Fold recognition and alignment in the 'twilight zone'

Jamie Hill

Department of Statistics

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2013

Acknowledgements

During the last four years I have enjoyed the friendship and support of many wonderful people. Thank you all.

This thesis would have been a great deal more lurid without the artistic direction of my supervisor, Charlotte. She has been a constant source of encouragement, humour, and good advice.

Specific thanks are due to my examiners Willie and Gil; collaborators Seb, JP, and Saulo; the ever-sarcastic Kat; confidante and repressed Slytherin, Hannah; and dragon-wrangler Emma (also songstress).

Finally, I owe a debt of gratitude to Hippogryph and Sabine, who, in their different ways, made the last year so pleasant and unpredictable!

Abstract

At present, the most accurate approach to predicting protein structure, comparative modelling, builds a model of a target sequence using known protein structures as templates. Comparative modelling becomes markedly less accurate in the 'twilight zone', where the target protein shares little sequence identity with all known templates. There are two main causes of this inaccuracy: first, it becomes difficult to identify good structural templates; second, it becomes difficult to determine which amino acids in the template are structurally equivalent to those in the target. These are problems of fold recognition and target-template alignment respectively. In this thesis, new approaches are developed to address both these problems.

The alignment problem is investigated in the special case of membrane proteins. These are key modelling targets as they resist structure determination and are pharmaceutically important. The approach taken here is to use 'environment specific substitution tables' (ESSTs)– that is, to alter the alignment scoring system for each local environment of the template structure. We show how ESSTs can be made for membrane proteins, tested for robustness of construction, and used to infer the most important evolutionary pressures acting on protein structure. The incorporation of ESSTs into a multiple sequence alignment method leads to more accurate alignments of membrane proteins, and so to more accurate models.

Recently, algorithms have been developed that predict contacts in protein structures from a multiple sequence alignment of homologous sequences. We explore the potential of these predictions for fold recognition by developing an algorithm that makes no use of amino acid identity, and so should be agnostic to the existence of a 'twilight zone' . We show that whilst this is not the case, our method is complementary to state-of-the-art approaches.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

Nature has given rise to proteins which, singly or in complex, perform diverse functions: from enzymes that catalyse chemical reactions, to channels that control the molecules that can pass through biological membranes, to photosynthetic centres that harness the energy in light, through toxins, antifreezes, scaffolds, and motors.

The sequence of a protein determines the structure it will assume in a given environment, and the structure of a protein is the primary determinant of its function. An understanding of the connection between structure and function is necessary to investigate the molecular basis

of disease, or to design synthetic proteins for commercial uses. However, for the majority of proteins no structure is known, and the structure must be inferred from the sequence. The connection between sequence and structure is the subject of this thesis.

This thesis concerns comparative modelling (also called 'homology modelling') in the difficult 'twilight zone' of sequence identity [Vogt *et al.*, 1995]. At present comparative modelling is the most accurate way of predicting the structure of a protein. It is based on the assumption that a protein of unknown structure (the 'target') will be structurally similar to a protein possessing a similar sequence (the 'template'). Comparative modelling can be divided into three stages (Figure 1.1): fold recognition (identifying a good template), alignment (identifying structurally equivalent residues in the target and template), and coordinate generation (building of the model, which in its simplest form consists of copying the coordinates of the aligned residues from the template to the target).[1] Models built by comparative modelling become less accurate as the sequence identity between the target and template decreases. In particular, within the twilight zone ($< 30\%$ sequence identity) the sequence identity between the target and template is a poor guide to their structural similarity.

Although the structures of the target and template diverge with decreasing sequence identity, this is not the principal cause of the decreasing accuracy of the models. Instead, fold recognition and alignment techniques decrease in power at lower sequence identity. This thesis seeks to improve these facets of comparative modelling: Chapters 3 and 4 focus on improving alignment for the special case of membrane proteins, and Chapter 5 describes a new approach to fold recognition for both soluble and membrane proteins.

Experimental information on membrane proteins is difficult to obtain, making prediction of their structure especially important, and meaning that in many cases the best template for a given sequence falls within the twilight zone of sequence identity. Chapter 3 discusses how the local environment of membrane proteins affects the substitution preferences in their

[1] Although the methods developed in this thesis can be used to build a model from multiple templates, only single template modelling is described. This serves to simplify the presentation without greatly restricting the generality – for example, some coordinate generation programs allow single template alignments to be combined into a multiple template model.

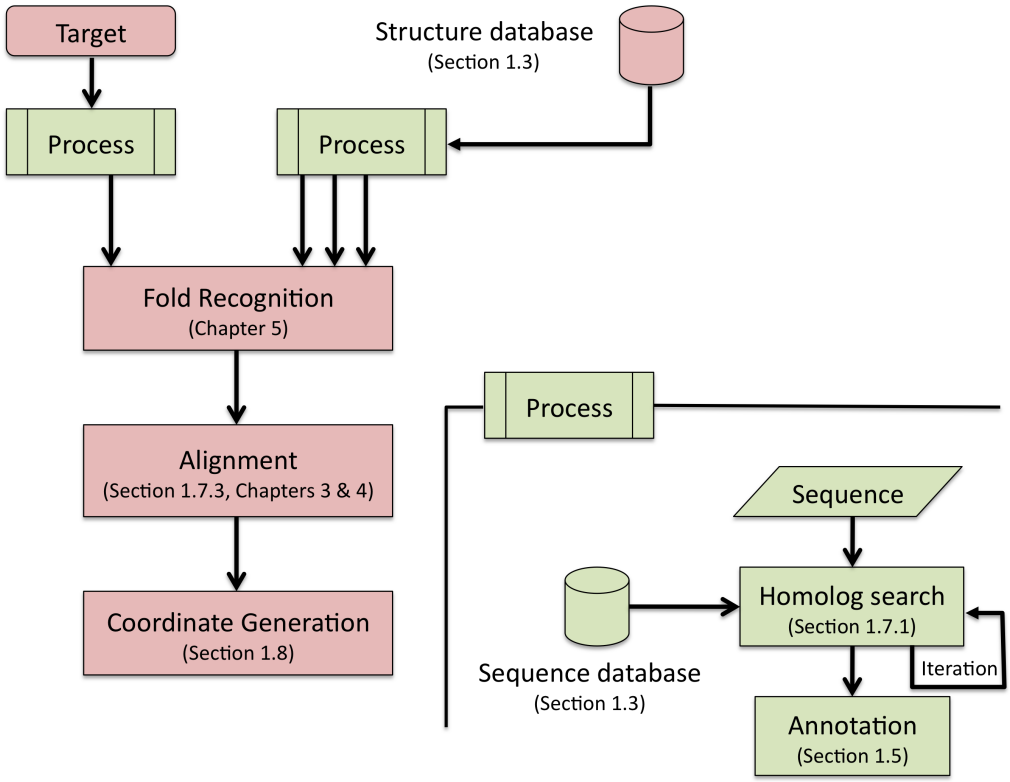


Figure 1.1: Flowchart showing the steps in comparative modelling of a target protein sequence. A template is selected from a structure database (fold recognition), then structurally equivalent residues are identified (alignment), and a model is created (coordinate generation). The inset 'Process' chart shows how a protein sequence may be processed to add information useful in structure prediction.

sequences, and demonstrates that these preferences can be used to improve pairwise alignment. Chapter 4 shows how this approach can be applied to multiple sequence alignment, leading to more accurate alignments than are possible with methods designed for soluble proteins. Chapter 5 uses signals of correlated substitution to perform fold recognition for both soluble and membrane proteins. The shortcomings of this approach are investigated in terms of the conservation of signals of correlated substitution over evolutionary timescales. Future directions are suggested in the final chapter.

1.2 Protein sequence and structure

Proteins are polymers made from amino acid monomers. There are 20 common naturally occurring amino acids, each of which is associated with a letter of the latin alphabet: for example the amino acid glycine is associated with the letter 'G'. A protein is specified by its 'sequence': an ordered list of its constituent amino acids (sometimes called residues), usually written in single letter form.

The largest protein sequences consist of tens of thousands of amino acids, but a typical protein is approximately 200 amino acids long. A biologically active protein may consist of several sequences in complex. Here each independent sequence is described as a protein or protein chain.

Protein sequences are written in the order of their synthesis (from the N-terminus to the C-terminus). For example, the sequence

```
MAST...
```

describes a protein produced gradually from the ribosome with the first synthesised amino acid being methionine, the second synthesised amino acid being alanine, and so on.

The most common naturally occurring amino acids are shown and discussed in Figure 1.2. Amino acid side chains vary in shape and hydrophobicity, and it is largely through such side chain differences that a protein sequence determines protein structure. For example, in soluble proteins it is usually energetically favourable for amino acids with hydrophilic side chains to be exposed at the surface of a protein, and for hydrophobic side chains to face the protein core. Side chains also affect the probability with which common structural motifs ('secondary structure', Section 1.2.2) are assumed.

The function of a protein is determined by its three-dimensional structure. Although only a handful of residues may be involved in a specific function such as a catalytic site, the full structure will determine function in a looser sense – for example through its effect on binding partners, diffusion rates, and stability.

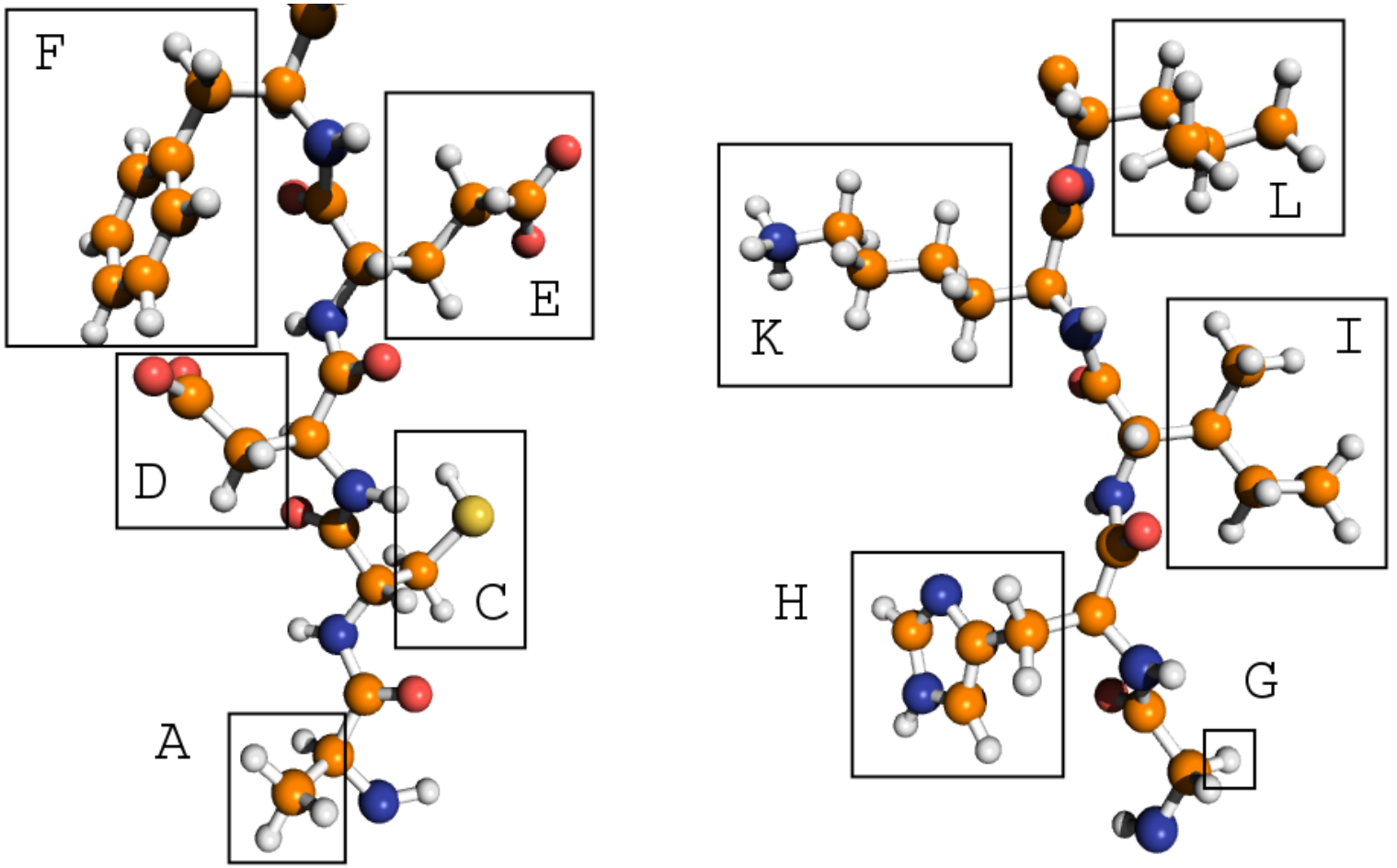
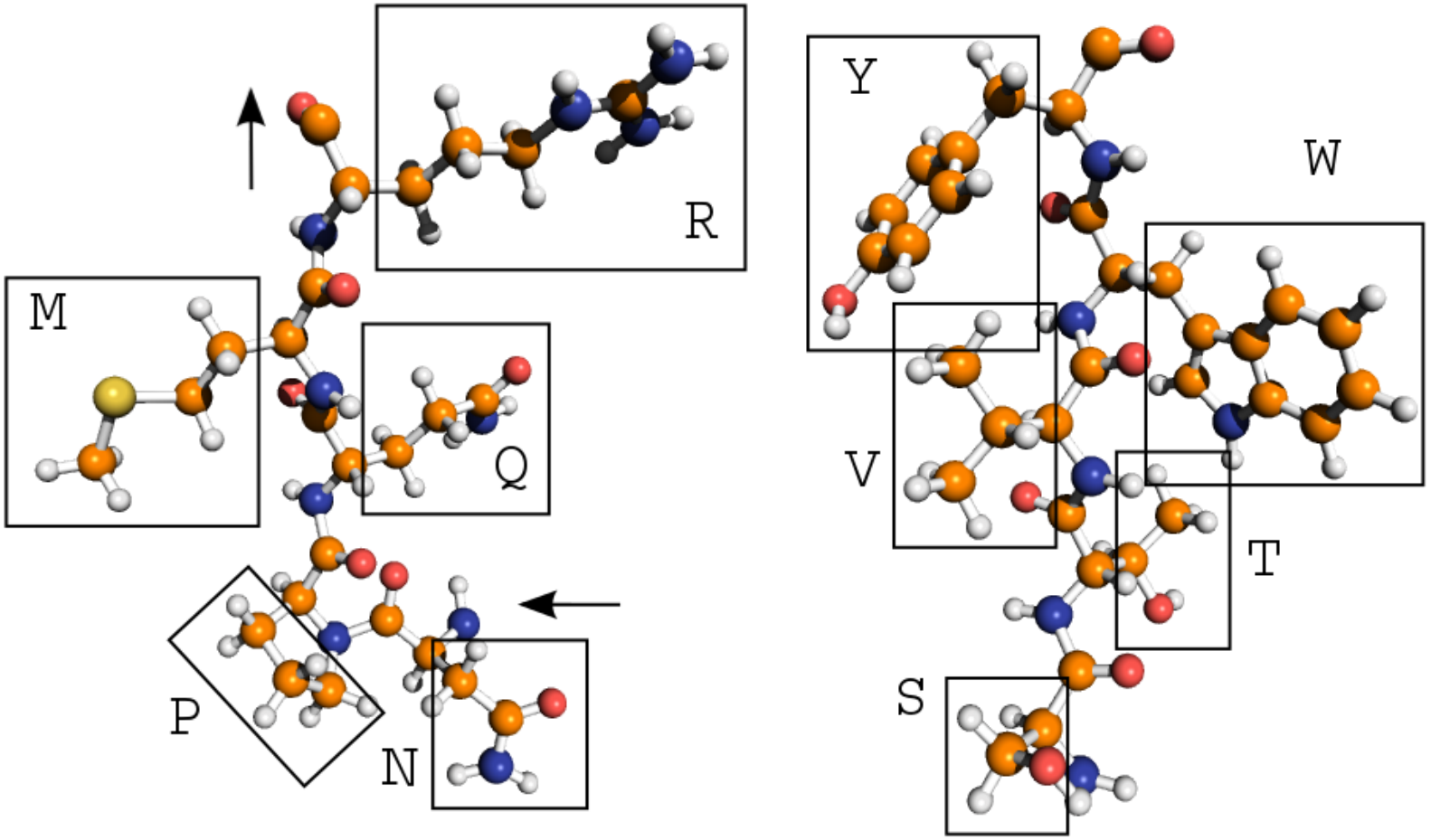


Figure 1.2: Computer-generated protein chains consisting of common amino acids. Carbon atoms are shown in orange, nitrogen in blue, oxygen in red, sulphur in yellow, and hydrogen in white. The ball and stick representation does not realistically depict the sizes or positions of actual atoms, and the exact number of hydrogen atoms depends on pH. Black rectangles enclose the side chains of each amino acid and are labelled with the one letter code for that acid. The carbon atom from which the side chain branches is the C_α atom. The first carbon atom in the side chain is the C_β atom (glycine 'G' does not have a C_β atom). The unboxed parts form the protein backbone, which repeats in the order $C_\alpha CN$ from the bottom to the top of the image *(caption continues on the next page)*



(Caption continued from previous page) Two cysteine residues 'C' may form a disulfide bond through the exposed sulfur atom at the end of the side chain. Proline 'P' has side chain atoms that link with the main chain, meaning that it often introduces a change in the direction of the main chain (as shown by the black arrows). Isoleucine 'I', threonine 'T', and valine 'V' branch out from their C_β atoms, and so do not easily form helices. Unless otherwise stated, all molecules in this thesis are visualised with *The PyMOL Molecular Graphics System, Schrödinger LLC.*

1.2.1 Domains

A single protein sequence may have a structure containing several regions that appear to fold independently of one another. Here each of these regions is defined as a 'domain'. The same domain is often found in other structurally dissimilar proteins, suggesting that domains are the natural scale at which protein structure should be analysed. At present, all known protein structures can be broken down into < 2000 distinct types of domain (so called 'folds') [Murzin *et al.*, 1995]. Physical considerations suggest that many more folds could be realised [Kuhlman *et al.*, 2003; Taylor *et al.*, 2009].

Other definitions of domains exist which overlap with these structured domains to a greater or lesser degree [Alexandrov and Shindyalov, 2003]. For example, the Pfam sequence database defines domains as common subunits of sequence found in different proteins (Punta *et al.* [2012], see Section 5.1.3).

1.2.2 Secondary structure

The atomic arrangement of amino acids favours two patterns of hydrogen bonding which give rise to common structural motifs: α-helices and β-strands. These, along with rarer motifs such as 3_{10}-helices[1], are collectively termed 'secondary structure'. Those parts of a protein that are not in secondary structure are variously termed 'loop' or 'coil' regions. Secondary structure is usually strongly conserved in homologous protein structures, and the selective pressures that maintain the secondary structure lead to different amino acid substitution probabilities compared with loop regions.

Some examples of secondary structure are shown in Figures 1.3 and 1.4. An α-helix is formed by a local pattern of hydrogen bonding with bonds existing between the carbonyl group of residue i and the amino group of residue $i + 4$ (which I will denote as $(i, i + 4)$). Two or more β-strands assemble into a β-sheet, with hydrogen bonding between the backbones of

[1] A summary of 3_{10}-helices in real protein structures is given by Vieira-Pires and Morais-Cabral [2010]. This also furnished the example in Figure 1.3

the strands. As the constituent strands of a sheet may be far apart in sequence, this form of hydrogen-bonding is non-local.

1.3 Sequence and structure databases

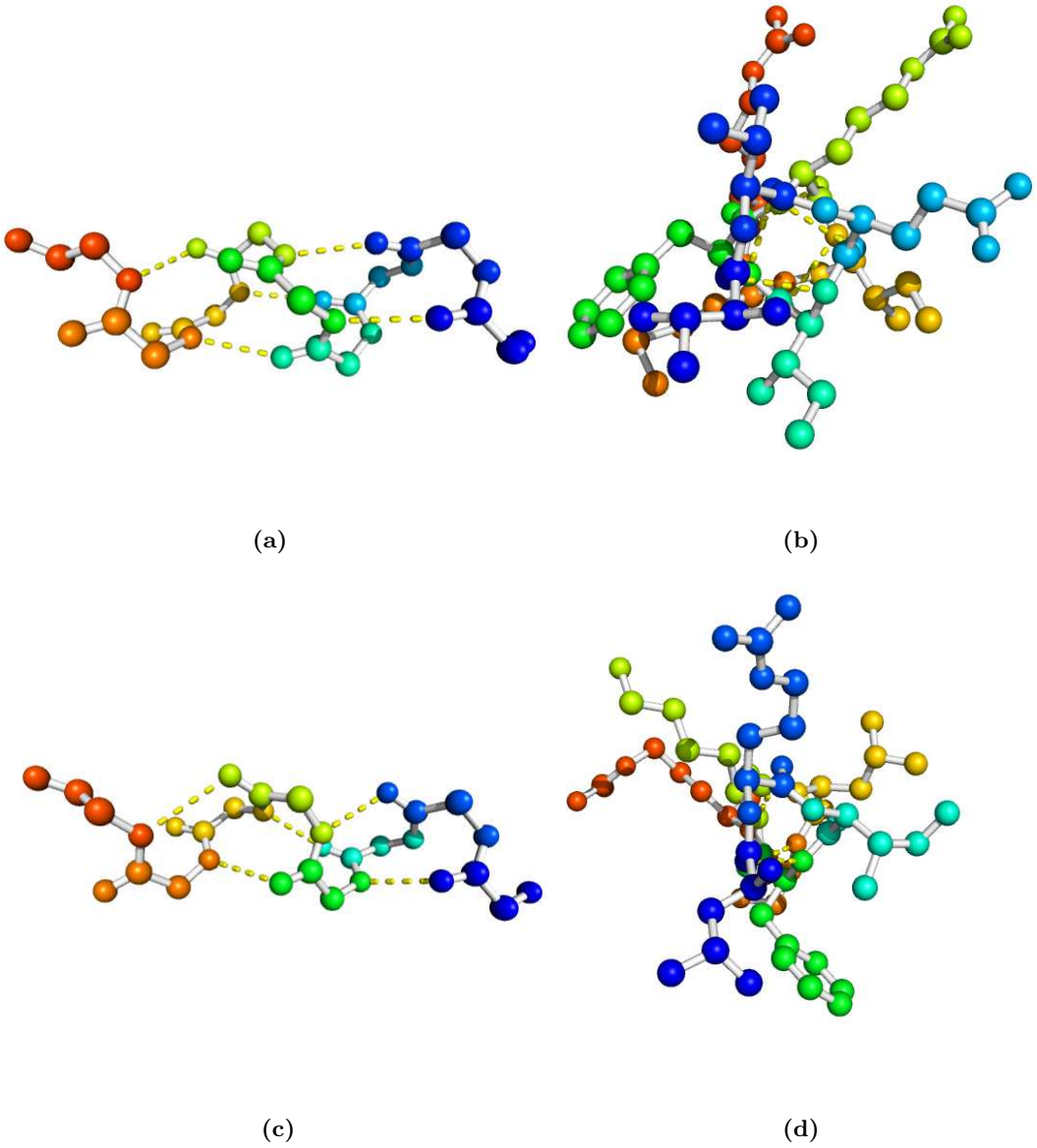
The protein data bank (PDB) is an open access repository of protein structural data [Berman *et al.*, 2000]. Most deposited structures are obtained by X-ray crystallography, but NMR and electron-microscopy data are also included. The deposited data is biased towards certain structure types such as lysozymes and kinases. In August 2013 there were $\sim$ 90000 entries in the PDB which contained $\sim$ 35000 non-redundant chains (here defined as having $< 90\%$ sequence identity).

