Fold Selection and Structure Prediction. *PLoS One*, **5**, e15386.

Zhang, Y. and Skolnick, J. (2004a). Scoring Function for Automated Assessment of Protein Structure Template Quality. *Proteins*, **710**, 702–710.

Zhang, Y. and Skolnick, J. (2004b). SPICKER: A clustering approach to identify near-native protein folds. *J. Comput. Chem.*, **25**, 865–871.

Zhou, H. and Skolnick, J. (2011). GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction. *Biophys. J.*, **101**, 2043–2052.

Zhu, K., Pincus, D. L., Zhao, S., and Friesner, R. A. (2006). Long loop prediction using the protein local optimization program. *Proteins*, **65**, 438–452.

Zhu, K., Day, T., Warshaviak, D., Murrett, C., Friesner, R., and Pearlman, D. (2014). Antibody Structure Determination Using a Combination of Homology Modeling, Energy-Based Refinement and Loop Prediction. *Proteins*, **82**, 1646–1655.