Several databases exist that list the membrane proteins in the PDB. Two of these, OPM and PDB_TM, provide estimates of how a membrane protein is positioned in the membrane, and are discussed in Section 1.5.4. The Membrane Proteins of Known Structure database presents a manually-curated set of known structures in a hierarchical classification scheme [White, 2013].

In this work the NCBI nr and UniRef [Suzek *et al.*, 2007] non-redundant sequence databases are used. For NCBI nr, 'non-redundant' means that identical sequences are excluded, even if they come from different species. UniRef provides sets of sequences that are non-redundant at different levels of sequence identity: the August 2013 release contained 26 million sequences with $< 100\%$ identity (the UniRef100 database), 16 million sequences with $< 90\%$ identity (UniRef90), and 8 million sequences with $< 50\%$ identity (UniRef50).

1.4 Homology in proteins

Protein sequences undergo mutation, and over time descendants of an ancestral protein sequence will diverge. However, mutation is subject to selective pressures that act to preserve function. As function is more directly related to protein structure than sequence, this leads to less marked divergences in structure than in sequence. Indeed, proteins with random levels of sequence



(a) **(b)**

(c) **(d)**

Figure 1.3: **a** Side view of the backbone of an α-helix with the $(i, i+4)$ hydrogen bonding pattern indicated by dashed yellow line hydrogen bonds. **b** The same helix viewed from above and with side-chains added. The side-chains are splayed out at $\sim 100^\circ$ intervals. **c** Side view of the backbone of a 3_{10} helix with $(i, i+3)$ hydrogen bonding. **d** In the 3_{10} helix the protein backbone forms a triangle with side chains being thrown off at approximately $\sim 120^\circ$ intervals. These examples are actually different sections of one continuous helix taken from the voltage sensor of a human voltage-gated potassium channel (PDB code 2R9R, Long *et al.* [2007])

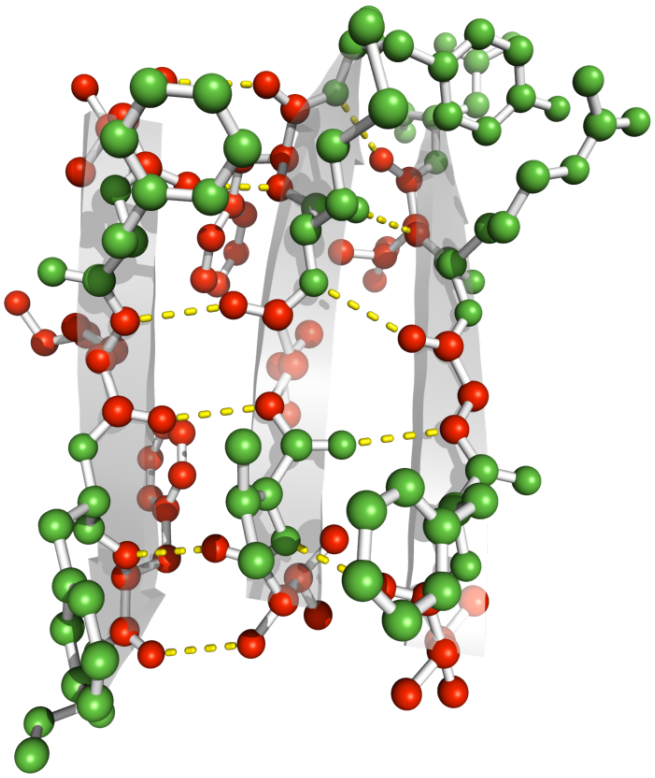


Figure 1.4: Three adjacent β-strands illustrating the two types of β-sheet. The middle strand forms an anti-parallel β-sheet with the strand on the left, and a parallel β-sheet with the strand on the right. Strands are often depicted as cartoon arrows (grey). Hydrogen bonds are shown by dashed yellow lines – they form a neat ladder between anti-parallel strands, but are crooked between parallel strands. Side chains are projected alternately above and below the plane of the paper. Residues in green have side chains above the plane and residues in red have side chains below the plane. This example forms part of a thioredoxin structure (PDB code 3HZ4).

identity can assume similar structures.

In this thesis, sequence similarity is often reported in terms of 'sequence identity', but other measures are mentioned in Section 1.7.2. Above the twilight zone, it can generally be assumed that pairs of proteins with higher sequence identity will have a more recent common ancestor. Proteins with a presumed common ancestor are said to be 'homologous'. For sequences with

low sequence identity, homology may also be inferred from structure. Measures of structural similarity are discussed in Section 1.6.

1.5 Annotation of structural features

Protein sequences are easier to work with than protein structures. A sequence takes up less space on a computer hard drive, is a one-dimensional rather than a three-dimensional object, and can be easily manipulated with string operations. These advantages have led to attempts to abstract structural features into a sequence form, with each residue in a sequence annotated with a letter denoting a structural feature. For many features there also exist methods of predicting an annotation in the absence of structure data.

In Chapters 3 and 4 of this thesis, sequences are annotated with their secondary structure, accessible surface area, and in the case of membrane proteins, their positioning within a membrane. The following sections describe how these annotations are made. Secondary structure annotations are given special emphasis because of their ubiquity.

1.5.1 Structure-based assignment of accessible surface area

The JOY program provides a single interface for several forms of structure annotation [Mizuguchi *et al.*, 1998b]. It calculates the accessible surface area of a residue by rolling a sphere of 1.4Å radius over the van der Waal's surface of a protein structure [Lee and Richards, 1971]. This sphere approximates the size of a water molecule. Accessible residues have $> 7\%$ of their side chain surface area exposed; other residues are inaccessible [Hubbard and Blundell, 1987].

1.5.2 Structure-based assignment of secondary structure

The DSSP [Kabsch and Sander, 1983] and STRIDE [Frishman and Argos, 1995] programs are among the most widely-used methods of automatically annotating each residue in a structure with its secondary structure type. In this thesis DSSP is used, which recognises 8 types of

secondary structure based on patterns of backbone hydrogen bonds. The DSSP criteria for assigning an annotation to residue i are described in terms of 'bridges' and 'turns':

Turn Residue i has a hydrogen bond of the form $(i, i+n)$ where $n \in \{3, 4, 5\}$

Bridge Defined between patches of three residues $i-1, i, i+1$, $j-1, j, j+1$ when one of the following pairs of bonds exists:

- $(i-1, j)$ and $(j, i+1)$
- $(j-1, i)$ and $(i, j+1)$
- (i, j) and (j, i)
- $(i-1, j+1)$ and $(j-1, i+1)$

A single residue can meet the bonding criteria for several annotations, so annotations are assigned in the order H,B,E,G,I,T,S,C where:

H Conventional α-helix minimally consisting of two $n = 4$ turns, one at $i-1$ and one at i

B Any lone bridge residue

E Any string of bridge residues. The definition is relaxed to allow for β-bulges, where extra non-bridge residues are found on one strand of a β-sheet

G 3_{10}-helix minimally consisting of two $n = 3$ turns, one at $i-1$ and one at i

I π-helix minimally consisting of two $n = 5$ turns, one at $i-1$ and one at i

T Any turn not assigned as part of a helix

S A bend: vectors are drawn along the main chain linking the C_α atoms of residues $i-2$ and i, and of residues i and $i+2$. If the angle between these vectors exceeds 70° the chain is bent around position i

C Any other residue

Note that the above definitions are arbitrary: their utility lies in that they are simple to calculate and agree with expert human judgement. Despite working from known structures, the manner in which annotations are assigned means that C_α-only protein structures cannot be annotated, and neither can isolated β-strands. The programs used in this thesis often further reduce the DSSP states:

JOY (See Section 1.5.1) The conformation of a protein backbone can be specified by two torsion angles at each residue, ϕ and ψ, which are rotations about the C_α atoms. JOY defines residues with $0 \leq \phi < 180°$ as P. Such 'positive-phi angle' residues are rare. JOY further merges annotations $H, G, I \rightarrow H$, and annotations $B, T, S, C \rightarrow C$

PSIPRED (Section 1.5.3) Merges $H, G \rightarrow H$; $B, E \rightarrow E$; $I, T, S, C \rightarrow C$.

HHsearch (Section 1.7.3) Merges the rare I annotation into the C annotation

1.5.3 Sequence-based prediction of secondary structure

Secondary structure is usually strongly conserved across closely-related ('homologous') proteins, and signals of secondary structure are found in the compositions of columns in a multiple sequence alignment (see Figure 1.5 for an example multiple sequence alignment). It is possible to predict a protein's secondary structure from a set of related sequences using programs that are often based on neural networks (e.g. Faraggi *et al.* [2012]; Jones [1999a]; Mirabello and Pollastri [2013]; Rost and Sander [1993]).

In this thesis the PSIPRED method is used. PSIPRED uses position specific scoring matrices (PSSMs) from PSI-BLAST (Section 1.7.1) as inputs to a two-stage neural network for secondary structure prediction [Bryson *et al.*, 2005; Jones, 1999a]. A PSSM is a matrix of scores encoding the propensity for a given position in a multiple sequence alignment to mutate to each possible amino acid. In the first neural network there are 315 inputs. These are the 20×15 scores in the PSSM over a window of 15 residues centred about the target column, and a further 15 inputs indicating whether the window extends over the ends of the protein. This first network produces

a set of 3 outputs, one for each of the three predicted secondary structure types (H, E, C see Section 1.5.2). The second neural network aggregates the 3×15 outputs for the 15 residue window to predict a final secondary structure type.

A recent study found PSIPRED to be among the most accurate of 12 predictors of secondary structure, with an average of 80% of predictions being correctly assigned to the states H, E, C (approximately 75% C, 80% E, 85% H) [Zhang *et al.*, 2011]. The more complex SOV measure assesses prediction not on a per-residue basis, but in terms of the amount of overlap between real and predicted secondary structure elements [Zemla *et al.*, 1999]. The same study found PSIPRED to have a three-state SOV score of $\sim 78\%$.

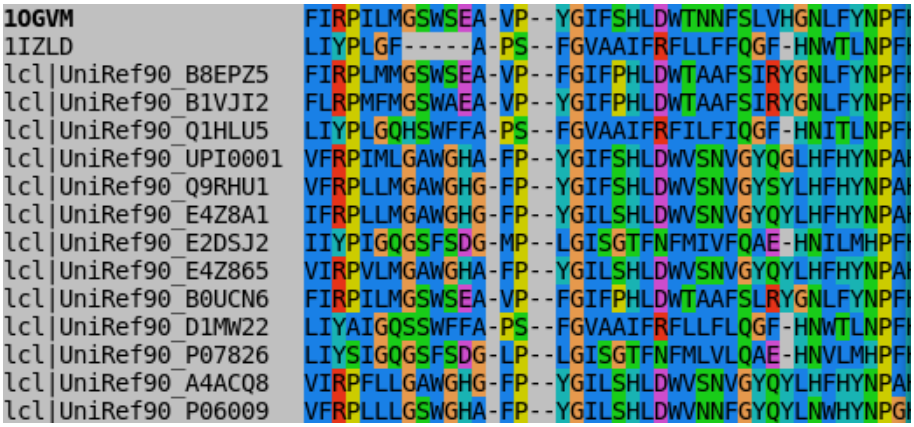


Figure 1.5: A snippet of a multiple sequence alignment (MSA). Each row contains one protein sequence. The degree of variation within a column, and the types of amino acids found in a column can be indicative of the column's role in a protein. The first column within this snippet varies, whereas the penultimate column is completely conserved. This alignment was made using MP-T (Chapter 4), and forms the basis for the comparative model built in Figure 4.8.

1.5.4 Structure-based assignment of membrane positioning

In this thesis, the iMembrane program [Kelm *et al.*, 2009], which depends on the CGDB database [Scott *et al.*, 2008], is used to assign annotations of membrane positioning to protein structures. CGDB is a database of coarse-grained molecular dynamics simulations of membranes assembling around a protein structure. iMembrane annotates each residue in a protein according to the

fraction of simulation time for which it is in contact with the simulated membrane, and its position within this membrane. To annotate a protein that is not in the database, iMembrane transfers the annotation from a structurally similar protein, if one is available.

At least two other databases provide estimates of membrane positioning. The PDB_TM database maximises an objective function that assumes membrane contacting residues to be predominantly hydrophobic, and membrane-spanning regions to lie approximately parallel to the membrane normal [Tusnády *et al.*, 2004]. The OPM database finds the most energetically favourable location of a rigid protein in a membrane using a sophisticated solvation potential [Lomize *et al.*, 2011].

1.6 Structure alignment

If the structures of two proteins are known they can be aligned (superimposed) to facilitate comparison. When aligning the structures of two proteins, a balance must be struck between alignment length (aligning as many residues as possible), and alignment fidelity (minimising the measure of difference between the two structures). It is unclear how to balance these factors, and so finding the ideal superposition of two protein structures is an ill-defined problem. Alignment fidelity itself is measured in several different ways, of which three are RMSD, TM-score, and GDT_TS.

RMSD is the **R**oot **M**ean **S**quare **D**eviation between equivalent atoms in two structures. The TM-score [Zhang and Skolnick, 2004] is an attempt to overcome the length-dependence of RMSD. It is a value in the range (0,1] and is defined by the equations:

$$\text{TM-score} = \frac{1}{L_N} \max \left[\sum_i \frac{1}{1 + \left(\frac{d_i}{d_0}\right)^2} \right] \qquad d_0 = 1.24 \sqrt[3]{L_N - 15} - 1.8 \tag{1.1}$$

where L_N is the length of one structure, the sum is taken over all aligned pairs of residues, and

d_i is the distance between the i^{th} pair. For a constant value of d_0 the score tends to increase as L_N decreases. The equation governing d_0 is an attempt to remedy this[1] – the size of a globular protein scales as $\sqrt[3]{L_N}$. Some proteins such as membrane proteins are not globular, so it is likely that this correction does not work as well for them as for soluble proteins. GDT_TS (**G**lobal **D**istance **T**est **T**otal **S**core) was developed for model quality assessment [Zemla *et al.*, 2001]. It is the average of the fraction of model residues aligned within 1.0, 2.0, 4.0 and 8.0Å of their position in the native structure.

Structure alignment programs work by identifying an equivalent set of residues, and then minimising a measure of alignment fidelity between these residues. The initial identification of equivalent residues often involves chaining together short structurally-similar fragments. There are a large number of programs designed for structure alignment, and only a few of the most popular are mentioned here.

In the DALI program [Holm and Sander, 1993] fragments are six residues long, need not be adjacent in sequence, and are assessed for structural similarity by comparing the intra-atomic distances of one fragment with the intra-atomic distances of another (i.e. there is no need to superimpose one fragment on the other). FATCAT [Ye and Godzik, 2003] uses eight residue long fragments that must be adjacent in sequence, and which are assessed for structural similarity by their RMSD after superposition. In contrast to the other algorithms mentioned here, FATCAT also allows hinges to be introduced into an alignment to allow for conformational changes between related protein structures. SAP [Taylor, 1999] and TM-align [Zhang and Skolnick, 2005] do not use fragment chaining – TM-align instead uses a fast heuristic to establish a set of equivalences. The slower but more accurate fr-TM-align algorithm [Pandit and Skolnick, 2008] adds fragment chaining back to the start of the TM-align alignment process.

TM-align is the most commonly used structure-alignment program in this thesis. It is used to provide 'gold standard' target-template alignments in Chapters 3 and 4, and to set an upper limit to structure based fold-recognition in Chapter 5. It is now described in more detail.

[1] The numbers in this equation have no special theoretical basis. They are chosen to provide a length independent TM-score for the superposition of unrelated globular proteins.

Example program: TM-align

TM-align is an iterative algorithm which treats proteins as rigid bodies and superimposes them so as to optimise the TM-score. To begin the procedure a first alignment is needed. This is obtained through combinations of alignment of secondary structure annotations, and gapless sequence alignment so as to maximise the TM-score. Each iteration then runs as follows:

1. Rotate the coordinates of the residues of the first structure onto their equivalents in a second structure using a variant of the Kabsch rotation matrix [Kabsch, 1976, 1978] that optimises the TM-score.

2. Define a scoring matrix $S_{a,b}$ between residue a in the first structure and residue b in the second structure. Each element of the matrix has the form of the summand in Equation 1.1.

3. Obtain a new sequence alignment using this matrix as a scoring function.

The main advantages of using TM-align are that it is fast and it reports the TM-score. In many applications in this work structure alignment is performed to obtain a measure of structure similarity. The implications of a given value of the TM-score for structure similarity are understood [Xu and Zhang, 2010].

1.7 Sequence alignment

Any form of sequence alignment requires 1) a scoring scheme for matching residues, 2) a penalty scheme for introducing gaps (indels) in the alignment, and 3) an algorithm to find the best alignment under these schemes.

A 'substitution table' lists the scores for matching any pair of amino acids, with higher scores for amino acids that are more likely to mutate into each other over evolutionary timescales. Substitution tables are discussed in detail in Section 3.1.2. If the sequences to be aligned have known homologs, the amino acids present at each position in the homologs can be taken into

account when calculating scores from a substitution table, leading to a 'Position Specific Scoring Matrix' (PSSM) or 'sequence profile'. For example, a sequence profile could be created for the alignment in Figure 1.5, and used when aligning a new sequence with the existing alignment.

Almost all aligners penalize gaps with a simple affine gap penalty: opening a gap incurs a large one-off penalty, and a smaller penalty is added for each subsequent gapped position. The Needleman-Wunsch algorithm or one of its derivatives can be used to retrieve the alignment with maximum score [Gotoh, 1982; Needleman and Wunsch, 1970; Smith and Waterman, 1981].

Alternative alignment algorithms become possible when the scoring system is given a probabilistic interpretation.[1] This is commonly done by defining a Hidden Markov Model (HMM), which represents an alignment by a series of states (typically 'match', 'insert' and 'delete' states). Each state has a fixed probability to transition to another state, and a probability to emit a pair of amino acids (in the match state) or an amino acid and a gap (in the insert or delete states). When restricted to this probabilistic setting, the Needleman-Wunsch algorithm is equivalent to the HMM-specific Viterbi algorithm.

An alignment between sequences $x = x_1, x_2 \dots x_n$ and $y = y_1, y_2 \dots y_m$ can be specified by a series of states $\pi = \pi_1, \pi_2, \pi_3, \dots$. For example, if π_1 was a match state then x_1 would be aligned with y_1. The probability of aligning any sequence x with any sequence y according to states π can be written as $P(x, y, \pi)$. The Needleman-Wunsch/Viterbi algorithm finds the set of states π^* that maximises this probability

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, P(x, y, \pi). \tag{1.2}$$

Unfortunately, in many cases the alignment π^* and its probability $P(x, y, \pi^*)$ are not the quantities that should be optimised. This is illustrated by separately considering the tasks of sequence search, multiple alignment, and target-template alignment.

In sequence search, the aim is often to determine the probability that x and y are related

[1] The book of Durbin *et al.* [1998] is the standard reference on this topic, and provides much of the material in this section.

(generated by the same HMM). The maximum score from the Viterbi algorithm, $P(x, y, \pi^*)$, approximates this so long as no other alignments are nearly as plausible as π^*

$$P(x, y) = \sum_{\pi} P(x, y, \pi) \approx P(x, y, \pi^*). \tag{1.3}$$

The HMM formalism allows the exact value of $P(x, y)$ to be computed using the 'forward algorithm', which effectively aggregates signals of relatedness over all possible alignments.

The related 'forward-backward' algorithm is often used in consistency based multiple sequence aligners (Section 1.7.2) to calculate the probability that two positions x_i and y_j are aligned to one another, $P(x_i \sim y_j)$. Finally, in target-template alignment the goal may not be to find the most likely alignment, but to find the alignment that contains the highest expected number of correct positions. This can be achieved by finding

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \sum_{(i,j)\in\pi} P(x_i \sim y_j). \tag{1.4}$$

As the above discussion suggests, it is useful when describing sequence alignment programs to distinguish between the tasks of sequence searching, multiple alignment, and target-template alignment, even though a single program may be suited to more than one of these tasks. In each of the following sections, one of these tasks is considered and a representative program is described in more depth.

1.7.1 Sequence search

Search programs scour a database for sequences homologous to a query sequence. They require an alignment scheme that allows the quality of matches to be statistically assessed, and prioritise speed over accuracy of alignment. In fact, for many applications no alignment is necessary, and only the probability of two proteins being related is reported – for example, when determining if a protein in one species has a homolog in another.

The oldest approaches such as FASTA [Pearson, 1988] and BLAST [Altschul, 1990] rely on

pairwise sequence comparisons, but more recent methods such as PSI-BLAST [Altschul *et al.*, 1997] and HMMER [Eddy, 2011] compare each database sequence with a sequence profile of previously identified homologs so as to detect more distant relationships. It is also possible to search against databases of profiles using tools such as HHblits [Remmert *et al.*, 2012]. Searching with profiles is usually achieved by a two step scheme where the results of an initial sequence search are turned into a profile that is used in a second search. This process can be iterated with results from each search used to refine the profile used in the next iteration. However, iterative approaches can lead to large numbers of false positives if a profile becomes 'polluted'. A common cause of this is termed 'homologous over-extension', and occurs when a non-homologous region of a sequence retrieved in an early iteration is matched to other sequences [Gonzalez and Pearson, 2010].

Example program: PSI-BLAST

The BLAST family of methods is ubiquitous. Several descendants of the original BLAST algorithm [Altschul, 1990] exist for protein sequences, of which PSI-BLAST is used here [Altschul *et al.*, 1997]. PSI-BLAST is iterative: the first iteration uses the common BLAST algorithm, which is then modified to allow subsequent iterations. The steps of the PSI-BLAST algorithm are as follows:

1. A search is made for fixed-length 'words' ($\sim$ 3 residues long) in the query sequence that align with a high score to words in a second sequence from a database. Each such pairing of words is called a 'hit'.

2. The resulting set of hits is reduced to a set of paired hits, where pairs lie within a pre-determined distance of each other ($\sim$ 40 residues) and *could* be linked by an un-gapped alignment. No check is made to see if the resulting un-gapped alignment would suggest a plausible evolutionary relationship.

3. The second hit of each pair is extended in both directions so that the aligned region of

the hit increases in size. Extension stops when the current score falls a certain threshold below the maximum score observed during extension.

4. If the maximum score from the above step is sufficiently high, a gapped alignment is made, starting from the centre of the extended hit. Other than the introduction of affine gap penalties this step is identical in form to the first extension.

5. An e-value is calculated for the extended hit (or the hit after the gapped alignment if such an alignment has been made). Hits with e-values beneath a user-determined threshold are reported. Hits with e-values beneath a second user-determined threshold are used to construct a query profile for the next iteration.

6. If the original query sequence is of length L then the query profile is a $20 \times L$ matrix. This is used in place of the 20×20 matrix used to score hits in step 1. The construction of the profile is itself a multiple stage process.

   (a) Each sequence used in constructing the profile is aligned to the query sequence (or the query profile in later iterations) to produce a multiple sequence alignment (MSA). The score for each column is calculated using a sub-alignment extracted from this MSA. The sub-alignment for column i has rows for every sequence with an amino acid in column i, and columns for the positions common to all of these rows.

   (b) Due to phylogenetic biases, not all sequences should be treated as equally representative of the evolutionary pressures at a position. The sequences within a sub-alignment are re-weighted to account for this. Thus, each position in the profile may contain a different total sequence weight.

   (c) Positions have 'pseudocounts' (synthetic observations of amino acids) added to them in the hope of making the position represent more closely the underlying evolutionary pressures at that position. Pseudocounts are added in varying numbers from column to column [Altschul *et al.*, 2009].

7. This process is repeated from step 1 until either no new hits are found or a pre-determined number of iterations have elapsed.

The design of PSI-BLAST has implications for both search specificity and alignment accuracy. A single false positive sequence included in an early iteration may pollute the profile leading to many subsequent false positive hits. There is no clear consensus on how to mitigate this problem: the user could choose a stricter e-value threshold or reduce the number of iterations. PSI-BLAST profiles are built first by aligning every sequence onto the query, and later by aligning every new sequence onto the query profile. Compounding pairwise alignments into a single profile is not expected to lead to an accurately aligned profile, so PSI-BLAST results are often realigned before use.

As their role in iterative searching makes clear, PSI-BLAST's utility largely depends on the accuracy of its e-values. The expected number of hits between sequence a of length m and sequence b of length n is:

$$E = \mathbb{E}\{(m - L_a(S))(n - L_b(S))\}\ ke^{-\lambda S} \tag{1.5}$$

Where λ and k respectively rescale and centre the substitution matrix used to calculate score S [Karlin and Altschul, 1990]. The first term is a finite-size correction estimate of the number of possible hits that could be extended to give a score at least S. Under the simplifying assumption that all hits are extended only in the C-terminal direction, some hits will begin too near the end of the sequence to achieve a high score. The total number of possible hits is $m \times n$, but $L_a(S)$ is the minimum length of a hit between a random sequence and sequence a needed to achieve score at least S (and equivalently for $L_b(S)$) [Park *et al.*, 2012].

1.7.2 Multiple sequence alignment

Multiple sequence alignment methods show the evolutionary relationships between groups of sequences (usually sequences that have been found by a program such as PSI-BLAST). The most

widespread approach is 'progressive multiple alignment', which first aligns pairs of sequences into profiles, and then aligns pairs of profiles into larger profiles. The ordering of these alignments is determined by a 'guide tree', with alignments starting at the leaves and proceeding to the root.

Fewer mistakes are made in aligning similar sequences than in aligning more distantly related ones, so the tree is usually constructed such that similar sequences are aligned first. This arrangement also means that the more difficult alignments near the root are made with profiles that contain many sequences, increasing the likely accuracy of the alignment method. It is tempting to think that the closer the tree is to the evolutionarily correct one, the better the final alignment will be – this was the motivation for some of the earliest progressive multiple alignment algorithms such as that of Hogeweg and Hesper [1984]. However, algorithmic details, such as the handling of gaps may mean that, for example, a sequence containing a long insertion should be aligned later than evolutionary history suggests.

To construct a tree, a measure of distance is required between each pair of sequences. Measures that have been used include estimates of expected alignment accuracy (ProbCons, Do *et al.* [2005]), functions of the total alignment score for each pairwise alignment (MSAProbs, Liu *et al.* [2010]), the percentage of identical residues between each pairwise alignment (T-Coffee, Notredame *et al.* [2000]), and transformations of this percentage identity (MUSCLE, Edgar [2004]). Most programs construct trees by UPGMA [Michener and Sokal, 1957] or neighbor-joining [Saitou and Nei, 1987; Studier and Keppler, 1988]. Potentially more accurate maximum-likelihood methods of tree estimation are generally eschewed due to their greater time requirements.

One weakness of the progressive multiple alignment process is that once a profile has been made it cannot be changed. This means that small errors during early alignments are fixed in place, and may cause larger errors as the guide tree is followed back to the root. There are two general schemes to avoid this: consistency schemes aim to use as much sequence information as possible even in the early alignments, whereas iterative schemes refine the output of progressive alignment.

Perhaps the best known consistency-based method is T-Coffee. It does not score the matching of two profiles by a substitution table, but by a weighted sum of the number of times that a pair of positions are aligned through an intermediary. The weights are determined by sequence identity. For example, in the alignment below, sequences A and B are shown aligned separately to sequence C (left), and aligned to each other (right). The match of 'I' with 'I' does not occur through sequence C so this scores only $3/7$ (the fractional identity between A and B). The match of 'G' with 'P' meanwhile scores $3/7 + 3/5$.

```
A: GIRAFFE          A: GIRAFFE
   \ || //             |||||||
C:  CRANE           B: PIRANHA
   / ||||
B: PIRANHA
```

Most modern multiple sequence aligners use consistency in the more explicit form of conditional probabilities that two amino acids are aligned with a third (e.g. Do *et al.* [2005]; Liu *et al.* [2010]).

In Chapter 4 the performance of several multiple sequence alignment methods is assessed on membrane protein data, and an overview of individual methods is provided in Table 4.2. One of these methods is the MAFFT L-INS-i program described in the next section. MAFFT L-INS-i has been chosen for more detailed description because it includes both consistency and iterative refinement, and was used in the construction of the substitution tables of Chapter 4.

Example program: MAFFT L-INS-i

Mafft is an actively developed suite of multiple sequence alignment programs [Katoh, 2002; Katoh and Toh, 2008; Katoh *et al.*, 2005]. The different versions are designed for a variety of tasks, from aligning small numbers of sequences accurately to handling tens of thousands of sequences. The version used throughout this work is MAFFT L-INS-i, which is thought to be

the most accurate on average.

As described above, multiple sequence alignment follows the ordering of a guide tree, with alignments starting at the leaves and proceeding to the root. MAFFT L-INS-i estimates a guide tree by supplying a modified version of the UPGMA algorithm with distances between sequences calculated from pairwise alignments.

Alignments between profiles are scored by a combination of an amino acid substitution score and a consistency score. The substitution score matches amino acids with similar physicochemical properties. The consistency score favours matching amino acids that fall within local regions of conserved physicochemical properties, or that are present (i.e. not deleted) in many input sequences.

The overall score between position i in profile S occupied by sequences labelled by s, and position j in profile T occupied by sequences labelled by t is a sum over all pairings of sequences within the profiles:

$$\sum_{s \in S, t \in T} w_1(s,t) \left(W(a(s,i), a(t,j)) + w_2 \frac{S(s,t,n)}{L(s,t,n)} \sum_{a \in n, b \in n} \frac{f(s,a) + f(t,b)}{2} \right) \tag{1.6}$$

Here $w_1(s,t)$ is a weighting factor that reduces the score if sequences s and t are closely related. The identity of the amino acid at position i of sequence s is given by $a(s,i)$, and the term $W(a(s,i), a(t,j))$ is the amino acid substitution score between positions i and j. The second scoring term is the consistency score, which is weighted by a constant factor w_2. The consistency score is defined over ungapped segments obtained from the local pairwise alignment of sequences s and t. If position i in sequence s and position j in sequence t both belong to ungapped segment n, then $S(s,t,n)$ is the substitution score computed for this segment, and $L(s,t,n)$ is the length of this segment. If the positions do not belong to the same segment $S(s,t,n)/L(s,t,n) = 0$. Note that this part of the consistency score is equal for matching any two residues within the same ungapped segment.

The final part of the consistency score is the average of $f(s,a)$ and $f(t,b)$. These are the

sums of the sequence weights (an estimate of the number of independent sequences) of all the sequences that have a residue equivalent to residue a in sequence s (or residue b in sequence t). This term ensures that segments that are present in many distantly related sequences receive the highest scores.

Once an alignment has been made according to this procedure, it is refined in an iterative process. Each iteration splits the alignment into two clades according to the guide tree, and re-aligns these two groups to each other. In the iterative refinement stage MAFFT optimises a combination of the WSP score described by Gotoh [1995] and the consistency score. WSP is an approximation to the first substitution score term of Equation 1.6 that can be calculated extremely quickly on a tree structure.

1.7.3 Alignment for comparative modelling

Comparative modelling has three distinct steps (Figure 1.1): fold recognition (the identification of a suitable template), alignment to this template, and coordinate generation. Fold recognition can be seen as a special case of the sequence search task described in Section 1.7.1, target-template alignment is the subject of this section, and model building is discussed in the next section.

In contrast to sequence search methods, target-template alignment or 'threading'[1] prioritises alignment accuracy over speed: an alignment can potentially inform months of experimental effort, so it is in many cases unimportant whether it takes seconds or hours to make. Although threading methods may use multiple sequences, they differ from multiple sequence alignment methods in that they only require two of those sequences to be accurately aligned (the target and the template). Finally, threading methods can use structural information from the template, such as the location of α-helices, to improve alignment.

The eponymous threading method, THREADER, 'threaded' a target sequence on to the 3D structure of a template [Jones *et al.*, 1992]. The sequence information in the template was

[1] Threading is often taken to denote a target-template alignment that is made with structural information. Our usage emphasises the aim, rather than the method.

discarded, and the alignment was assessed in terms of the compatibility of the target sequence with the coordinates of the template. This assessment was performed using a set of statistical potentials that expressed the probability for two atoms in a pair of amino acids separated by a certain distance in sequence to have the required spatial separation. Most current threading methods include sequence information for both the target and template in the form of sequence profiles, and use less information from the 3D structure of the template than THREADER.

HHsearch [Söding, 2005] (described in detail below) and PROMALS [Pei and Grishin, 2007] incorporate sequence profiles and predicted secondary structure in an HMM-based maximum expected accuracy alignment (Equation 1.4). Each is also capable of using additional sources of information: HHsearch can include DSSP annotation of the template structure; PROMALS can include multiple templates and structure alignments between such templates [Pei *et al.*, 2008]. They are among the most accurate available threading methods despite using only predicted structural information by default.

Compared with these methods, SPARKS-X [Yang *et al.*, 2011] and RAPTORX [Peng and Xu, 2010, 2011] use a large amount of additional information. SPARKS-X compares predictions of solvent accessibility and dihedral angles (ϕ and ψ, see Section 1.5.2) from the target with the actual values from the template. It then optimises an alignment score that combines these measures with comparisons of sequence profiles and secondary structure. RAPTORX is motivated by the observation that not all of these scores are independent – for example, ϕ and ψ angles correspond closely with secondary structure type – and so should be combined non-linearly. It scores an alignment by a linear sum of regression trees, each of which has different weights for structure and sequence features. The number of considered features is large and includes sequence profiles, predicted secondary structure, predicted solvent accessibility, sequence similarity based on a statistical potential, measures of the number of contacts each amino acid type favours, alignment length, a substitution table constructed from structure alignments and so on.

An older approach, FUGUE [Shi *et al.*, 2001], neatly circumvents the interdependence of

structural features by using these features only to alter the expected frequency of different amino acid substitutions. For example, FUGUE has one set of substitution tables for solvent accessible α-helical regions, and another set for solvent inaccessible loop regions. In Chapter 3 we develop new 'environment specific substitution tables' that incorporate membrane positioning as a structural feature, and test these on pairwise alignments in FUGUE. In Chapter 4 we develop a multiple sequence alignment threading method based on the concept of FUGUE.

HHsearch is used as a benchmark in Chapter 4 and is the principal method with which contact-based fold recognition is compared in Chapter 5. As its name suggests, it is an effective search approach that could also have been listed in Section 1.7.1. It is now described in more detail.

Example program: HHsearch

HHsearch aligns two HMMs to each other, each possessing a set of *M*atch, *I*nsert and *D*elete states. Two columns of a pair of HMMs can be aligned if they are in one of the following paired states: $\{MM, MI, IM, GD, DG\}$ where G represents a gap in one of the HMMs. The total score of an alignment is:

$$S = \log P_{transition} + \sum_{k} \log \left(\sum_{a=1}^{20} \frac{q_k(a) p_k(a)}{f(a)} \ (+\text{SecStruc}) \right) \tag{1.7}$$

Here the $P_{transition}$ term is the product of the transition probabilities between pairs of states. This term embodies the position specific analogues of gap penalties. The logarithm contains the score for aligning two HMM match states: $q_k(a)$ is the probability that the column in the first HMM corresponding to the k^{th} column in the alignment emits amino acid type a ($p_k(a)$ is the equivalent statement for the second HMM). The function $f(a)$ is the background probability of observing amino acid a. When column k of the alignment includes an insert state, then e.g. $q_k(a) = f(a)$ and the logarithm evaluates to zero. HHsearch optionally includes predicted secondary structure into the score of an alignment. This is done by adding an estimate of the

probability that two HMM columns have the same secondary structure given their predicted secondary structure:

$$\text{SecStruc} = 0.15 \sum_{\sigma} P(\sigma) \frac{P(\rho_k^q, c_k^q|\sigma)}{P(\rho_k^q, c_k^q)} \frac{P(\rho_k^p, c_k^p|\sigma)}{P(\rho_k^p, c_k^p)} \tag{1.8}$$

Here $\rho_k^q \in \{H, E, C\}$ is the PSIPRED secondary structure prediction for the column in the first HMM that corresponds to the k^{th} column in the alignment, and $c_k^q \in \{0, 1 \ldots 9\}$ is the confidence value associated with this prediction. The three possible values of ρ are internally associated with a refined labelling of seven types of secondary structure from DSSP denoted by σ. The probabilities are calculated from sequence alignments of a set of known structures.

For sequence search applications, once the best alignment has been identified it is re-scored, and the new score is used to calculate an e-value. The re-scoring makes use of the observation that high-scoring pairs of columns in an alignment of non-homologous sequences are randomly distributed, whereas high-scoring columns in an alignment of homologous sequences clump together in patches. The new score is implemented by adding an autocorrelation measure to each column. In sequence search, HMM-HMM alignment comes with inherent drawbacks. For example, it is not possible, as it is with PSI-BLAST, to identify individual sequences that are related to a query: either every sequence in an entire HMM is matched or none are. This means that search performance depends on the accuracy with which HMMs are constructed.

1.8 Coordinate generation

Once a template has been identified and aligned with the target sequence, the final step in comparative modelling is to use this alignment as a blueprint for model building or 'coordinate generation'. There are two common approaches to coordinate generation, 'rigid-body assembly' and 'satisfaction of spatial restraints'. In rigid-body assembly, the coordinates of aligned regions in the template are copied directly onto the target. This process leaves gaps in the model where the target has an insertion with respect to the template, which are modelled with a separate

'loop-modelling' method such as PLOP [Jacobson *et al.*, 2004] or FREAD [Choi and Deane, 2010]. The coordinates of some side chain atoms may be copied from the template, but the identities of many side chains will change, and need separate modelling with programs such as SCRWL4 [Krivov *et al.*, 2009] and RASP [Miao *et al.*, 2011]. A popular rigid-body assembler is SWISS-MODEL [Arnold *et al.*, 2006].

Satisfaction of spatial restraints is implemented by the widely used MODELLER tool [Sali, 1993]. MODELLER is a general framework that represents restraints on protein structure using probability distributions. The number of restraints is typically high, and the best model satisfies as many as possible. Example restraints include probability distributions for different bond lengths, bond angles, and inter-residue distances, all of which can be conditioned on observations of these parameters in the template [Eswar *et al.*, 2007].

In Chapters 3 and 4 of this thesis, models of membrane proteins are built with MEDELLER, a rigid-body assembler designed specifically for use on membrane proteins [Kelm *et al.*, 2010]. MEDELLER builds models outwards from the centre of the transmembrane region, where membrane protein structure is most conserved. Model building stops when MEDELLER assesses that the local region of an alignment does not support the assumption of structural similarity between the target and template. This alignment assessment is performed by averaging substitution scores from tables constructed as in Chapter 3 over a sliding window. The remaining gaps in the model can be filled by the FREAD database loop-modelling method [Choi and Deane, 2010; Kelm *et al.*, 2013] and MODELLER.

1.9 Membrane proteins as test cases

Model building for membrane proteins is important, as it is particularly difficult to obtain membrane protein structural information experimentally. In the past few years the number of membrane protein structures has risen to the point where comparative modelling can be applied to a large number of sequences, but often with $< 30\%$ sequence identity between the target and

template. Within this 'twilight zone' of sequence identity alignment becomes challenging, and misalignment is the principal source of modelling error. The next chapter describes membrane proteins in more detail, and the composition of the membrane in which they reside. This serves as preparation for Chapters 3 and 4, which are concerned with improving the alignment of membrane protein sequences.

CHAPTER 2

Membrane proteins and biological membranes

2.1 Why focus on membrane proteins?

Proteins embedded in membranes constitute ∼30% of the human proteome [Almén *et al.*, 2009], but only ∼ 2% of structures in the PDB.[1] This disparity is not due to importance: the location of membrane proteins on the surface of cells and their involvement in signalling pathways make

[1] On 29th August, 2013 there were ∼ 93500 structures in the PDB and 2300 structures in the OPM database [Lomize *et al.*, 2011], see Section 1.3.

them targets for approximately 1/2 of current and future drugs [Overington *et al.*, 2006]. The disparity arises because membrane proteins are inherently hard to crystallise – the hydrophilic conditions required for crystallisation are often incompatible with the hydrophobic conditions in which a membrane protein assumes its native structure.

At present there are ~ 400 unique membrane protein structures.[1] Even this figure includes the same protein from different species [White, 2013]. The small number of structures means that for most membrane proteins only a sequence is known, making modelling of membrane protein structures especially important.

2.2 The membrane protein zoo

Membrane proteins can broadly be split into two classes: β-barrels, which are found in the outer membranes of Gram-negative bacteria and mitochondria, and α-helical bundles, which are found everywhere else. Membrane spanning β-barrel structures have greater rigidity and crystallise more readily than α-helical structures. However, hydropathy plots (plots of the hydrophobicity of residues averaged over a sliding window) allow transmembrane helices to be detected from sequence alone as regions of high average hydrophobicity, whereas β-barrels tend to have an irregular pattern of hydrophobic and hydrophilic residues that thwarts such analyses. Bioinformatics methods typically concentrate on the α-helical proteins [Elofsson and von Heijne, 2007].

2.2.1 β-barrel structures

Sequence searches suggest that there may be up to 400 families of β-barrel outer membrane proteins (OMPs) [Remmert *et al.*, 2009], one example of which is shown in Figure 2.1e. Almost all known structures have the following features [Schulz, 2005]:

[1] This number is from the MPStruc database [White, 2013], accessed 29th August, 2013. Here 'unique' excludes proteins with the same sequence but different substrate, or different experimental mutants of the same naturally occurring protein.

- Both the N terminus and C terminus are on the inside of the membrane

- The transmembrane barrels are composed of an even number of anti-parallel β-strands. This is thought to be because they are made up of varying numbers of copies of a common $\beta\beta$-hairpin precursor [Remmert *et al.*, 2010].

- The strands are chiral: they run up and to the right / down and to the left, at angles of 30 – 60 degrees from the vertical.

OMPs are often classified by a tuple (n, S) where n is the number of β-strands, and S is the 'shear number'. If you start at the uppermost residue of the first β-strand, and follow the hydrogen bonds around the barrel back to the first strand, the shear number is, roughly speaking, the number of residues between your start and end points. The largest structure to date contains 24 strands [Remaut *et al.*, 2008], but structures with 8–18 and 22 strands have also been determined.

In the set of solved structures there are several examples of trimeric pores called porins (Figure 2.1e). Each porin monomer is a barrel composed of 16 or 18 β-strands. These strands are linked by short 'turns' on the periplasmic side of the membrane, and longer 'loops' on the extracellular side. Initially the 16 stranded porins were classed as 'unspecific' in terms of the molecules that they allowed to transit their pore, but this classification has been questioned [Zeth and Thein, 2010]. OmpG is a monomeric pore of 14 strands which is sometimes termed a porin [Yildiz *et al.*, 2006].

To date only one mitochondrial OMP (VDAC-1) has been crystallised. This appears related to the trimeric porins of Gram-negative bacteria, but it has an odd number of 19 strands (meaning that two strands – the terminal strands – are parallel to each other), and is thought to assemble into a hexameric complex.

2.2.2 α-helical structures

It has been estimated that 1700 structures are required to provide a representative for 90% of observed α-helical membrane protein sequences [Oberai *et al.*, 2006]. Such a large coverage is possible with so few structures because some structural types have many representatives in genomes. G-protein coupled receptors (GPCRs) appear to be most widespread, with perhaps 800 examples in the human genome. A GPCR is shown in Figure 2.1f. Despite their name, GPCRs do not exclusively couple to G-proteins, but are instead distinguished by having 7 transmembrane helices [Fredriksson *et al.*, 2003]. Phylogenetic analyses suggest at least 5 main subtypes, of which the largest is the 'rhodopsin' group [Rosenbaum *et al.*, 2009].

A second large group of structures are the ATP-binding cassette (ABC) transporters – of which there are ~ 50 in humans. These pump a range of molecules in to (prokaryotes) and out of (prokaryotes and eukaryotes) a cell in an ATP-powered process. Each ABC transporter typically consists of two variable transmembrane domains, and two ABC subunits. However, exporters are dimerised such that each monomer contains one ABC subunit and one transmembrane domain of 6 helices. The transmembrane domains of importers are classed into two types, one with 10 helices per subunit $(10, 10)$, and the other with a variety of helical divisions including (5,5), (6,6) and (6,8). It is thought likely that other arrangements exist [Rees *et al.*, 2009].

The variety of known α-helical transmembrane proteins exceeds that of β-barrel transmembrane proteins: White [2013] lists more than 50 categories of α-helical structures but fewer than 10 categories of β-barrel structures. Some of these categories are discussed in White [2009], and some are illustrated in Figure 2.1.

2.2.3 Unusual membrane structures

The preceding discussion roughly divided membrane proteins into β-barrels in bacterial outer membranes, and α-helical structures elsewhere. However, β-strand based transmembrane proteins have been found in places other than outer membranes. For example, the α-hemolysin pore is a 14 stranded barrel made of 7 double-stranded monomers produced by the Gram-positive

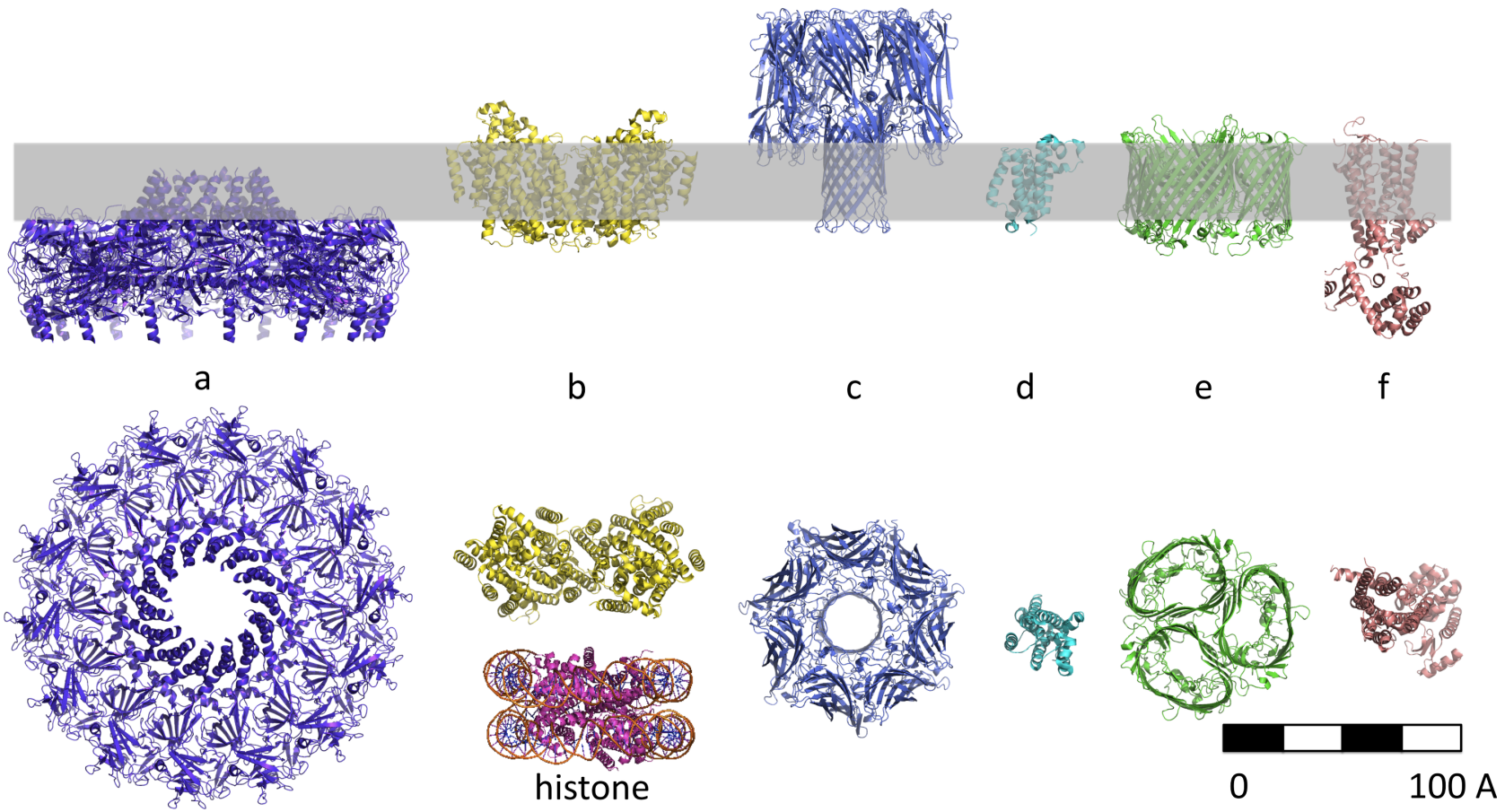


Figure 2.1: Side and top views of membrane proteins in a membrane (grey). A histone is included so as to show DNA to the same scale. **a** Outer membrane part of Type IV secretion complex of *E. coli* (PDB code 3JQO). This is one of two known examples of an α-helical OMP **b** Dimer of sodium/galactose symporter of *V. parahaemolyticus* (3DH4). With 14 helices in each monomer, this is among the largest helical bundles **c** α-hemolysin, a β-barrel protein that inserts into the plasma membranes of red blood cells (7AHL) **d** GlpG rhomboid protease – able to catalyse hydrolysis in a hydrophobic membrane (2IC8) **e** Sucrose-specific porin of *Salmonella typhimurium* (1A0T) **f** Human A2A adenosine receptor – an example of a GPCR (3EML).

bacterium *Staphylococcus aureus* (Figure 2.1c). The pore inserts into eukaryotic plasma membranes, leading to lysis of the target cell. Conversely, α-helical bundles have been found within the outer membrane. At present there are two known structures; one sits entirely in the outer membrane [Dong *et al.*, 2006], whilst the other is the outer membrane part of a larger complex that spans both membranes (Figure 2.1a, Chandran *et al.* [2009]).

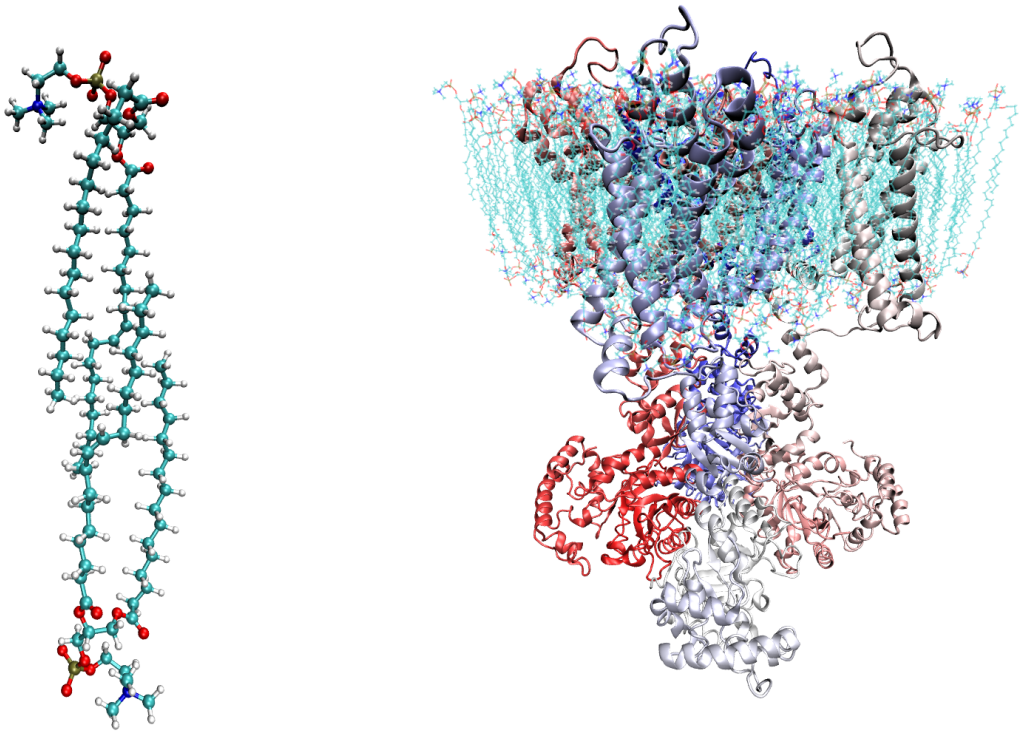
The mechanism of perforin, a protein produced by T-cells and apparently related to components of the Macromolecular Attack Complex poses problems for the classification of membrane proteins by their secondary structure. An α-helical domain is thought to refold into β-strands, which then insert into a vesicle (from the inside) to form a pore [Browne *et al.*, 1999; Law *et al.*, 2010].

2.3 The diversity of membranes

The structures of membrane proteins are constrained by the membrane in which they sit. Hence, to improve comparative modelling of these structures, some understanding of the membrane is also required. All cells have a plasma membrane, but eukaryotic cells also have membranes around each organelle. In a eukaryotic cell, each of the Golgi apparatus, endoplasmic reticulum, and mitochondria may have an area of membrane comparable to that of the plasma membrane.

These biological membranes are composed of amphipathic lipids: molecules with hydrophilic head groups and hydrophobic tails. In solution, these lipids pair up tail-to-tail, forming a two molecule-thick barrier that is approximately 40 Å wide (Figure 2.2). Membranes are made of a variety of lipids, and different membranes have different compositions.

The most common lipids are phospholipids such as p(hosphatidyl)-ethanolamine, p-serine, p-choline, and p-inositol. These each contain two fatty acid chains attached, via a glycerol linker and a phosphate group, to a polar head group (which is the suffix, e.g. choline). The fatty acid tails for a given phospholipid type vary in length between 12 and 24 carbon atoms, and have different numbers of C=C *cis* bonds.



(a) **(b)**

Figure 2.2: a Atomic representation of two 'POPC', p-choline type phospholipids arranged tail-to-tail. **b** A biological membrane consists of many such lipids. Here a 120 × 120 Å section of membrane is shown with an embedded potassium channel (PDB code 2R9R, Long *et al.* [2007]). Each monomer of the tetrameric channel is coloured differently (red, pink, white, and blue). Images were produced using VMD [Humphrey *et al.*, 1996].

Glycolipids – lipids with sugars forming the head group – are found in both prokaryotes and eukaryotes. They are distributed only on the extracellular facing side of the membrane. This is an extreme example of membrane asymmetry, but it appears that most, if not all, membranes have different compositions of lipids on the inner and outer leaflets. This is perhaps unsurprising given that the purpose of a membrane is to maintain a division between chemically non-equivalent regions.

Cholesterol is an abundant non-lipid component of eukaryotic plasma membranes, but is absent in prokaryotic membranes. Its structure contains 4 linked rings that prevent the molecule

from flexing. Portions of the membrane that are enriched in cholesterol and the phospholipid sphingomyelin form 'lipid rafts', and it has been suggested that some membrane proteins may preferentially attach through these. Cholesterol makes a membrane less permeable to small water-soluble molecules [Bretscher and Munro, 1993], and helps to keep the membrane viscosity constant as temperature changes.

Archaea and Gram-negative bacteria have unusual membranes. Archaea eschew fatty acid tails in favour of isoprenoid tails, which are linked via ether (rather than ester) linkages to the polar head groups [Boucher *et al.*, 2004]. Gram-negative bacteria have two membranes, of which the outer is covered by lipopolysaccharides – long chains of sugar molecules attached to a class of lipids called 'lipid A' [Lugtenberg and Vanalphen, 1983].

In what follows, I often refer to 'the membrane', but 1) prokaryotic and eukaryotic membranes differ 2) The membranes of different eukaryotic cell types differ 3) Within one eukaryotic cell there is a variety of membranes 4) Each of these membranes is asymmetrical between the upper and lower leaflet 5) Variations have been found between the composition of the same membrane in related species [Bretscher, 1972] 6) Within the same membrane lipids are not thought to be distributed homogeneously.

There is not enough data, for our problem, to treat each of these membranes separately, but this is not necessary as the variation between membranes is expected to be small compared with the differences between membrane and soluble environments.

2.4 The membrane and sequence alignment

At their heart, all sequence alignment methods use a substitution table that contains scores for the alignment of different types of amino acids. The unique physicochemical environments within a biological membrane will alter the substitution preferences for transmembrane amino acids. Over the next two chapters sequence alignment for comparative modelling is improved by incorporating these altered preferences into an alignment method. In Chapter 3 separate

substitution tables for each local environment of a membrane protein are derived. Comparison of these tables suggests that substitution preferences are mainly driven by the hydrophobicity of the local environment. We show that use of these membrane specific tables can improve pairwise alignments and protein models within the twilight zone. However, these pairwise alignments are much less accurate than alignments which utilise information from multiple sequences. The incorporation of membrane specific substitution tables into a novel multiple sequence alignment program is the subject of Chapter 4.

CHAPTER 3

Environment specific substitution tables improve membrane protein alignment

The material in this chapter forms the basis of the following published paper:

Hill, J. R., Kelm, S., Shi, J., and Deane, C. M. (2011).
Environment specific substitution tables improve membrane protein alignment.
Bioinformatics, 27(13), i15 - i23

3.1 Introduction

Membrane proteins are both abundant and important in cells, but our understanding of them is restricted by the small number of solved structures. Here we consider whether membrane proteins undergo different substitutions from their soluble counterparts, and whether these can be used to improve membrane protein alignments – and thereby improve prediction of their structure.

We construct substitution tables for different environments within membrane proteins. As data is scarce, we develop a general metric to assess the quality of these asymmetric tables. Membrane proteins show markedly different substitution preferences from soluble proteins; for example, substitution preferences in lipid tail-contacting parts of membrane proteins are found to be distinct from all environments in soluble proteins, including inaccessible environments. A principal component analysis of our tables identifies the greatest variation in substitution preferences to be due to changes in hydrophobicity; the second largest variation relates to secondary structure. We demonstrate the use of our tables in pairwise target-template alignments of membrane proteins using the FUGUE alignment program. On average, in the $10 - 25\%$ sequence identity range, alignments are improved by 28 correctly aligned residues compared with alignments made using FUGUE's default substitution tables. Our alignments also lead to improved structural models.

3.1.1 Chapter overview

The membrane is a radically different environment from the aqueous environment of soluble proteins. As discussed in the previous chapter, most membrane bilayers are composed of phospholipids with hydrophobic tail groups and charged head groups. This suggests for example that substitutions from charged residues to hydrophobic ones are unlikely in head-contacting regions, and conversely that substitutions from hydrophobic residues to charged ones are unlikely in tail-contacting regions.

Thus, membrane proteins will have unique patterns of substitutions [Mokrab *et al.*, 2010]. However, due to lack of data, membrane proteins are typically analysed with substitution tables optimised for soluble proteins. Identifying appropriate tables for membrane environments is expected to improve methods that depend on them, particularly target-template alignment [Mokrab and Mizuguchi, 2005].

Substitution tables have previously been developed for transmembrane regions. The JTT table appears to be the earliest example [Jones *et al.*, 1994], with the PHAT [Ng *et al.*, 2000] and SLIM [Müller *et al.*, 2001] tables following. All these tables were intended to be used in conjunction with a non-membrane table, such as BLOSUM62 [Henikoff and Henikoff, 1992]. This 'bipartite' scheme requires a separate algorithm to decide where to use each table.

Sequence alignment has been attempted using PHAT, with both bad [Forrest *et al.*, 2006], and good [Pirovano *et al.*, 2008] results when compared with alignments using only BLOSUM62. The SLIM table is optimised for homology detection: its authors explicitly caution against using it for alignment.

All of these previous examples use only a single transmembrane table, which is perhaps overly simplistic: substitution preferences are known to also depend on features such as the secondary structure and accessible surface area of a residue. These features can be taken into account by creating environment-specific substitution tables (ESSTs) – a set of substitution tables, one for each combination of structural features or 'environment'.

To create an ESST, substitutions in each environment must be counted within columns of an alignment of related proteins. In early studies tables were constructed by counting substitutions between homologous proteins of known structure [Overington *et al.*, 1992; Shi *et al.*, 2001]. More recently, tables have been made by counting substitutions between one structure and many sequences [Mizuguchi *et al.*, 2007]. The latter method is used here.

Two problems complicate the construction of ESSTs for membrane proteins:

1. Although secondary structure and accessible surface area can be determined as for soluble proteins, the location of a residue within the membrane bilayer cannot be directly inferred

from the solved structure. Here we use the annotation program iMembrane to infer the local membrane environment (Kelm *et al.* [2009], see Section 1.5.4).

2. Once an ESST has been created, how can we tell if it is representative of its environment across all membrane proteins? Here we describe a metric of ESST quality that is robust against perturbations in the observed frequencies of individual substitutions.

Guided by our quality metric, we create 'good' membrane ESSTs and analogous soluble ESSTs, and make global comparisons between them. We construct a dendrogram to illustrate inter-table distances, and perform a principal component analysis to detect the dependence of substitution patterns on environment type. Membrane environments are found to be far more diverse than soluble protein environments. For example, the difference in substitution patterns between any pair of lipid tail layer environments is greater than the difference between soluble protein environments of the same secondary structure and accessibility.

FUGUE (Shi *et al.* [2001], see Section 1.7.3) is a commonly used program to produce target-template alignments with ESSTs. We compare the performance of FUGUE for membrane protein alignment when used with three sets of substitution tables: FUGUE's default soluble protein tables, our membrane specific tables, and a PHAT/BLOSUM62 pair of tables. For comparison we also performed alignments with MUSCLE, which does not use structural information [Edgar, 2004]. Our membrane-specific ESSTs consistently improve pairwise alignment accuracy, especially at low sequence identity. In the $10 - 25\%$ sequence identity range, 54/99 alignments are improved by at least 10 residues compared with the next best method (the default FUGUE tables), whereas only 6 alignments are worsened by the same amount. In this range the average improvement per alignment is 28 more residues aligned correctly. These alignment improvements are found to lead to corresponding improvements in structure prediction.

3.1.2 Substitution tables in more depth

A substitution table, such as that shown in Figure 3.1, scores the propensities for amino acids to mutate into one another. Typically the scores are obtained by applying Bayes' rule to determine the probability that two amino acids a and b are related by mutation [von Ohsen *et al.*, 2001]

$$P(\text{related}|a,b) = \frac{P(a,b|\text{related})}{P(a)P(b)}P(\text{related}). \tag{3.1}$$

The logarithm of $P(\text{related}|a,b)$ is usually reported, and the term $P(\text{related})$ is taken to be constant.

Determining the probabilities in this equation requires a method of inferring substitutions from a set of present day proteins. One of the oldest approaches to this is used in Dayhoff's Point Accepted Mutation (PAM) tables [Dayhoff *et al.*, 1978]. To build a PAM table, an evolutionary tree is inferred from a closely related set of sequences (originally sharing $> 85\%$ sequence identity), and the substitutions occurring at the nodes of the tree are tabulated. By repeating this procedure over many sequence sets and weighting the results by evolutionary distance, the probability of substitutions between amino acids over a normalised time period can be inferred, and probabilities over longer time periods can be extrapolated.

A problem with Dayhoff's approach is that the substitution probabilities depend on the accuracy with which the evolutionary tree is reconstructed. A given set of sequences may admit more than one 'most parsimonious' tree. Worse, as the set of sequences must be closely related, it is possible that the PAM tables only describe substitution probabilities at small evolutionary distances.

A descendant of Dayhoff's approach is the VTML table series, which can account for substitutions between distantly related sequences [Muller *et al.*, 2002]. The tables are built in an iterative process: at each step the current table is used to provide new distance estimates between the sequences, and these are used to build the next table until convergence.

In this thesis we follow the most common, BLOSUM approach to table construction [Henikoff

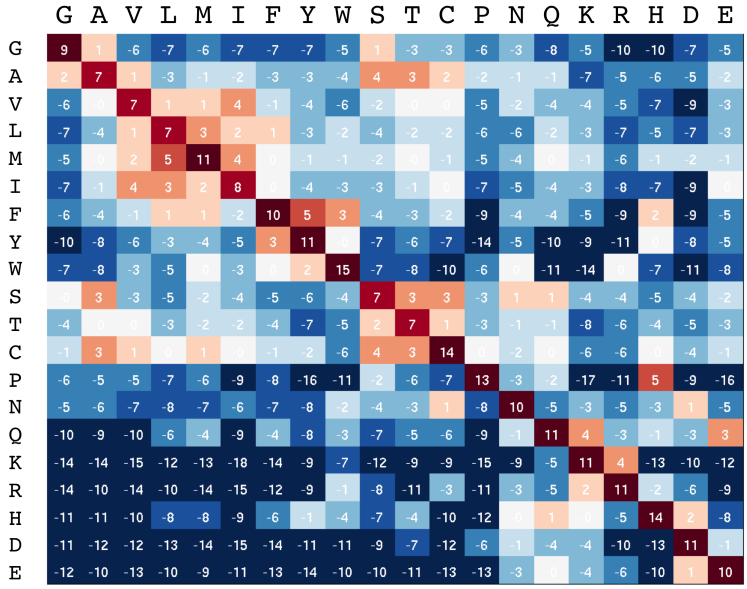


Figure 3.1: An example substitution table labelled by log-odds scores. High scores are coloured red, and represent substitutions that take place more often than expected by chance. The hot red diagonal shows that an amino acid is more likely to be conserved than substituted by another amino acid type.

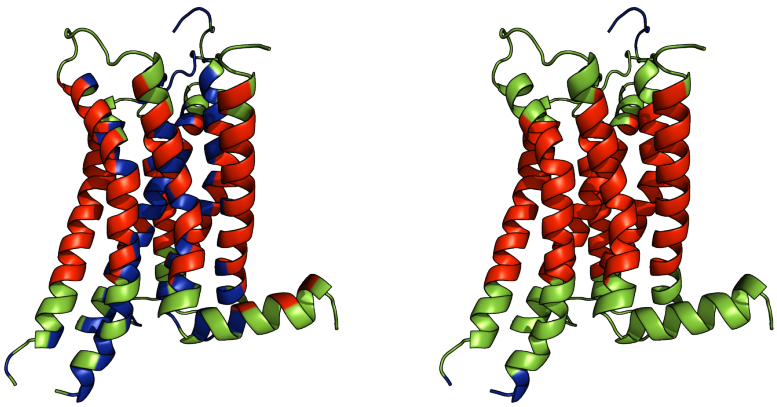
and Henikoff, 1992]. BLOSUM counts substitutions between alignments of divergent present day proteins – unlike PAM it infers no ancestry – and weights these counts to prevent bias from over-represented sequence families. The form of the weighting is simple: if two sequences share more than a threshold level of sequence identity, they are placed in the same cluster. The totality of amino acid substitutions between two clusters at a particular position adds up to one normalised substitution count. This scheme does not directly count the short timescale substitutions between sequences above the threshold level of identity, and treats all substitutions beneath this threshold equally. Nevertheless, for sequence alignment, BLOSUM tables have been found to be of comparable accuracy to VTML tables, and of greater accuracy than PAM tables [Edgar, 2009].

3.2 Methods

3.2.1 Environment descriptors

We use iMembrane to annotate each residue in a membrane protein by the part of the membrane it contacts. iMembrane uses coarse-grained molecular dynamics simulation data from the CGDB database [Scott *et al.*, 2008] to annotate three contact environments. Residues that are in contact with the membrane for less than 10% of the simulation time are annotated as non-membrane contacting (n); the remaining residues are annotated (h) if they spend more time in contact with lipid heads, and (t) if they spend more time in contact with lipid tails (Figure 3.2a). By taking a consensus of these contact annotations, a membrane protein can be divided into layers corresponding to the lipid heads (H), lipid tails (T), and not-in-membrane (N) regions (Figure 3.2b).

Secondary structure and accessible surface area are annotated using JOY (Section 1.5.1).



(a) **(b)**

Figure 3.2: A membrane protein structure colour-coded by iMembrane annotations **a** The contact annotation. Green regions are in contact with lipid head groups (h) and red regions with lipid tail groups (t). Blue regions are not in contact with the membrane (n), in this case because they face the core of the protein **b** The layer annotation divides the protein into slabs according to membrane positioning. Green regions are in the head layer (H), red regions are in the tail layer (T), and blue regions fall outside the slab model of the membrane (N). This structure is the transmembrane region of the human A2A adenosine receptor, (PDB code 3EML, Jaakola *et al.* [2008])

The secondary structure types used are helix (H), β-strand (E), coil (C), and +ve ϕ angle (P). By convention, a residue is deemed accessible (A) when more than 7% of its surface area is exposed, and otherwise is inaccessible (a) [Hubbard and Blundell, 1987].

Each environment is a combination of these annotations. There are thus $72 = 3$ (membrane contact n,h,t) $\times 3$ (membrane layer N,H,T) $\times 4$ (secondary structure H,E,C,P) $\times 2$ (accessibiliity A,a) distinct possible environments.

3.2.2 Alignments for table generation

Transmembrane protein structures were identified from the PDB_TM database [Tusnády *et al.*, 2005] (accessed on 10/09/10) and downloaded from the PDB [Berman *et al.*, 2000]. Each was then split into its component protein chains. Redundant chains – those with greater than 80% sequence similarity – were removed using cd-hit [Li and Godzik, 2006]. Chains without iMembrane search hits were also removed, leaving 328 chains.

For each chain, related sequences were obtained from 5 iterations of PSI-BLAST [Altschul *et al.*, 1997] using an e-value threshold of 1×10^{-3} for keeping a hit, and a threshold of 1×10^{-5} for including a hit in the sequence profile of the next iteration. PSI-BLAST searches were made against the NCBI nr database (accessed 04/09/10). These sequences were then aligned to their corresponding structures with MUSCLE, and the alignments used to generate the membrane substitution tables.

Soluble tables were generated from four different alignment sets. The first of these was made as above – that is, by aligning multiple homologous sequences with each structure. The structures were obtained by taking the first structure from each family in the HOMSTRAD database [Mizuguchi *et al.*, 1998a], and the sequences were found by searching the NCBI nr database (accessed 22/06/08). After filtering, this yielded 423 soluble chains which were used to produce our standard soluble tables. The other three alignment sets (SUB177, SUB371 and HOMSTRAD) are multiple structure alignments, and tables derived from them are used only to validate our standard soluble tables.

SUB177 and SUB371 are described in the original FUGUE paper [Shi *et al.*, 2001]. SUB177 is a set of 177 protein families comprising 706 structures used to build the default tables of FUGUE. SUB371 is a set of 371 protein families comprising 1357 structures used to test the stability of the SUB177-derived tables. The HOMSTRAD set comprises 1032 families and more than 3000 structures.

3.2.3 Table construction

We constructed ESSTs by counting substitutions between clusters of sequences (clustered at 60% sequence identity) from a multiple sequence alignment, in a process similar to that used to make BLOSUM tables (Section 3.1.2). Substitutions were counted by the JSUBST program and tabulated in an environment specific counts matrix A^E (where E labels the environment). Each matrix element A^E_{ba} is the weighted number of times that a substitution is observed from residue 'a' and environment E in a cluster containing a structure to residue 'b' in another cluster. Substitutions to and from gaps were not counted, but all columns in the alignments were included when constructing the matrices.

The entries of the ESST S^E were obtained from the following formula[1], which is derived following the principle of Equation 3.1:

$$S^E_{ba} = \frac{3}{log(2)} log \left(\frac{A^E_{ba} / \sum_b A^E_{ba}}{\sum_{a,E} A^E_{ba} / \sum_{a,b,E} A^E_{ba}} \right). \tag{3.2}$$

Given that the structure has a residue a, the numerator of the logarithm is the probability of a substitution $a \rightarrow b$ in the matched sequence. The denominator is the probability that any substitution in any environment will go to b rather than another residue. The prefactors (and the taking of the logarithm itself) are a standard rescaling. ESSTs are generally asymmetric ($S^E_{ba} \neq S^E_{ab}$), and are rounded to the nearest integer.

[1] Before calculating S^E, a small constant of 1/100 of a count was added to each entry A^E_{ba} to prevent S^E_{ba} evaluating to $-\infty$ in rare cases.

3.2.4 Identifying consistent tables

How can we identify substitution tables that are unrepresentative of their environments? A crude method is to label as unrepresentative all those tables with fewer than a minimum number of counts. However, this method can run into problems – a rare environment might be extremely consistent in the substitutions it allows, such that the number of counts is small, but the data is representative.

Here we use a combination of a count threshold and a 'self-consistency' score. The latter is obtained as follows. By normalising the columns of a counts matrix A^E_{ba}, we can interpret each entry as the probability that $a \to b$ in environment E.

When a vector of amino acid counts is multiplied by this matrix, it changes according to the substitution probabilities encoded in the matrix. After a large number of rounds of substitution (matrix multiplications), the resulting vector of amino acid counts is invariant under substitution. Mathematically, this vector is the eigenvector of the matrix with eigenvalue $+1$.

It is assumed that the distribution of amino acids in a given environment, averaged over all proteins, is stable over time. Thus, a representative table should have a limiting distribution of amino acids that is close to the distribution observed in the alignments used to construct it.

The self-consistency score, 'Q' is calculated as

$$Q = 1 - \frac{1}{2}\sum_{i=1}^{20} |v_i - w_i|, \qquad\qquad w_i \approx \sum_a A^E_{ia} / \sum_{a,b} A^E_{ba}, \tag{3.3}$$

where v is the normalised eigenvector of the probability matrix with eigenvalue $+1$, and w is a normalised vector of the observed amino acid frequencies, which can be estimated as shown. This score has the desirable property of taking values between 0 (totally inconsistent) and 1 (identical).

A simple interpretation of this score exists. It is the maximum fraction of residues that

could remain the same if substitutions occurred according to the probabilities encoded in the counts matrix over many iterations.

The self-consistency score is scale-invariant, so it provides a measure of table quality that is independent of the number of counts. Figure 3.3 shows a useful scheme for visually identifying poor tables. The fraction of the total number of counts and Q are plotted for each table with increasingly large subsets of the data. A stable counts matrix should tend to a stable level of Q as more data is included.

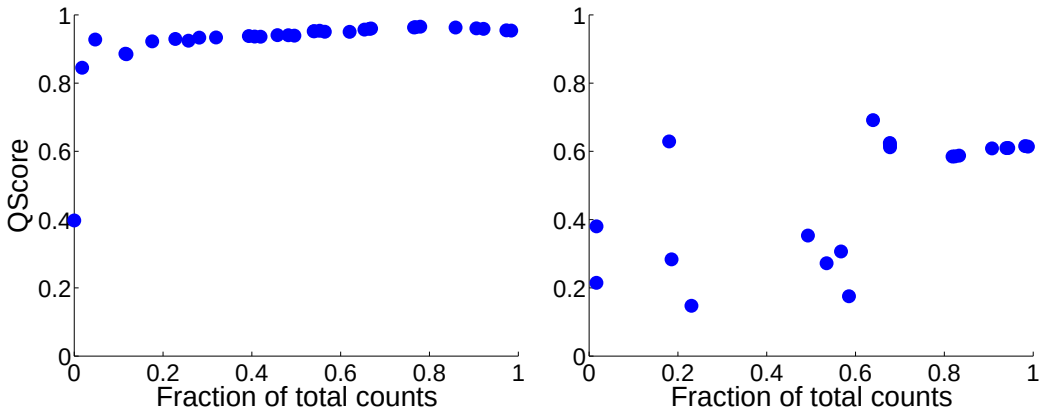


Figure 3.3: A high quality table (left) and low quality table (right). Each point plots the fraction of total counts and consistency of a table when constructed with 20 more alignments than the preceding point. Some points are superimposed. The high quality table is constructed for accessible residues in helical secondary structure lying at the interface between the membrane head and tail layers. The low quality table is constructed for inaccessible residues with +ve ϕ angle annotation in the membrane tail layer.

3.2.5 Table analysis and visualisation

The relative similarity of tables was visualised in two ways. Firstly a dendrogram was constructed based on the Euclidean distance between ESSTs. The dendrogram was built using single linkage clustering – meaning that new branches join existing clades based on the smallest distance between a member of the clade and the new branch. This linkage has the advantage that the dendrogram does not change under a rescaling of the data.

Secondly, following the example of Gong *et al.* [2009] on soluble tables, a principal component analysis in multi-dimensional 'substitution space' was performed [Hotelling, 1933]. This selects a set of 2 or 3 orthogonal axes that explain the greatest amount of variation in the data, and thus projects substitution space down into 2D or 3D with minimal distortion.

3.2.6 Testing of alignment accuracy

We tested target-template alignment on pairs of homologous membrane proteins with known structure taken from the Medeller test set [Kelm *et al.*, 2010]. We used one element of each pair as the target, and the other as the template, and filtered the set such that no two templates, and no two targets, had more than 80% sequence identity. This left 408 pairs of proteins ranging from $0 - 100\%$ identity, with a median sequence identity of 14%.

Alignments were made using FUGUE with the default tables, the PHAT/BLOSUM62 tables, and our membrane tables. Annotations of the template by iMembrane and JOY determined where each table was to be applied. Pairwise alignments were also made using the MUSCLE program. The quality of these alignments was assessed against the implicit sequence alignments generated by the structure alignment program TM-align (Section 1.6, Zhang and Skolnick [2005]). An example of this procedure is given below in which 9 residues are correctly aligned over a total alignment length of 10 residues:

```
TM-align
  Template Structure   --AGGA-CGPAA ...
    Target Structure   AAAGGAFCA-AL ...

Tested method
  Template Structure   --AGGA--CGPAA ...
     Target Sequence   AAAGGAFC-A-AL ...
             Correct?  --YYYYYNNYYYY
```

3.3 Results

3.3.1 Validation of substitution tables

The default tables used in FUGUE were obtained by counting substitutions between homologous structures. Due to the scarcity of membrane protein structures, we counted substitutions between a structure and related sequences, following a similar method to that of Mizuguchi *et al.* [2007]. To assess the validity of this procedure we compared eight soluble ESSTs generated by this method (our 'sequence derived tables') with those derived from the SUB177, SUB371 and HOMSTRAD structure sets (Table 3.1).

We defined soluble environments by a combination of secondary structure (H, E, C, P) and accessible surface area (A, a) annotations (see Section 3.2.1 for a description of these annotations). This led to 8 possible environments, which can be conveniently referenced by a letter code e.g. EA for the β-strand, accessible environment. To achieve consistency with later notation for membrane proteins we prefix each soluble environment with the letter 's' (e.g. sEA).

Of the eight tables, larger differences were seen in the sEa (soluble, β-strand, inaccessible),

Table 3.1: Comparison of soluble substitution tables derived from sequence and structure alignments

Quality measure	sCa	sCA	sEa	sEA	sHa	sHA	sPa	sPA
Counts (1000s)	570	1 358	625	460	752	1 099	84	369
Q	0.95	0.98	0.95	0.98	0.95	0.99	0.71	0.91
SUB177	98	12	119	25	77	10	156	103
SUB371	57	7	78	19	41	1	138	99
HOMSTRAD	32	5	75	5	31	0	140	78

Counts is the number of normalised substitution counts used in constructing our sequence derived tables, and Q is the self-consistency score for these, calculated as described in Section 3.2.4. The SUB177, SUB371 and HOMSTRAD rows list the number of entries in these soluble structure derived tables (out of a possible 400) that differ by more than 2 log-odds units from our soluble sequence derived tables. This 2 unit threshold is chosen to be a reasonable measure of dissimilarity. Differences between sequence and structure derived tables decrease as the number of families included in the structure set increases.

sPa (soluble, +ve ϕ, inaccessible), and sPA (soluble, +ve ϕ, accessible) environments due to the greater number of rare substitutions in these environments. As the scores are logarithmic, small variations in the number of rare substitutions lead to disproportionately large effects on their log-odds scores.

As expected, the self-consistency score, Q, was highest when there were fewest differences with the structure derived tables. The small number of differences between the structure and sequence derived tables, particularly when larger numbers of structures were used, suggests that sequence derived tables are representative of substitution preferences. Below, only the sequence derived tables are compared.

3.3.2 Membrane environment selection

The annotations described in Section 3.2.1 can be combined to give 72 possible environments. Larger numbers of environments lead to more specific substitution tables at the cost of each table being constructed with less data, and so potentially being of lower quality. Making tables for all 72 environments resulted in many tables having low self-consistency scores (35 tables had $Q < 0.9$). We thus sought to find a minimal set of environments that encompassed as much variation in substitution patterns as possible.

Intuitively, the membrane contact and membrane layer annotations contain similar information. Discarding the contact annotation led to tables with higher average consistency than discarding the layer annotation, so we largely ignored the contact annotation. There were two exceptions to this: accessible residues that lie in the tail layer but rarely contact the membrane can be identified as residues that line a pore (P). Accessible residues that are annotated with head contacts but are in the tail layer, or with tail contacts but are in the head layer, define an interface region (I) spanning the hydrophilic and hydrophobic parts of the membrane. We added these to the existing layer types to give five labels: H(ead), N(ot in membrane), T(ail), P(ore) and I(nterface) regions (Figure 3.4). This new system of layer types reduced the number of distinct possible environments to $32 = (3 \times 4 \times 2) + (2 \times 4 \times 1)$, as all layers could assume

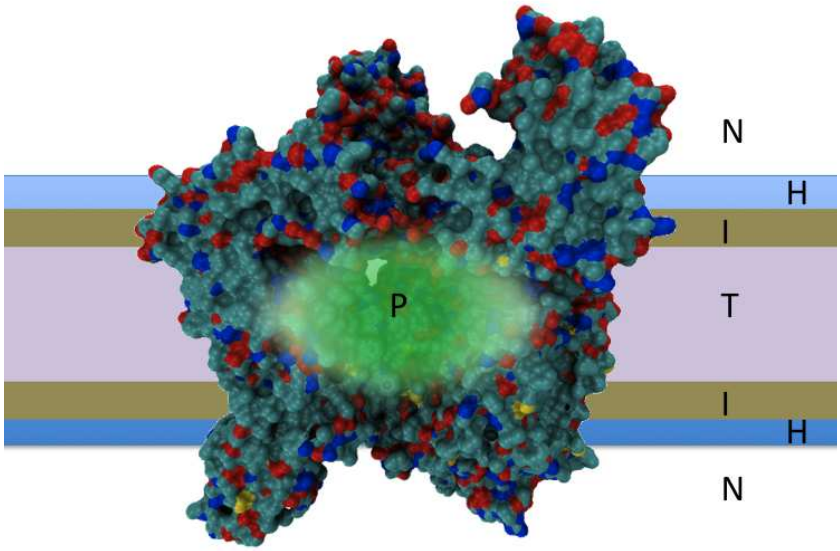


Figure 3.4: A schematic slice through a membrane protein in the membrane indicating the layer types used. 'N' is the region outside the membrane, 'T' and 'H' span the tail and head groups of the membrane lipids respectively, 'P' is the area lining the pore, and 'I' is the interface region between the tail and head groups. The depicted structure is particulate methane monooxygenase (PDB code 1YEW, Lieberman and Rosenzweig [2005]). This figure was made with VMD.

each of the 4 secondary structure types (H,E,C,P), but only the 3 original layers (H,N,T) could have both accessible and inaccessible residues.

It is convenient to refer to environments using a three letter code, as in Section 3.3.1, e.g. 'IEA' = interface layer, β-strand, accessible residues. Letter codes always take the form layer (H,N,T,P,I) : secondary structure (H,E,C,P) : accessibility (A,a). An asterisk '*' is used when the exact letter does not matter. Under this system 'T**' refers to all tail layer environments, whereas 'T*A' refers to accessible tail layer environments.

Many +ve ϕ angle tables (*P*) suffered from low self-consistency scores, low count numbers, and poor stability. For example, the TPa environment of Figure 3.3 had a Q score of 0.64 from 10 060 counts. Low self-consistency scores are to be expected: the majority of substitutions in these environments involve glycine, and other substitutions may be too rare to be representative. To increase table quality, all accessible +ve ϕ environments were merged into a single

environment *PA → NPA, and similarly *Pa → NPa for inaccessible +ve ϕ environments. The combined environments are labelled 'N' layer so as to maintain a consistent notation.

Self-consistency scores and total numbers of substitutions for each table in the resulting set of 26 environments are shown in Table 3.2. As with the soluble tables in Table 3.1, accessible environment tables (**A) tend to have higher self-consistencies than inaccessible environment tables (**a).

Table 3.2: Self-consistency scores and number of counts for each membrane environment specific substitution table. Environment labels are described in Section 3.2.1.

ESST	Q	Counts	ESST	Q	Counts
NPa	0.72	36 437	PCA	0.96	45 654
NPA	0.88	146 253	PHA	0.96	96 771
TCa	0.90	46 512	HHA	0.97	147 118
NEa	0.92	137 556	THA	0.97	265 566
NHa	0.92	118 306	HEA	0.97	109 238
PEA	0.92	60 677	THa	0.97	171 736
TCA	0.92	44 326	HCA	0.98	228 302
HCa	0.93	65 124	TEA	0.98	211 862
HHa	0.93	63 665	NCA	0.98	350 138
NCa	0.93	113 341	IHA	0.98	56 569
ICA	0.95	44 362	IEA	0.98	34 660
HEa	0.95	48 262	NEA	0.98	148 662
TEa	0.95	102 801	NHA	0.98	253 458

3.3.3 Clustering of tables

The Euclidean distance between the log-odds tables was used to create a 'family-tree' of the different environments (Figure 3.5). When calculating the distance, each substitution was normalised by its standard deviation across all the tables. This prevented the distance measure from being dominated by a handful of extreme substitution changes.

It has been suggested that loops of membrane proteins that extend above and below the membrane behave similarly to loops in soluble proteins (e.g. Tastan *et al.* [2009]). We also see

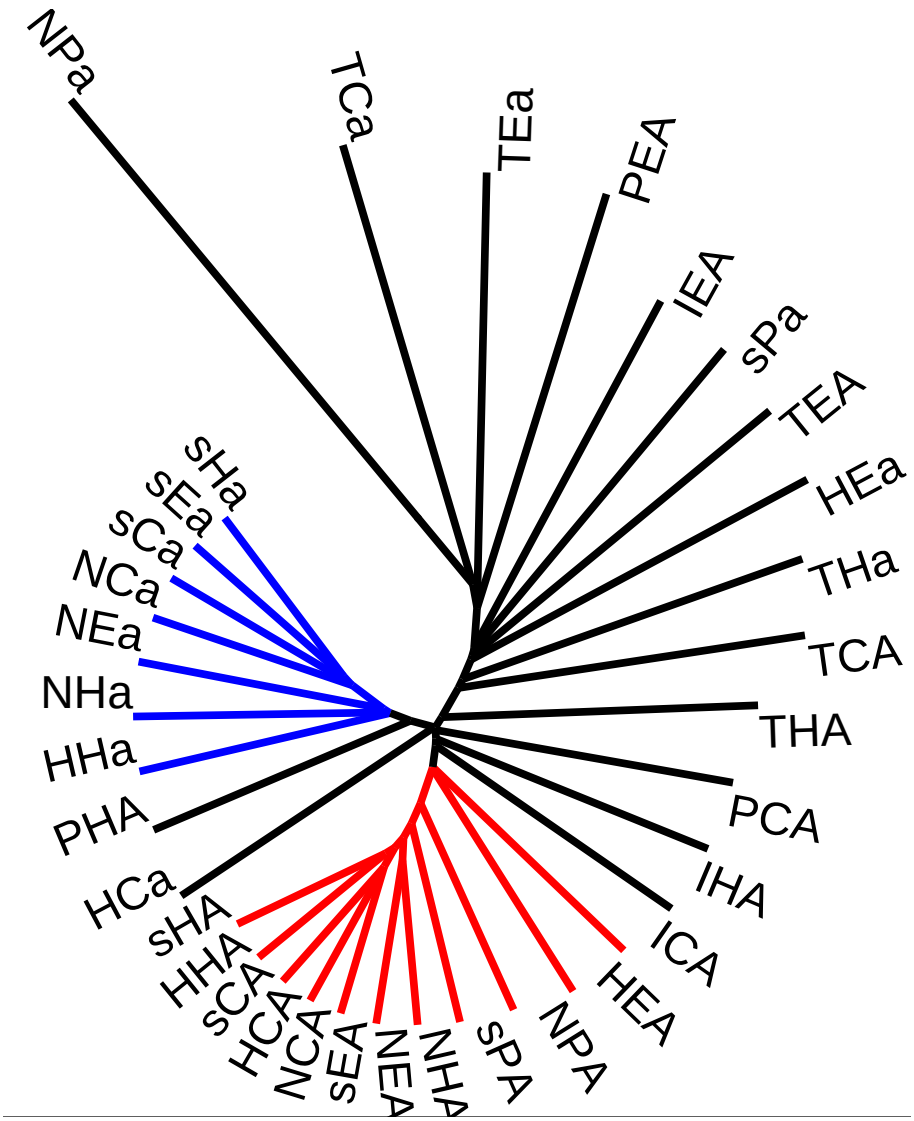


Figure 3.5: Dendrogram of ESSTs constructed by single-linkage clustering. A split is seen between accessible (red) and inaccessible (blue) environments. Tail-layer environments (T**) appear not to cluster. Note that here, as elsewhere, 'NPa' and 'NPA' refer to combined +ve ϕ environments that include residues in the transmembrane regions (see Section 3.3.2).

this in our results, where each table of the form NC* clusters with its sC* counterpart. As might be expected, the not-in-membrane tables (N**) are most similar to their soluble equivalents (the notable exception being that sHA clusters with HHA rather than NHA).

The tail-contacting environments are clear outliers, and do not cluster. Additional outliers are +ve ϕ environments (*P*), and some β-strand environments (*E*). This last may be because much of our β-strand data came from outer-membrane porins of Gram-negative bacteria. In Gram-negative bacteria, the outer membrane is extremely asymmetric: the inner leaflet is composed of phospholipids whereas the outer leaflet is composed of lipopolysaccharides. Additionally, inward-facing solvent-exposed residues are in contact with the periplasm rather than the cytosol. Evidence for the uniqueness of the β-strand environments can also be seen in their composition. Compared with other membrane environments, accessible β-strands within the membrane rarely contained cysteine, and the TEA environment was abundant in tyrosine.

The remaining environments separate by accessibility. Surprisingly, within the inaccessible clade (Figure 3.5, blue), the soluble secondary structure environments are more similar to each other than to their membrane equivalents. An accessible clade in Figure 3.5 is coloured red from the same level as the inaccessible clade. The pore-lining and interface environments lie just beyond these clades, suggesting that these environments have distinct properties, and therefore that their use is sensible.

A PCA plot allows patterns in substitutions to be discerned. Figure 3.6 accounts for 48% of the variation in the data with 3 principal components. Figure 3.6c shows that the differences between accessible and inaccessible environments cause most of the variation between tables – they are largely separated along the first principal component (the main exceptions being accessible tail-layer tables, T*A). This first component can broadly be identified as a measure of 'hydrophobicity'. Looking at the labelled points in Figure 3.6a, as the first principal component increases we move from tail layer to interface layer to head layer inaccessible environments, corresponding to decreasing hydrophobicity.

The second principal component appears to relate to secondary structure. Moving from

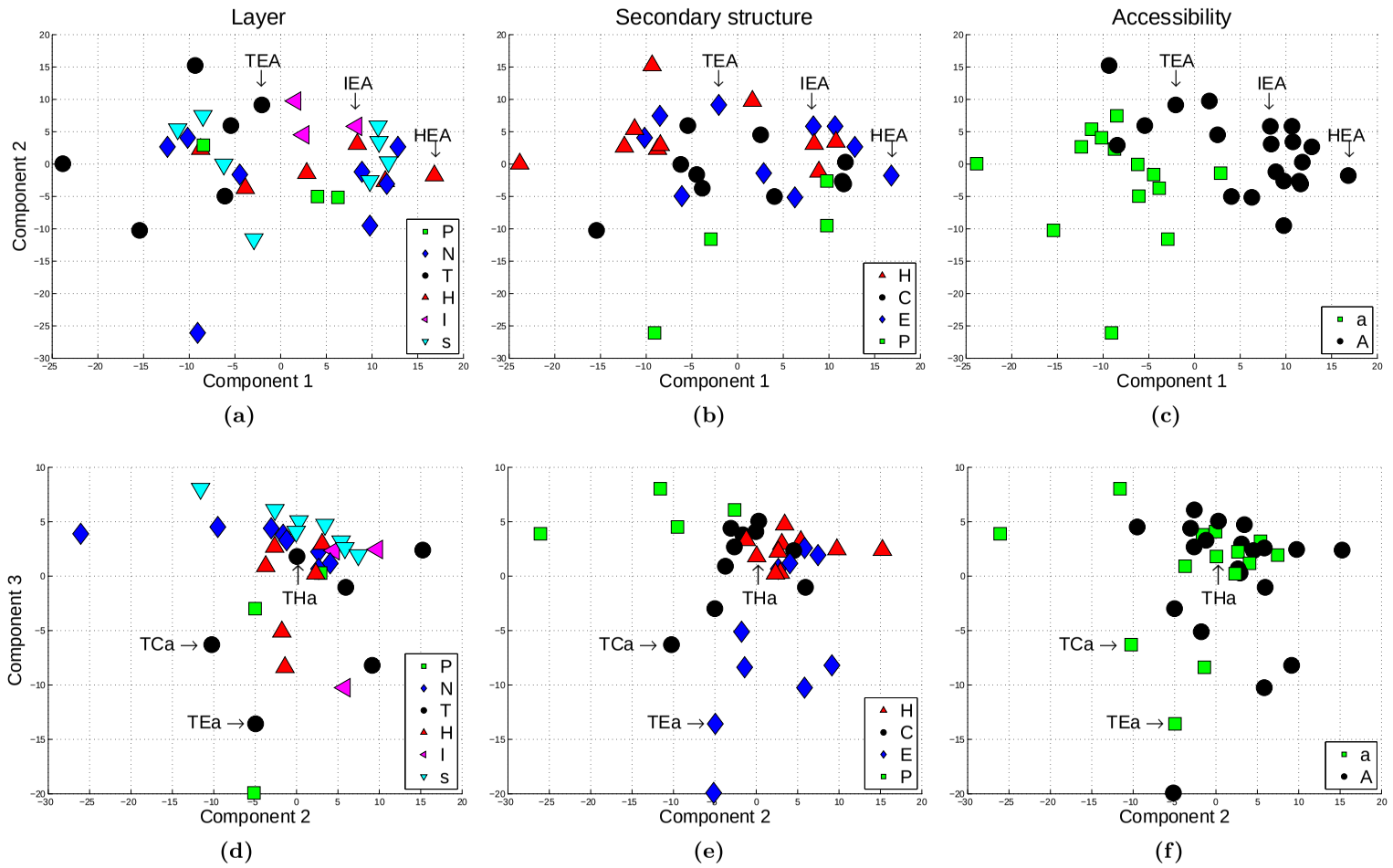


Figure 3.6: Principal component analysis of ESSTs. The top row and the bottom row are views of the same data along different principal components. The columns colour-code the data-points by layer type, secondary structure, and accessibility respectively. This allows the three-letter table code of each point to be read off from left to right. The labelled tables are ordered by secondary structure in the second principal component – reading panel (e) from left to right we first encounter TCa, then TEa, then THa. A similar ordering holds for other layer and accessibility types.

left to right in Figure 3.6e, we encounter the labelled points in the order TCa, TEa, THa as the second component increases. The same ordering is found for other layer types within the membrane. However, for soluble and not-in-membrane environments the order instead runs coil tables, helix tables, β-strand tables (e.g. sCa, sHa, sEa).

The bottom row of plots shows that different secondary structure environments cluster in the second and third components. The third principal component appears to be dominated by the differences between β-strand environments.

3.3.4 Target-template alignment

The previous section discussed the variations in substitution preferences in different environments. Now we demonstrate that a knowledge of these differences improves target-template alignment.

Pairwise alignments were made with the sequence alignment program MUSCLE, and the threading program FUGUE. Three different sets of substitution tables were used in FUGUE a) the default soluble tables, b) our membrane tables, and c) the PHAT/BLOSUM62 tables in a bipartite scheme. In this last case, PHAT was applied to residues with a 'T' layer annotation (including pore-lining residues), and BLOSUM62 was used elsewhere.

As the same program, FUGUE, was used with each set of tables, fair comparisons can be made between them. Gap penalties were determined separately for each set of tables.

3.3.5 Gap penalty determination

In the case of FUGUE, the optimal alignment is that which maximises the sum of the table entries S^E_{ba} for each pair of aligned residues. Not all residues will align, even between very similar proteins, and penalties to the alignment score must be determined for introducing gaps into the alignment. FUGUE distinguishes between several types of gaps (see Shi *et al.* [2001] for details). Gaps are penalised in order of severity as follows:

1. Gap within a secondary structure element (H)

2. Gap at the end of a secondary structure element (L)

3. Gap in a loop region (VL)

4. Gap at a terminus (VVL)

There are actually 8 types of gap penalty: each of the above categories can initiate a gap or be an extension of an existing gap. Initiating a gap results in a larger penalty than continuing an existing gap: the alignment is thus biased to a small number of large insertion/deletion events rather than a larger number of smaller events.

Table 3.3: Gap penalties for each set of tables used with FUGUE

	initiation				extension			
Tables	H	L	VL	VVL	H	L	VL	VVL
Default	28	20	20	8	4	4	2	2
Membrane	35	25	20	8	4	4	2	2
PHAT/BLOSUM62	26	24	16	8	6	4	2	2

A subset of 72 protein pairs was selected at random from the 408 pairs of proteins in the alignment dataset (see Section 3.2.6), and alignments made with perturbations of the default FUGUE gap penalties. Perturbations were made such that gap opening penalties were at least as large as gap extension penalties, and such that more 'severe' gaps had penalties at least as large as less 'severe' ones. The size of the perturbation steps ranged from 1–5 units and depended on the size of the default penalties. The alignment quality with the default FUGUE tables differed little as the penalties were changed. In view of this, and as most users are unlikely to change the gap penalties, the default penalties were kept.

For the membrane tables, only the two secondary structure gap initiation penalties were found to substantially differ from the default values (Table 3.3). The increase of penalties in these cases is to be expected, as the sizes of transmembrane secondary structure elements are constrained by the membrane thickness. The subsequent analysis uses these revised penalties,

but membrane tables with the default gap penalties lead to similar results. The PHAT/BLOSUM62 tables are scaled differently from the others, so their penalties are not directly comparable.

3.3.6 Alignment accuracy

Alignments were made for the remaining 336 pairs of proteins in the alignment dataset. None of the methods performed well in the $0-10\%$ sequence identity range, but beyond this the membrane tables gave a consistent alignment advantage. At $> 35\%$ sequence identity, the alignments show few differences. Figure 3.7 compares the alignments of membrane specific tables to common alternatives. PHAT/BLOSUM62 is omitted for clarity – its performance is comparable to that of default FUGUE.

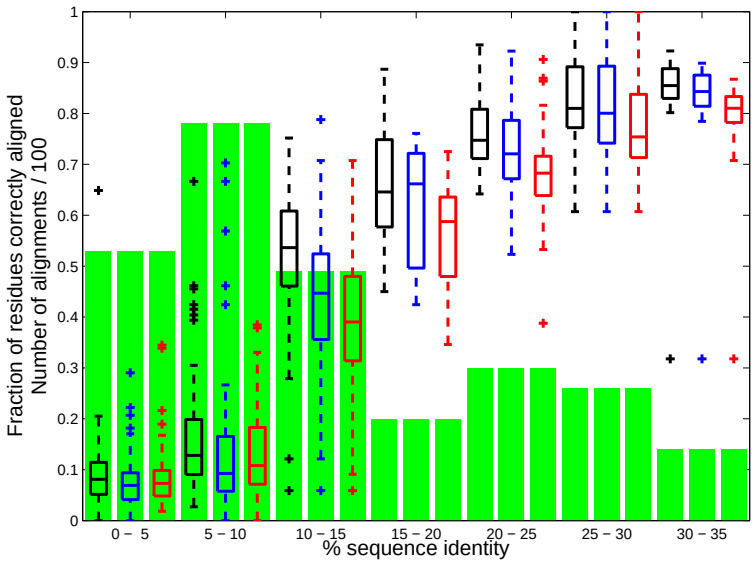


Figure 3.7: Box plots of the fraction of residues aligned correctly as sequence identity increases. There are three boxes at each sequence identity, corresponding to membrane FUGUE (black), default FUGUE (blue), and MUSCLE (red). The green bars show the number of alignments divided by 100. For example, there are 78 alignments in the $5-10\%$ sequence identity range.

Consideration of the outliers in Figure 3.7 is informative. In the $30-35\%$ sequence identity range, the three methods in the figure appear to have correctly aligned only $\sim 30\%$ of the residues in one target-template pair (PDB codes 1SU4A (target), 1XP5A (template)). In fact,

these proteins are an identical rabbit Ca^{2+}ATPase in different conformations. In this case the TM-align rigid-body structure alignment does not capture local similarities, leading to the $30-35\%$ sequence identity figure, and the low assessment of performance of the sequence alignment methods. Outliers in the $0-10\%$ sequence identity range are mostly due to short alignment lengths.

Figure 3.7 gives a broad picture of performance differences, but does not distinguish between a small alignment improvement on a short protein and a much larger improvement on a bigger protein. Table 3.4 lists the number of times that 'membrane FUGUE' correctly aligns at least 10 residues more (Win) or fewer (Loss) than another method.

Table 3.4: Alignment quality of membrane tables vs other methods

% Identity	Number of Alignments	*membrane FUGUE vs* default FUGUE		MUSCLE		PHAT/ BLOSUM62	
		Win	Loss	Win	Loss	Win	Loss
0–5	53	12	5	14	4	11	3
5–10	78	32	10	29	8	30	6
10–15	49	36	4	43	1	33	6
15–20	20	10	2	14	0	11	2
20–25	30	8	0	18	0	12	3
25–30	26	4	1	14	0	6	1
30–35	14	1	0	6	0	3	1
> 35	66	1	2	3	1	1	0
Total	336	104	24	141	14	107	22

For each sequence identity range, the number of alignments where membrane FUGUE correctly aligns at least 10 more (Win) or 10 fewer (Loss) residues than the named alternative method is listed. For example, in the $10-15\%$ sequence identity range, membrane FUGUE correctly aligns at least 10 more residues than default FUGUE in 36 out of 49 alignments.

Membrane FUGUE often improved alignment by more than 10 residues. Table 3.5 gives the number of correctly aligned residues across all the alignments in each sequence identity bracket. Membrane FUGUE outperforms all other methods in all brackets, except at $> 35\%$ sequence identity where the differences between the methods are marginal.

If the alignment set is divided into α and β proteins the same trends in accuracy are seen for both, with membrane FUGUE outperforming the other methods. The principal difference is the scarcity of β-type alignment pairs at higher sequence identities.

Table 3.5: Number of correctly aligned residues for each set of tables

% Identity	Number of residues	membrane FUGUE	default FUGUE	MUSCLE	PHAT/ BLOSUM62
0–5	12 915	**1 132**	872	910	1 007
5–10	22 734	**3 571**	2 555	3 001	2 721
10–15	24 349	**12 915**	10 819	9 893	10 427
15–20	7 576	**5 042**	4 697	4 145	4 467
20–25	9 156	**6 900**	6 607	6 145	6 565
25–30	5 644	**4 608**	4 522	4 300	4 479
30–35	4 792	**3 448**	3 403	3 274	3 402
>35	18 881	17 578	**17 586**	17 547	17 545
Total	106 047	**55 194**	51 061	49 215	50 613

3.3.7 Structure prediction

Models were built with MEDELLER for each of the 336 default FUGUE, and membrane FUGUE alignments. Models were also built for the implicit sequence alignments from TM-align.

MEDELLER provides different model-building options that prioritise accuracy or coverage. However, the relative quality of the models produced by different alignment methods showed little sensitivity to the model-building details. Results described below are for the default 'high-accuracy' models, but results for the 'naive' and complete models are similar.

Reasonable alignments were only achieved above 15% sequence identity (Figure 3.7), and beyond 35% identity alignments differed little between methods. In the $15 - 35\%$ range the average RMSD between the model and the native structure was: 3.4Å (membrane FUGUE), 4.1Å (default FUGUE), 2.0Å (TM-align). The mean sequence identity was 24%.

3.4 Discussion

We constructed substitution tables for membrane proteins by aligning single structures to multiple homologous sequences. This method, already used in the literature, allows a small number of structures to be leveraged to build tables at the cost of increased error in table construction. To address this problem, we suggested a method of assessing the quality of such tables, and used this method to build tables that were stable and consistent with the data used to construct them.

A principal component analysis of the individual tables revealed that residues in contact with lipid-tails have some substitution preferences typical of hydrophobic regions. However, the differences in other substitution preferences mean that membrane proteins are not simply 'inside out'.

Globally, it appears that accessibility is the primary determinant of membrane substitution preferences, followed by secondary structure. Position within the membrane has a less clearly-defined, but substantial effect. Membrane tables showed greater variability than their soluble equivalents, suggesting that an environment-specific approach to membrane protein modelling will yield greater improvements than did the environment-specific approach to soluble protein modelling.

Evidence for this supposition was found in a set of 336 pairwise target-template alignments made by MUSCLE and FUGUE. MUSCLE makes no use of structural information, so it is unsurprising that it performed worst at target-template alignment. The default FUGUE tables and the bipartite PHAT/BLOSUM62 alignments performed better than MUSCLE and comparably to each other. Each makes use of different structural information – the default tables take into account the accessibility, secondary structure, and hydrogen-bonding of a residue; whereas PHAT/BLOSUM62 distinguishes between residues inside and outside the membrane.

Conflicting accounts of the performance of PHAT have previously been reported. It has been suggested that this is due to bad alignments when PHAT is mistakenly applied to non-

transmembrane residues [Pirovano *et al.*, 2008]. The good alignments here can most likely be attributed to the quality of the transmembrane annotation from iMembrane.

Our membrane tables distinguish between both membrane location, and secondary structure and accessibility. Compared with the best performing alternative tables, the use of the membrane tables led to 104 of the 336 alignments having at least 10 more correctly aligned residues, with only 24 alignments being worse by the same margin. These improved alignments translate into predicted structures with a lower average RMSD to the native structure (3.4Å membrane FUGUE, 4.1Å default FUGUE) within the $15 - 35\%$ sequence identity range.

Alignment might be improved by an iterative approach to table construction. The tables presented here were generated by counting substitutions between homologous sequences aligned to a single structure by MUSCLE. Instead, these alignments could be made by FUGUE using the membrane tables. The resulting improved substitution tables could then be used to realign the sequences. This procedure could be iterated until convergence.

Our substitution tables, which take into account the environments of residues in membrane proteins, substantially improve alignments between membrane protein sequences and structures. In turn, these improved alignments lead to better structural models.

Nevertheless, these results represent only a proof-of-principle for this approach. To demonstrate the method we have considered only pairwise alignment, but in the next chapter we develop MP-T, a program that includes multiple sequence information. MP-T also benefits from a bipartite scheme of gap penalties in which insertions and deletions are punished more severely in the transmembrane region.

CHAPTER 4

MP-T: improving membrane protein alignment for structure prediction

The material in this chapter forms the basis of the following published paper:

Hill, J. R. and Deane, C. M. (2013).

MP-T: Improving Membrane Protein Alignment for Structure Prediction.

Bioinformatics, 29 (1), 54 - 61

4.1 Introduction

As described in the previous chapter, membrane proteins are clinically relevant, yet their experimental structures are rare. Models of membrane proteins are typically built from template structures with low sequence identity to the target sequence, using a target-template alignment as a blueprint. This alignment is usually made with programs designed for use on soluble proteins. Biological membranes have layers of varying hydrophobicity, and membrane proteins have different amino acid substitution preferences than their soluble counterparts. Here we include these factors into an alignment method in order to improve target-template alignments, and consequently improve membrane protein models.

4.1.1 Chapter overview

In the absence of experimental structures, a model of a target sequence may be built by comparative modelling. This technique can be divided into three phases (Section 1.1): 1) Identifying a template ('fold recognition'), 2) Aligning the sequences of the target and template, 3) Generating atomic coordinates for the target from this alignment. Many methods perform fold-recognition and alignment simultaneously. In this chapter we maintain a distinction and focus solely on alignment.

Alignment methods differ in how they use information about protein structure and sequence. Sequence information is derived from a set of sequences that are homologous to the target and template. These sequences are often combined into a profile. Structure information is commonly incorporated either by the use of statistical potentials (e.g. Yang *et al.* [2011]), or by annotation of the template sequence e.g. with secondary structure [Shi *et al.*, 2001].

The alignment of membrane proteins presents unique challenges. Most soluble proteins are globular with a hydrophilic surface, whereas membrane proteins possess neither of these properties. Alignment methods incorporating statistical potentials derived from soluble proteins are thus expected to perform poorly on membrane proteins. At least two such methods,

pGenThreader [Lobley *et al.*, 2009] and SPARKS-X [Yang *et al.*, 2011], recognise this and filter transmembrane sections from their results.

Alignment methods that annotate a template include HHsearch [Söding, 2005] and PROMALS [Pei and Grishin, 2007] (see Section 1.7.3). These align pairs of profiles that are annotated with predicted secondary structure. PROMALS may also be used as a homology-extension method. In this approach a multiple sequence alignment is performed, but with each sequence replaced by a profile.

Most multiple sequence alignment (MSA) methods, such as MAFFT [Katoh and Toh, 2008], MSAProbs [Liu *et al.*, 2010], and 'default' T-Coffee [Notredame *et al.*, 2000] do not use any structure information, but instead derive their accuracy from a 'consistency' criterion and/or iterative refinement (see Section 1.7.2). Consistency based approaches aim to generate a MSA that accords best with a library of pairwise alignments between the sequences being aligned.

A small number of alignment methods have been designed specifically for membrane proteins. The AlignMe program has been developed to study LeuT-fold transporters [Khafizov *et al.*, 2010]. Praline^{TM} [Pirovano *et al.*, 2008] and TM-Coffee [Chang *et al.*, 2012] are homology-extension (multiple profile alignment) methods that have been found to perform well on alignments of transmembrane proteins from the BALiBASE benchmark [Bahr *et al.*, 2001]. Neither method uses secondary structure to aid alignment, but Praline^{TM} uses different scoring in regions annotated as being transmembrane.

Due to the scarcity of known structures, the best template for a membrane protein sequence is likely to have low sequence identity – often beneath the $\sim 30\%$ 'twilight zone'. Fortunately, even at this low level of sequence identity, accurate alignment may be possible as biological membranes have a sandwich structure with a hydrophobic middle (lipid tail layer) and hydrophilic edges (lipid head layer). These differing hydrophobicities constrain the amino acids likely to be found in each environment [Forrest *et al.*, 2006].

In the previous chapter we demonstrated that environment specific substitution tables improved the pairwise alignment of membrane proteins. Here we incorporate these tables into a

consistency-based multiple sequence alignment program, MP-T (Membrane Protein Threader). The evolutionary information provided by homologous sequences greatly increases alignment accuracy at low sequence identities compared with pairwise alignments.

Our approach involves no sequence weighting or iterative refinement yet is as accurate as leading profile-profile aligners such as HHsearch [Söding, 2005] and PROMALS [Pei and Grishin, 2007] whilst introducing fewer misaligned pairs of residues. MP-T leads to models with significantly higher GDT_TS [Zemla *et al.*, 2001] than the best alternative alignment method tested – 1/4 of models see an increase in GDT_TS of at least 4%.

4.1.2 Approach

In the previous chapter we performed pairwise target-template alignments. Our pairwise alignment method used structural information by annotating residues in the template with the region of the membrane that they contacted, their secondary structure type, and their accessible surface area. Different scoring systems were used for each combination of annotations (hereafter termed an 'environment') and gap penalties were also varied with environment as, for example, a gap is unlikely to be opened in the middle of a transmembrane helix.

MP-T has two separate alignment stages: a pairwise alignment stage that is similar to the 'membrane FUGUE' of the previous chapter, and a multiple alignment stage that makes no use of environment information. These are now described in turn.

As in the previous chapter, the pairwise alignment stage relies on annotations of membrane positioning by iMembrane, and secondary structure and accessibility by JOY. Environment specific substitution tables are built in a manner similar to that described in Sections 3.2.2 and 3.2.3. The main changes are that the alignments used in table construction are made with MAFFT L-INS-i [Katoh and Toh, 2008] instead of MUSCLE, and that substitutions for an environment are now counted symmetrically e.g. a substitution from alanine in a helix environment to glycine in a sequence with unknown environment $A_{helix} \rightarrow G_{sequence}$, implies a substitution $G_{helix} \rightarrow A_{sequence}$, even though the glycine could potentially not be in a helix.

Environments are merged to improve table quality as in the previous chapter.[1]

Unlike 'membrane FUGUE' , MP-T has gap penalties that vary depending on the membrane environment. Gap penalty values were optimised on the 72 pairwise alignments described in Section 3.3.5 of the previous chapter, by testing perturbations around plausible values. The resulting penalties, listed in Table 4.1, are most severe in transmembrane regions and in secondary structure elements. This agrees with the high conservation of secondary structure elements, and with the fact that transmembrane elements are constrained to be at least the width of the membrane. A terminal gap penalty of 50 per unaligned position is also used.

Table 4.1: Gap penalties for MP-T. Secondary structure means that a residue is annotated as being part of an α-helix or β-strand by JOY. Residues are classed as transmembrane if they are annotated as being in the tail layer by iMembrane.

Secondary structure?	✓	✓	×	×
Transmembrane?	✓	×	✓	×
Gap open	360	300	220	200
Gap extension	60	20	20	10

This pairwise method cannot be translated directly into a multiple sequence alignment method as only the template sequence has structural annotation. To remedy this, MP-T makes pairwise alignments between the template sequence and each homolog, and transfers structural annotation between aligned pairs of residues (step 3, Figure 4.1).

Alignments are then made between all pairs of sequences, and distances are calculated between the sequences based on their sequence identity. The distance measure is Grishin's estimate of the mean number of substitutions per position d for sequences sharing sequence identity q, which assumes that the substitution rate varies between sites and among amino acid types [Grishin, 1995]:

$$q = \frac{\ln(1 + 2d)}{2d} \tag{4.1}$$

[1]The substitution tables used here are available at http://www.stats.ox.ac.uk/proteins/resources

These distances are used to construct a guide tree, and to select homologs for the multiple alignment phase: only sequences judged by the guide tree to be descendants of the most recent common ancestor of the target and template are selected (step 4, Figure 4.1).

Multiple alignment then proceeds using our implementation of the T-Coffee objective criterion [Notredame *et al.*, 2000] (Section 1.7.2). The criterion requires a library of pairwise alignments, which is built up as the distances between sequences are calculated. During multiple sequence alignment no gap penalties are imposed, other than a terminal gap penalty of 5 per position.

The above-outlined approach merges our accurate pairwise aligner into a consistency-based membrane multiple sequence aligner.

4.2 Methods

4.2.1 Construction of training and test sets

Sequences were extracted from proteins with PDB codes in the Membrane Proteins of Known 3D Structures Database on 10th January 2012 [White, 2013]. The sequences were ordered as follows:

1. Sequences that formed the gap-penalty training set
2. Sequences present in the CGDB molecular dynamics database
3. Sequences from X-ray structures (sorted by resolution)
4. Sequences from NMR structures

As the list of sequences was traversed in the above order, a sequence was 'accepted' if it had $< 60\%$ sequence identity (as determined by a TM-align structural alignment [Zhang and Skolnick, 2005]) to all previously accepted sequences.

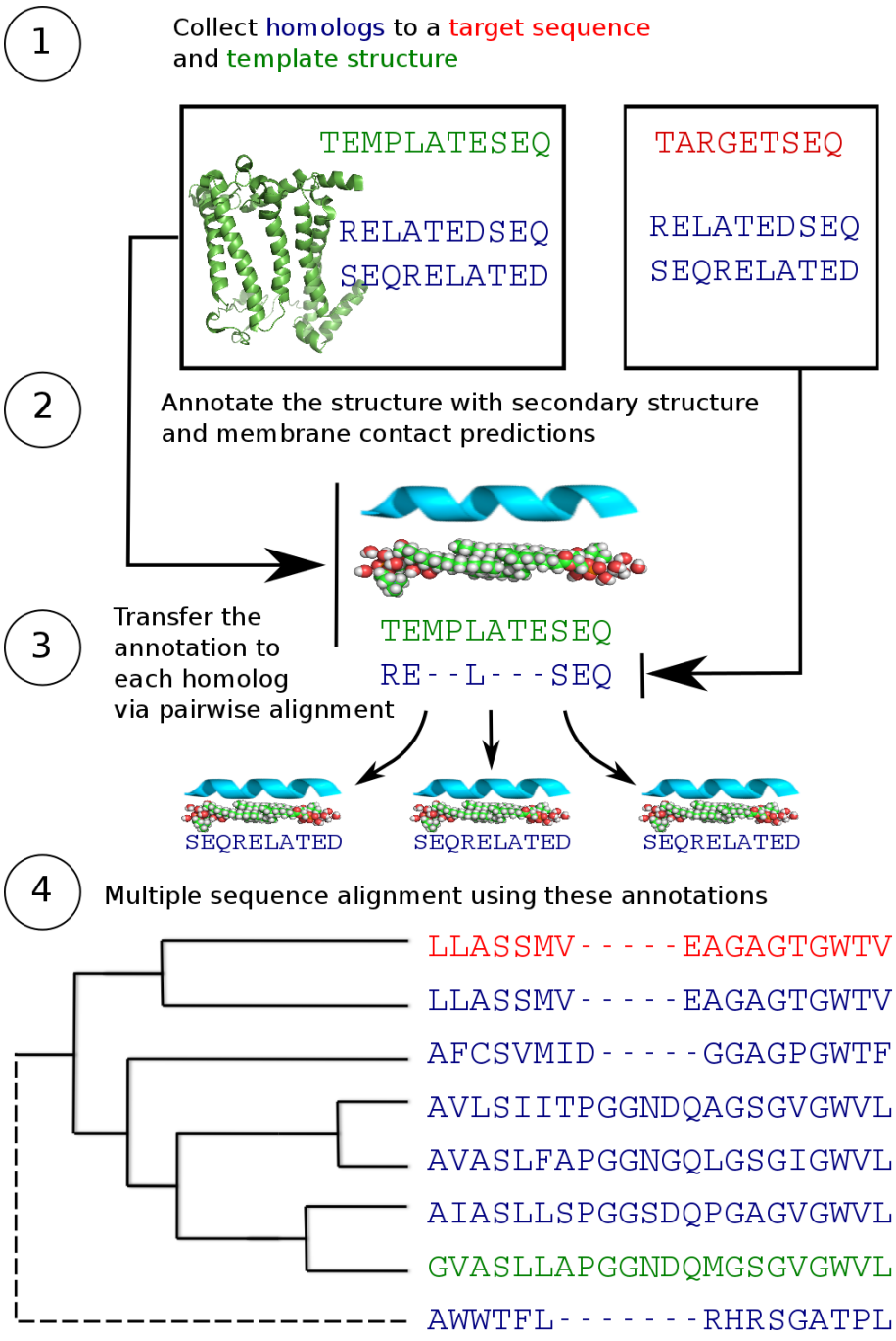


Figure 4.1: Schematic of the MP-T algorithm. The dashed line in the guide tree of step 4 shows a homolog that is excluded from multiple sequence alignment because it is more distantly related to the target and template than they are to each other.

Pairwise structure alignments were made between all pairings of accepted sequences. These alignments were then filtered such that no alignments had RMSDs of ≥ 3.5 Å, had TM-scores of < 0.5, contained a sequence with length < 60 residues, contained a co-crystallised soluble protein fragment, or were aligned over $< 60\%$ of the length of the shorter sequence. Alignments involving sequences without an iMembrane annotation were discarded. This procedure prevents bias in the selected sequences, and ensures alignments are between proteins with sufficient structural similarity for use in modelling.

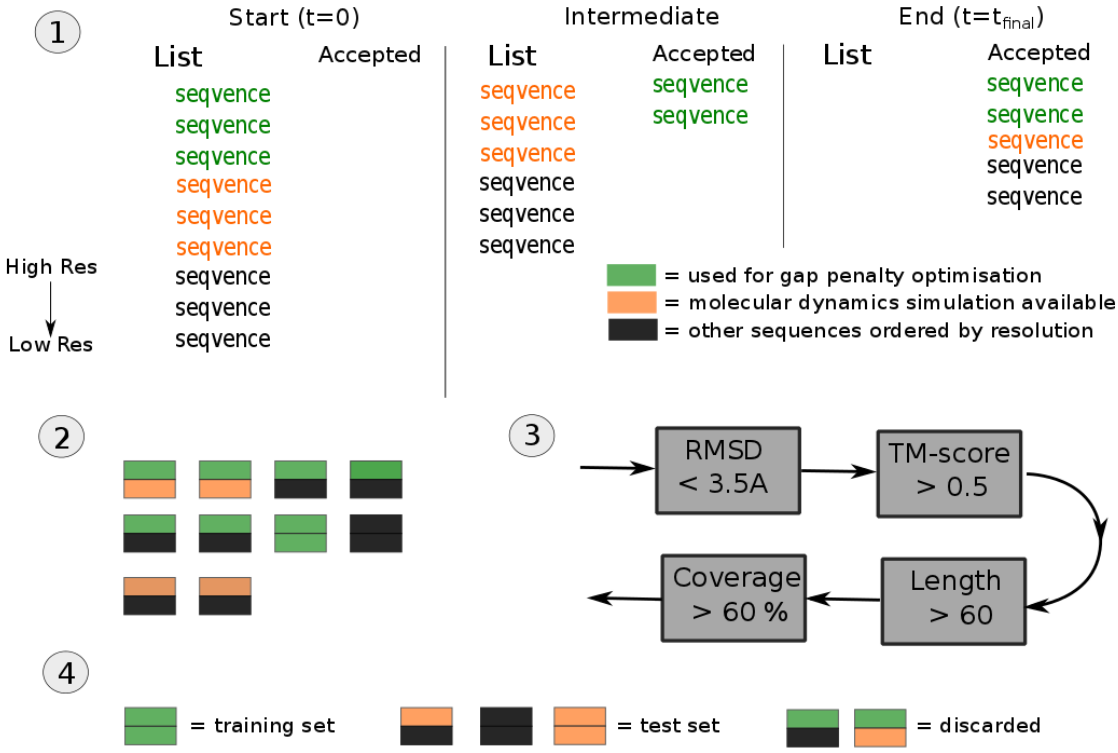


Figure 4.2: Schematic of test set and training set construction. 1) Candidate sequences from transmembrane proteins are ordered according to whether they were used to optimise gap penalties, whether they have been subjected to molecular dynamics simulations, and the resolutions of their PDB structures. This list is descended and sequences are 'accepted' if they are $< 60\%$ identical to all previously accepted sequences. 2) All possible structure alignments are made between sequences in the accepted list. 3) These alignments are filtered according to criteria of structure similarity and sequence length. 4) The surviving alignments are partitioned into a training set, a test set, and a set that is discarded.

The alignments were then partitioned into a training set and a test set. The training set was composed of 73 alignments that satisfied these requirements and where both sequences were taken from the first category of the above list. The test set consisted of 165 alignments where neither sequence was part of the first category. A schematic of this procedure is provided in Figure 4.2. Properties of the training and test sets are plotted in Figure 4.3.[1]

The alignments in the test set are diverse – spanning an order of magnitude in length and sequence identity (Figure 4.3). Secondary structure content varies from $> 75\%$ α-helix to $> 75\%$ β-strand. Sequences forming soluble parts of transmembrane complexes are also represented, with 50 of the 165 alignments having sequences without a transmembrane domain under our definition (Section 4.2.4).

4.2.2 Alignment input

Alignments were made with the profile-profile alignment programs PROMALS [Pei and Grishin, 2007] and HHsearch v2.0.5 [Söding, 2005], and with the multiple sequence alignment programs MP-T (this work), MUSCLE v3.8.31 [Edgar, 2004], MSAProbs v0.9.5 [Liu *et al.*, 2010], MAFFT v6.864*b*, T-Coffee v9.01 [Notredame *et al.*, 2000], and clustalΩ v1.0.3 [Sievers *et al.*, 2011]. The multiple sequence alignment programs are summarised in Table 4.2. We allowed both profile-profile programs to select their own homologs (see next section), but tested all multiple sequence alignment programs on identical input sequences which were collected as described below.

Homologous sequences to each chain were extracted from the UniRef90 database (accessed 6$^{\text{th}}$ February 2012) [Suzek *et al.*, 2007] by running PSI-BLAST [Altschul *et al.*, 1997] for 5 iterations with e-value thresholds of 10^{-3} to keep a hit, and 10^{-5} to incorporate a hit into the search profile. Homologs with $< 15\%$ sequence identity to the query over the aligned region were discarded, as were those with lengths $< 2/3$ or $> 3/2$ that of the query. The remaining homologs were randomly ordered and then made non-redundant by UCLUST [Edgar, 2010] at 80% sequence identity.

[1] All alignments can be found at http://www.stats.ox.ac.uk/proteins/resources

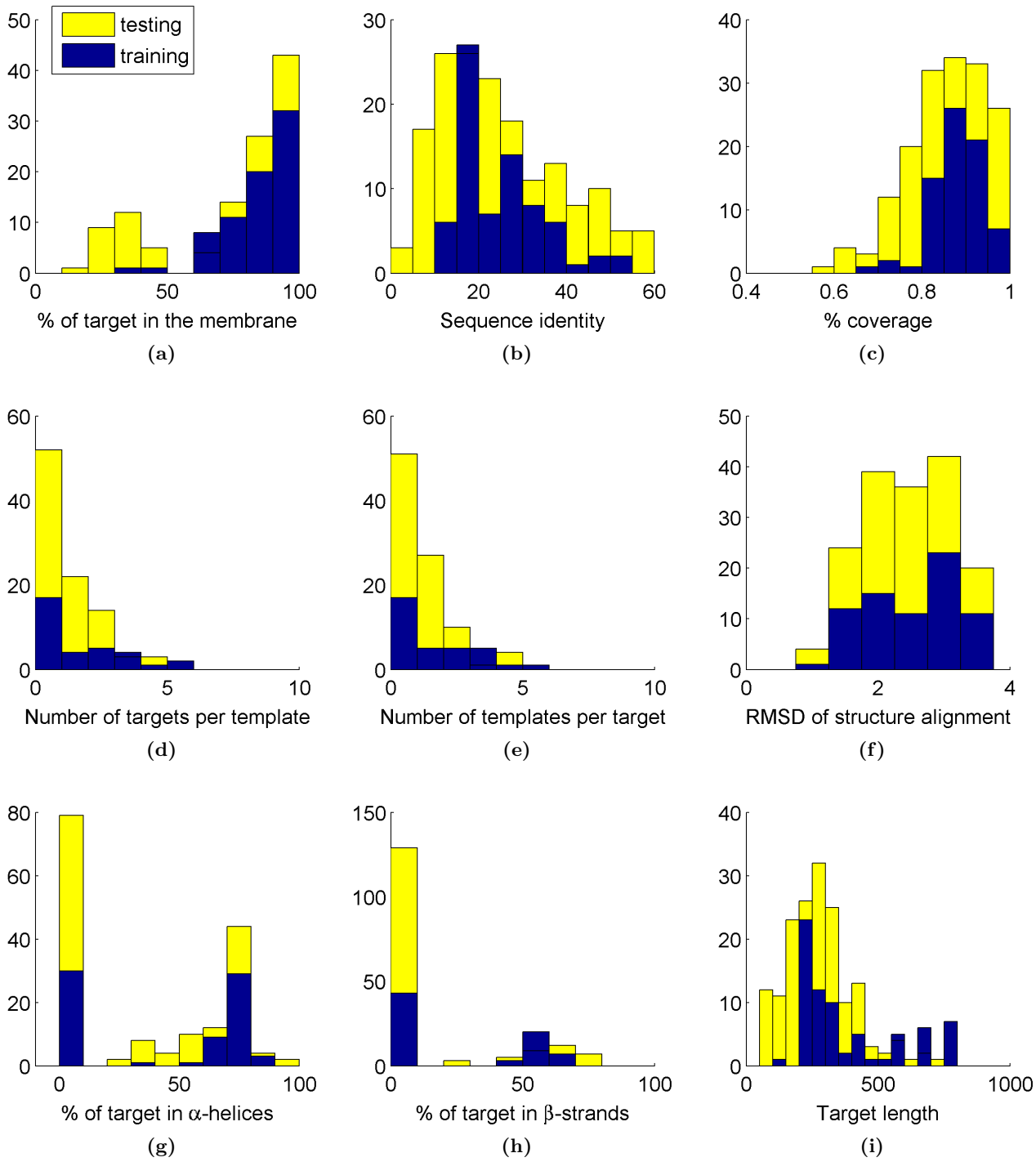


Figure 4.3: Comparison of various properties of the test set (yellow) with those of the training set (blue). The principal differences are that the test set contains alignments without a transmembrane domain or with a transmembrane domain that makes up a smaller fraction of the target sequence (a), and contains more short proteins (i).

For each alignment, the homologs from the template and target were combined in equal numbers (alternating the order to first take a homolog from the template, then one from the target, with any surplus being discarded), filtered to have lengths $> 2/3$ and $< 3/2$ that of the template, and again made redundant at 80% using UCLUST. The first 125 sequences were used as input to each aligner. This is similar to the way in which the general purpose PREFAB benchmark is constructed, but with a greater number of selected sequences (PREFAB alignments have at most 50 sequences) and different cut-offs [Edgar, 2004].

This protocol ensures that the multiple sequence alignment contains a reasonable number of sequences with little bias towards either the target or template, a wide range of sequence identity, and sequences of approximately equal length.

4.2.3 Optimisation of alignment programs

We attempted to find optimal settings for each alignment method based on its performance on our training set.

The default settings of MUSCLE and MSAProbs were found to perform well on the training set and so no changes were made. We tested the L-INS-i and G-INS-i modes of MAFFT and used the former as it had the higher accuracy. We used the PSI-Coffee mode of T-Coffee (with the UniRef90 database for sequence search), as we found it to be more accurate than the default T-Coffee settings, whilst providing results directly comparable with those of TM-Coffee. By default, ClustalΩ constructs trees using a variant of the fast mBed algorithm and does not perform iteration. We found two rounds of iteration and non-mBed tree construction improved performance (options: –full –full-iter –iter 2) and so ran these settings.

Although PROMALS [Pei and Grishin, 2007] is capable of performing a multiple profile alignment, we found that a pairwise profile-profile alignment was more accurate on our training set. PROMALS comes with its own sequence database (UniRef90 dated Feb 2007) against which it finds homologs using its own copy of BLAST.

HHsearch [Söding, 2005] is part of HHsuite v. 2.0.5. The suite contains the HHblits search

Table 4.2: Summary of characteristics of multiple sequence alignment methods used in this Chapter. PROMALS is included for completeness, but it is not used as a multiple aligner here.

Method	Year	Aligned unit	Iteration	Consistency
MP-T	2013	sequence	×	Heuristic
MSAProbs	2010	sequence	✓	Probabilistic
MAFFT L-INS-i	2008	sequence	✓	Heuristic
PROMALS	2007	profile	×	Probabilistic
PSI-Coffee	2009	profile	×	Heuristic
MUSCLE	2004	sequence	✓	×
clustalΩ	2011	sequence	✓	×

tool [Remmert *et al.*, 2012] which we used to generate HMMs for the target and template. Searches were conducted against the nr20 database (version dated 11$^{\text{th}}$ January 2010). Both HMMs were annotated by PSIPRED v. 2.5 [Jones, 1999a], and the template was further annotated with DSSP secondary structure states by JOY [Mizuguchi *et al.*, 1998b]. Best performance on the training set was obtained by combining a local initial Viterbi algorithm with a global MAC realignment.

We note that HHsearch and PROMALS enjoy the advantages of secondary structure annotation and homolog selection aimed at optimizing their performance. Our method, MP-T, also has these advantages. PSI-Coffee is a homology extension method and so uses more sequences in each alignment than are available to other methods. MSAProbs, MAFFT L-INS-i, MUSCLE, and ClustalΩ are directly comparable to each other.

4.2.4 Assessment of alignments and models

Alignment accuracy was assessed with reference to a TM-align structure alignment between the target and template. Two separate assessments were made: one over the transmembrane (TM) domain only, and the other over the entire sequence. Transmembrane domains were defined, using the 'membrane layer' annotation from iMembrane, to be regions that contain fewer than 30 consecutive not-in-membrane layer residues, more than 15 tail layer residues, and are at

least 40 residues long. Putative domains were trimmed to begin and end with at most 15 consecutive not-in-membrane layer residues. In the test set 115 of the 165 alignments contained TM domains: most of the other alignments formed soluble parts of TM complexes.

Alignment accuracy is commonly assessed by two related measures: the modeller score, F_M, and the developer score, F_D . These scores are different normalizations of the number of correctly aligned residues in an alignment. In F_M this number is divided by the number of aligned residues in the alignment being assessed; in F_D it is divided by the number of aligned residues in the reference alignment [Sauder *et al.*, 2000]. An alignment with high F_M includes few mistakenly aligned residues, but may not reproduce much of the reference alignment; one with high F_D reproduces a lot of the reference alignment but may misalign residues.

A third measure of alignment accuracy is the number of alignments where one method aligns 10 residues more correctly than another. Compared with F_M and F_D this measure avoids the problems of normalization and is robust against anomalously good or bad alignments.

Models of transmembrane domains were built with MEDELLER [Kelm *et al.*, 2010] and assessed using the GDT_TS measure calculated by the TM-score program over aligned pairs of residues [Zhang and Skolnick, 2004]. GDT_TS is the average fraction of residues in the model that are within 1, 2, 4, and 8 Å of their position in the experimental structure.

4.3 Results

4.3.1 Homolog selection

It is unclear how homologs should be chosen to help create a good sequence alignment between two distantly related proteins. In this section we discuss variations on the procedure used throughout the rest of this work, which is described in Section 4.2.2.

MSA accuracy is thought to deteriorate as the number of aligned sequences increases above ~ 100 [Thompson *et al.*, 2011] but 1000s of homologs can be returned in a BLAST search. We tested MP-T on our training set with $25 - 150$ homologs in steps of 25, and found that the

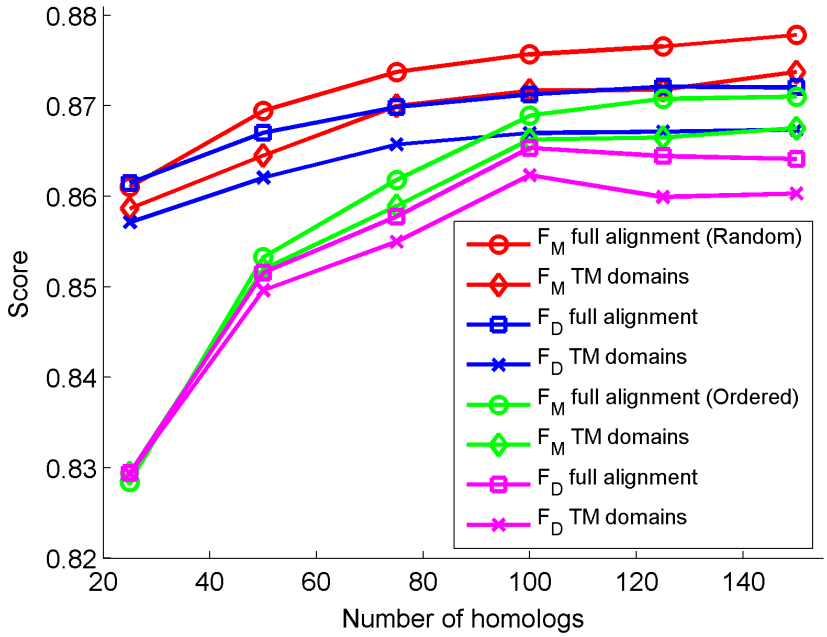


Figure 4.4: Alignment accuracy on our training set increases as the number of homologs is increased. Selecting homologs to be as similar as possible to the target and template (green and purple lines) is a less effective strategy than randomly selecting homologs (red and blue lines), showing that diverse sequence information is required for accurate alignment.

accuracy tended to a fixed upper limit in this range, with little improvement from $75 - 150$ homologs (Figure 4.4).

A feature of our homolog selection procedure is randomization. One alternative approach is to order homologs by their BLAST score – favouring more nearly related sequences. For MP-T, selecting homologs by BLAST score always led to a decrease in accuracy compared with selecting homologs randomly (e.g. green lines are beneath red lines in Figure 4.4). This suggests that homologs should be diverse and spread over the entire range of sequence identity.

4.3.2 Tree-building

Progressive multiple alignment requires a 'guide tree' to determine the order of alignment. Counterintuitively, simple tree-building heuristics such as single-linkage clustering and UPGMA

have been found to lead to more accurate alignments than phylogenetically more precise methods such as neighbor-joining (NJ) [Plyusnin and Holm, 2012; Wheeler and Kececioglu, 2007].

In MP-T tree building determines not just the order in which sequences are aligned, but also which sequences are used to align the target and template. For example, a tree with the target and template on adjacent leaves would give a standard pairwise alignment, whereas a tree with the target and template meeting at the root would involve all other sequences (this is illustrated in step 4 of Figure 4.1, where the sequence linked by the dashed line is not used to align the green template and red target sequences). We tested the tree-building procedures NJ, BIONJ [Gascuel, 1997], single-linkage clustering, and UPGMA on our training set to determine whether simpler algorithms still led to more accurate alignments in this case.

The algorithms differed in how many sequences they used to align the target and template. For BIONJ and NJ on average $\sim$ 70% of the possible number of homologs were used; for UPGMA and single-linkage clustering $\sim$ 90% of homologs were used. Despite this, changes in tree-building did not greatly affect accuracy. Single-linkage clustering appeared to be most accurate and NJ least accurate. We use UPGMA in the remainder of this work.

4.3.3 Alignment accuracy

MP-T alignments had a significantly higher average F_M (lower error rate, see Section 4.2.4) than all other methods, both over the transmembrane (TM) domain and the full alignment (Table 4.3). Little variation was seen between the top methods in F_D, the fraction of the reference structure alignment that was reproduced, but methods that made use of secondary structure annotation (MP-T, PROMALS, and HHsearch) achieved higher scores over the TM domain. In addition to the methods discussed in Section 4.2.2, Table 4.3 contains results for a version of MP-T that uses no environment information (*MP-T (1 table)*), and a version that uses no homologous sequences (*MP-T (pair)*).

The small variation in F_D over full alignments allows the intrinsic error rates of the different methods to be compared. The low error rates of MP-T appear to be a consequence of the T-

Table 4.3: Performance of different methods over the transmembrane (TM) domain and full alignment. MP-T has significantly higher values of F_M than all other methods. The top-scoring methods differ little in F_D.

	TM domain		Full alignment	
Method	F_M	F_D	F_M	F_D
MP-T	**66.3**	65.8	**69.6**	**70.1**
MP-T (1 table)	65.3	63.7**	69.2	68.5**
MP-T (pair)	59.1**	60.5**	61.5**	63.8**
MSAProbs	62.5*	62.1*	68.2*	69.0
MAFFT L-INS-i	64.1**	63.8**	68.4*	69.1
HHsearch	64.1**	65.3	67.5**	69.6
PROMALS	64.4**	**66.0**	67.4**	69.9
PSI-Coffee	65.4*	64.6*	68.7*	69.1
MUSCLE	60.8**	61.3**	65.1**	66.6**
clustalΩ	63.9**†	63.8*	67.9**†	68.5**

Entries marked with asterisks are significantly different from MP-T ($p < 0.1$ are marked *, $p < 0.01$ are marked **). Significance is assessed by a Wilcoxon signed-rank test.
† In 2 cases clustalΩ does not align any residues in the target with those in the template. We set F_M for these alignments to 0.

Coffee objective criterion – use of a single environment-independent substitution table in MP-T drastically reduced performance but still led to fewer incorrect pairings than for other methods (*MP-T (1 table)*, Table 4.3). After MUSCLE, the two profile-profile methods PROMALS and HHsearch incorrectly aligned the most residues. We find that in general multiple sequence alignment programs incorrectly align a smaller proportion of residues between the target and template than profile-profile methods. This is probably because in a MSA there are more ways of incorrectly pairing residues without matching a residue in the target with one in the template.

For 15 alignments no method aligned more than 25% of the residues correctly. These 'un-alignable' proteins typically had low sequence identity (average 7%) and were either short, or had the majority of their residues in β-strands (8 were shorter than 150 residues; 5 were majority β-strand). The short proteins paired here may not be homologous, but merely share a large secondary structure feature. The over-representation of β-strands may highlight difficulties in

deriving a reference alignment from a β-barrel structure alignment. For example, models that are built based on an alignment that differs significantly from the reference should be poor, but in the case of the 8-stranded β-barrels 3QRA and 3DZM, a model was built with a GDT_TS of 67.5%, despite the MP-T alignment including only 26% of the residue pairings in the reference alignment.

Table 4.4 shows the number of alignments for which each column method correctly aligned at least 10 more residues than the corresponding row method. The first entry in the table is assessed over the 115 transmembrane domains, and the entry in parentheses is assessed over the whole target sequence (165 alignments). MP-T consistently outperformed all other methods more often than they outperformed it, with the advantage being largest over transmembrane regions. The MP-T (pair) column performed a pairwise alignment; its poor performance illustrates the importance of homologs in creating an accurate alignment.

As PROMALS came nearest to parity with MP-T (Table 4.4), we selected it to investigate the types of alignments each performed best. Figure 4.5 shows all alignments where PROMALS and MP-T outperformed each other by at least 10 residues over the full alignment. Several cases where MP-T performed less well (rectangles to the left of the black line), may be attributable to the use of a small number of homologs to make the alignment (thin rectangles in the figure). This may have arisen either because the procedure in Section 4.2.2 returned few homologs, or because the guide tree excluded many homologs (see Section 4.1.2).

MP-T appeared to perform better on longer proteins (longer bars to the right of the black line) which is to be expected: differences between a single substitution table and environment specific tables accumulate with sequence length. However, the improved performance may also reflect the composition of the training set, which contained only one alignment with a target sequence shorter than 200 residues (Figure 4.3i). PROMALS may have performed better at making very distant alignments $< 15\%$ sequence identity, whereas MP-T had an advantage in the $15 - 30\%$ identity range, which is arguably more useful for modelling.

Equivalent plots for the other methods are shown in Figure 4.6. The four methods using

Table 4.4: Number of times column method beats row method by at least 10 residues. The first entry is assessed over the transmembrane region only, whereas parenthetical entries are assessed over the whole target sequence. For example there are 20 alignments in which HHsearch beats clustalΩ in the transmembrane region, and 26 where HHsearch beats clustalΩ overall.

	MP-T	MSAProbs	MAFFT	HHsearch	PROMALS	PSI-Coffee	MUSCLE	clustalΩ	MP-T (pair)
MP-T	-	6 (12)	7 (13)	17 (21)	17 (25)	13 (18)	8 (9)	12 (15)	9 (8)
MSAProbs	**20 (24)**	-	**14** (16)	17 (20)	**25 (30)**	**15 (17)**	10 (10)	12 (12)	13 (14)
MAFFT	**22 (25)**	13 **(17)**	-	19 (23)	**19** (23)	**16 (21)**	8 (9)	**17** (18)	14 (13)
HHsearch	**28 (38)**	**20 (33)**	**20 (35)**	-	**25 (34)**	**25 (34)**	18 (25)	20 **(30)**	14 (19)
PROMALS	**21 (31)**	11 (28)	15 **(30)**	16 (27)	-	11 (22)	10 (20)	16 (27)	12 (13)
PSI-Coffee	**20 (25)**	8 (13)	12 (19)	16 (21)	**20 (25)**	-	5 (6)	15 (21)	9 (12)
MUSCLE	**37 (52)**	**30 (50)**	**21 (37)**	**28 (43)**	**32 (43)**	**28 (41)**	-	**23 (34)**	20 (25)
clustalΩ	**20 (27)**	**14 (24)**	13 **(24)**	20 (26)	**21 (29)**	15 **(24)**	5 (7)	-	14 (20)
MP-T (pair)	**40 (70)**	**39 (73)**	**41 (74)**	**34 (68)**	**38 (68)**	**36 (65)**	**32 (60)**	**38 (72)**	-

An entry is in bold if the column method beats the row method more often than the row method beats the column method. Better performing methods have more bold entries in their column. Details of how each program was run are provided in Section 4.2.3.

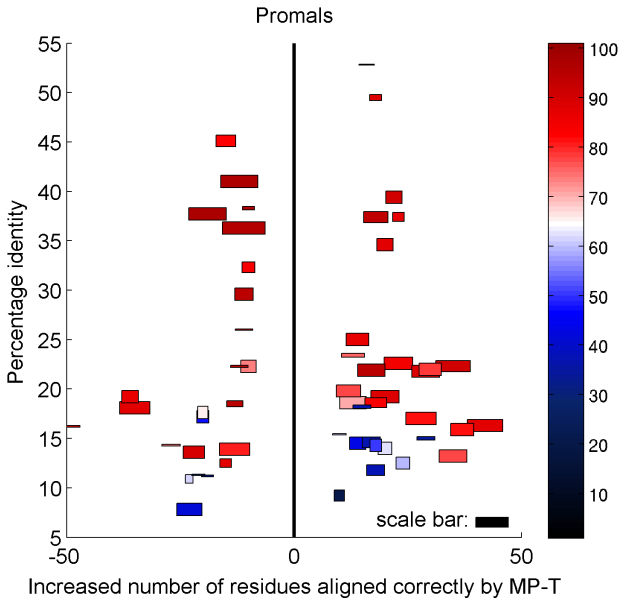


Figure 4.5: Alignments for which PROMALS or MP-T aligned at least 10 residues more correctly than the other. The assessment is made over the full target sequence. The size of each rectangle is a proxy for the size of the multiple sequence alignment for each input: rectangle height is proportional to the number of sequences in the MP-T alignment, rectangle length is proportional to the length of the target sequence. The scale bar shows a rectangle corresponding to an alignment made using 100 sequences and with a target length of 500 residues. Rectangles are coloured by the fraction of the structure alignment that the better performing method reproduced. One point at $(-72, 9)$ falls outside the axes of the graph: PROMALS aligned 63% of the residues correctly.

the same input sequences as MP-T – ClustalΩ, MAFFT, MSAProbs, and MUSCLE – tended to perform worse than MP-T when there were very few homologs (thin rectangles to the right of the black lines), as these cases were essentially a test of pairwise alignment accuracy as in Chapter 3. MUSCLE was clearly the least accurate alignment scheme.

Alignment methods showed a greater range of accuracy over majority β-strand targets than over majority α-helical targets. On average β-strand targets are shorter than other targets, and have lower sequence identity reference alignments that contain a higher density of gap open events. These factors may have differentiated methods by their gap-penalty schemes.

The above results were obtained when only the template sequence was structurally annotated. However, MP-T is capable of accepting more annotations, derived either from experi-

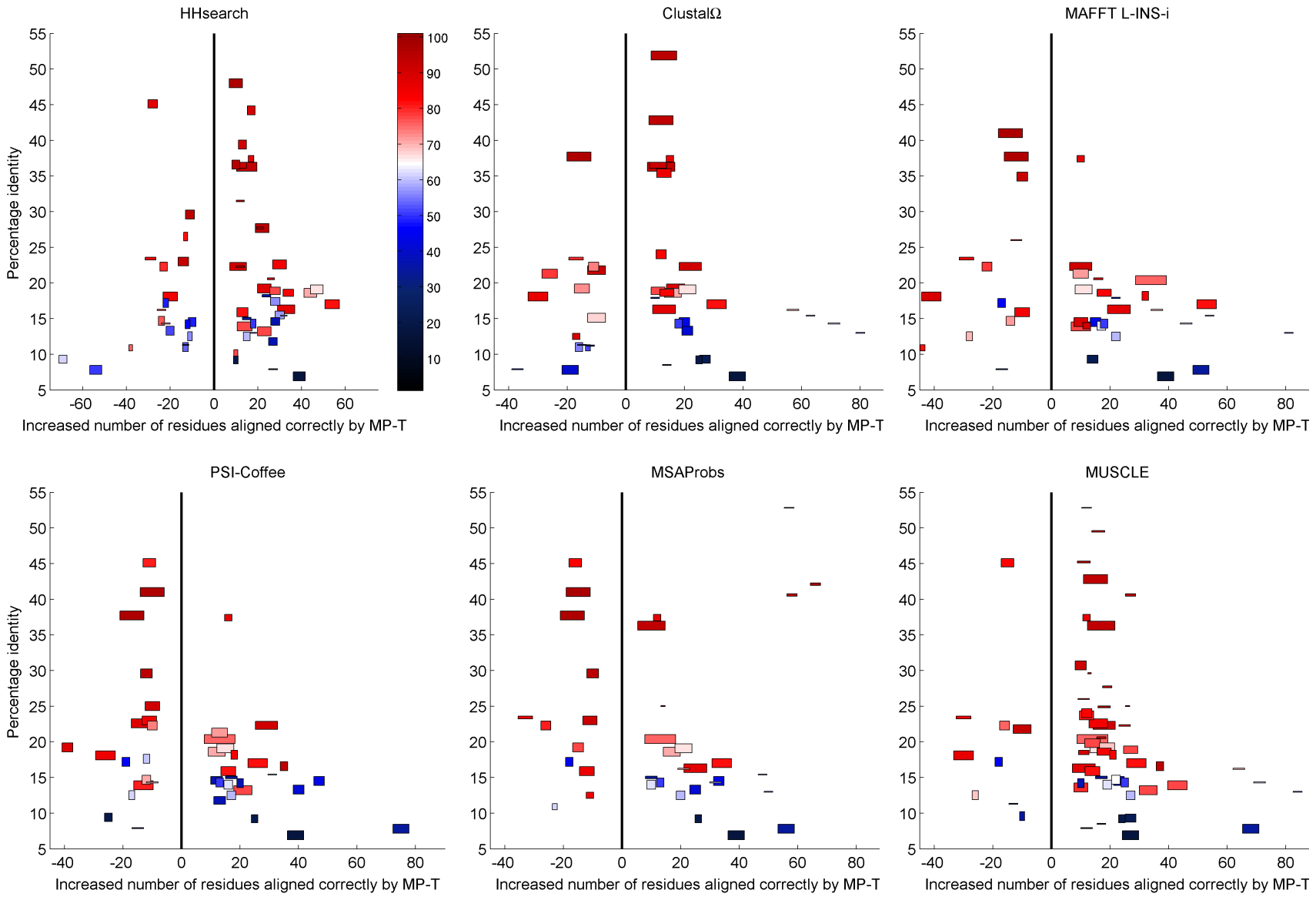


Figure 4.6: Alignments for which the named method or MP-T aligned at least 10 residues more correctly than the other over the full alignment. See Figure 4.5 for a description of the plot style. The centre and right columns show alignment methods that used the same sequences as MP-T. The plots for PSI-Coffee and MUSCLE each omit a point at high sequence identity to the right of the black line. The MUSCLE plot additionally omits a point to the right of the axes. The HHsearch plot omits a point at $(-10, 4)$. Note that the x-axis for the HHsearch plot differs from that of the others.

mental structures or from prediction programs such as PSIPRED. MP-T's internal transfer of structural annotation between sequences is an implicit prediction of secondary structure, solvent accessibility, and membrane-positioning. Additional annotations should lead to improved alignments if they have a lower error rate than these implicit predictions. An approximate bound to the accuracy gains from adding more annotations was found by providing structural annotations for the target sequence as well as for the template structure. This increased F_M and F_D by $\sim 0.7\%$ over both TM domains and full alignments.

4.3.4 Model accuracy in the transmembrane region

We built models for the 115 transmembrane domains in the test set – which in most cases, such as ion channels, is the domain of interest. Models were built using MP-T, PROMALS, and MUSCLE alignments with the high-accuracy mode of MEDELLER [Kelm *et al.*, 2010] which incorporates the FREAD loop-modelling method [Choi and Deane, 2010]. The loops for most targets were present in FREAD's database, meaning the models built here should be better than those for blind structure prediction.

PROMALS and MUSCLE were chosen for model building alongside MP-T as they respectively had the best and worst alignment accuracy of the methods against which we compared. Model accuracy was assessed by calculating the model GDT_TS. This is a number in the range $(0, 1]$ with higher values corresponding to more accurate models. To isolate the contribution of the alignment method to the model, GDT_TS was calculated only over pairs of residues that were aligned in the input sequence alignment. Models were built using the full sequence alignment, as in the MEDELLER procedure this cannot worsen the transmembrane model, but can improve loop-modelling.

MP-T produced models of significantly higher GDT_TS than PROMALS or MUSCLE ($p < 10^{-4}$, Wilcoxon signed-rank test). The distribution of improvements in model GDT_TS gained by using MP-T rather than PROMALS is shown in Figure 4.7. For example, for 30 of the 115 models, MP-T alignments led to a $2.5 - 7.5\%$ improvement in model GDT_TS compared to

models made with PROMALS. The small height of bars to the left of the origin shows that MP-T rarely generated significantly worse models than PROMALS. The lower panel of Figure 4.7 shows that PROMALS alignments led to models with higher coverage than MP-T. On average, MUSCLE's models had the lowest GDT_TS and the lowest coverage, demonstrating that less accurate alignments lead to less accurate models.

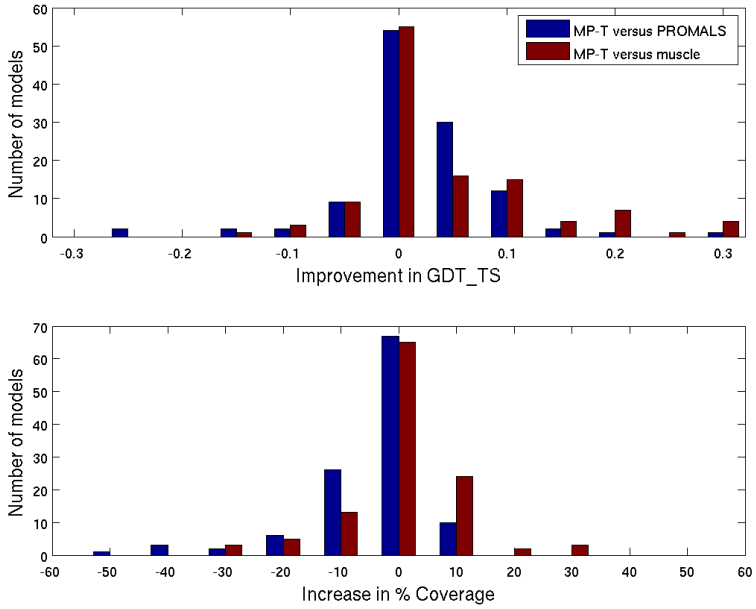


Figure 4.7: Distribution of improvements in model accuracy (top) and model coverage (bottom) from using MP-T rather than PROMALS or MUSCLE. Models produced from MP-T alignments provide lower coverage than PROMALS (bars are higher to the left of the origin in the lower panel) but are significantly more accurate (bars are higher to the right of the origin in the upper panel, $p < 10^{-4}$ by Wilcoxon signed-rank test).

Figure 4.8 illustrates improved model-building at low sequence identity. The target (PDB code: 1IZL, chain D gray) and template (PDB code: 1OGV, chain M) were taken from photosynthetic reaction centres. Models from MP-T (left) and PROMALS (right) are colour-coded per residue by the distance to the corresponding residue in the native structure (0Å= blue, $\geq$ 5Å= red). Numbering transmembrane helices $1-5$ from left to right, MP-T modelled more of the loop region above helix 1, and more of helix 4 than PROMALS. MP-T assigned the correct residues to helix 1, whereas the PROMALS model introduced a large shift (in the red

helix each residue is at least 5Å from its true position).

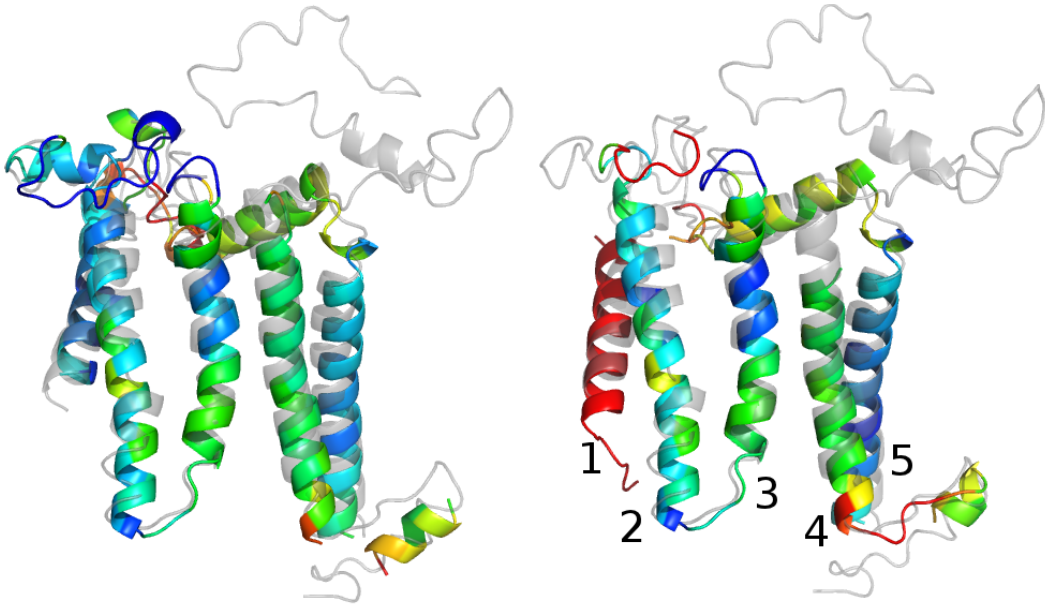


Figure 4.8: Part of a photosynthetic reaction centre, PDB code: 1IZL, chain D (gray), modelled using PDB code: 1OGV, chain M as a template. The target and template share 15% sequence identity. Residues in the model are coloured by the distance to their position in the crystal structure: blue residues are close to their native position, red residues are $\geq 5\AA$ away from their native position. The model on the left is built using MP-T whereas that on the right is built using PROMALS. The MP-T model provides a more accurate representation of helix 1, and models the region above this helix.

4.4 Memoir: a full membrane protein modelling pipeline

The material in this section is described more fully in the following published paper:

Ebejer, J-P, Hill, J. R., Kelm, S., Shi, J., and Deane, C.M. (2013).

Memoir: Template-based Structure Prediction for Membrane Proteins.

Nucleic Acids Research 41 (Web Server issue), W379 - 83

Many bioinformatics algorithms remain inaccessible to the wider scientific community because they require the user to be familiar with the command line, or to have administrator

privileges on their workstation. Even assuming that a user can run a program, the combination of a large number of features and small amount of help can lead to ineffective use of software. A good solution to these problems is to provide software in the form of a web server, where it can be run from any web browser. The web server format also ensures that users are always provided with the most up-to-date version of a program, and can be informed of developments through messages on the site's front page.

Sequence alignments from the MP-T algorithm are available from a stand-alone web server[1] and as part of the output of the Memoir web server[2]. Memoir is currently the only web server designed for the comparative modelling of general membrane proteins. It allows membrane protein models, such as that in Figure 4.8 to be produced from the input of a single protein sequence and the PDB code of a template protein (e.g. 1OGV, chain M). The Memoir website provides a video tutorial and written guide to inform users about the likely accuracy of the models they generate. Elements of an example results page are shown in Figure 4.9.

4.5 Discussion

We have created a program, MP-T, to perform target-template alignment for the comparative modelling of membrane proteins. Alignment accuracy is tested against 7 other alignment methods over 165 non-redundant alignments of membrane proteins. MP-T is found to introduce the fewest misaligned residues (δF_M from +0.9% to +5.5%) whilst aligning as accurately as the state-of-the-art methods HHsearch and PROMALS (Table 4.3). Alignments generated by MP-T also lead to significantly better models than those of the best alternative alignment tool (1/4 of models see an increase in GDT_TS of $\geq 4\%$). MP-T performs particularly well on longer membrane proteins (a category that includes drug targets such as GPCRs and ion channels), and in the twilight zone ($15 - 30\%$ sequence identity).

The accuracy of MP-T is derived from its effective use of information about accessible sur-

[1] http://opig.stats.ox.ac.uk/webapps/MPT/php/index.php
[2] http://opig.stats.ox.ac.uk/webapps/memoir/php/index.php

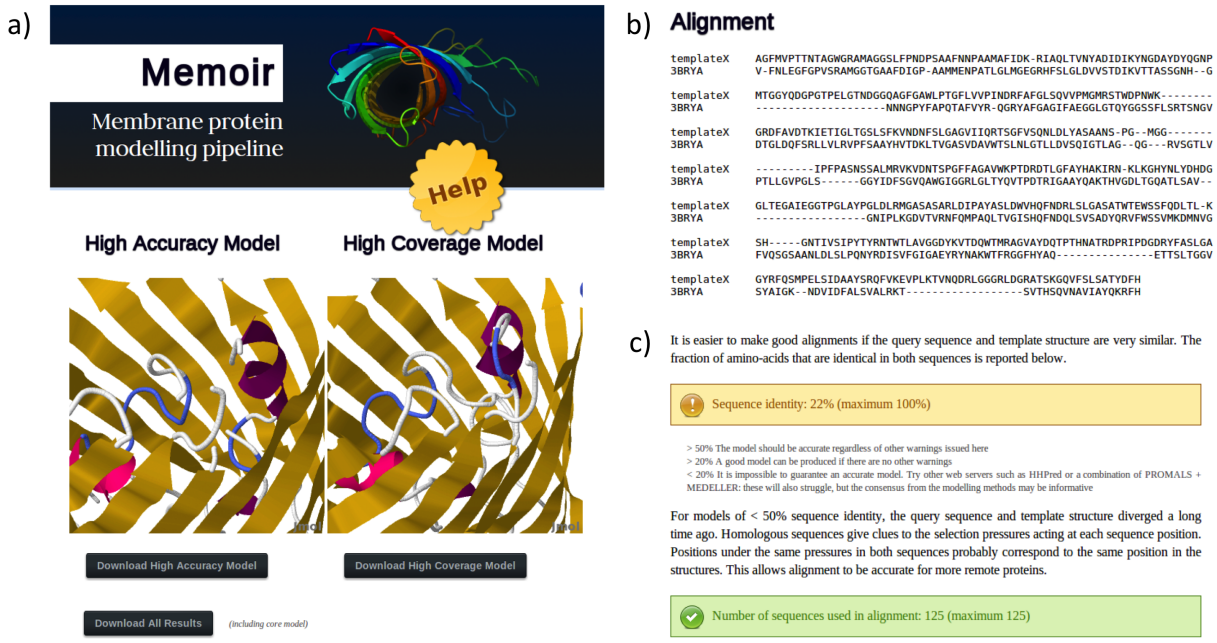


Figure 4.9: Parts of a Memoir results page **a** Two models are generated, one prioritising accuracy (the 'high accuracy' model) and the other completeness (the 'high coverage' model). They are displayed in the Jmol 3d graphics viewer and are available for download in PDB format. Additional information on model creation can be downloaded using the "Download all results" button **b** Also displayed is the alignment between the target and template structure that was used in model building **c** The alignment is accompanied by a guide to model quality, an extract of which is shown here. Values referenced in the guide, such as sequence identity, are calculated and displayed with traffic-light colour-coding (e.g. green for values that are likely to lead to a good model).

face area, membrane positioning, and secondary structure. This information is obtained from the template structure and used to make predictions about the same properties of homologous sequences. Gap penalties and substitution scores are adjusted on a per-residue basis depending on this information – for example the creation of gaps in the middle of the membrane is discouraged, and substitutions within a helix are scored according to the propensity for an amino acid type to be in a helix.

The choice of homologs is found to be the single greatest factor affecting alignment accuracy

for our method. Selecting homologs to favour sequences that are more closely related to the target and template leads to substantially less accurate alignments than random selection. It is likely that improvements in accuracy can be gained by further refining the selection process: for example, by loosening cut-offs if too few sequences are returned.

In agreement with previous studies, we find that guide tree construction does not strongly affect alignment accuracy. However, the general trend is that single-linkage clustering is a better method for building guide trees than either neighbor-joining, BIONJ or UPGMA.

We perform no sequence weighting, no iterative refinement, and use a simpler consistency criterion than those used in leading multiple-sequence aligners such as MSAProbs. Despite this, the use of environment-aware gap penalties and substitution tables allows us to produce more accurate models of transmembrane domains than other methods tested. Incorporation of environment-awareness into a more sophisticated aligner may yield even larger improvements in the quality of membrane protein models. MP-T is available as a web server, and also forms part of the Memoir web server, allowing researchers without specialist bioinformatics knowledge to generate MP-T alignments and membrane protein models.

Throughout this thesis we have so far assumed that a good template has been identified for use in alignment and modelling. In the next chapter we address the problem of identifying a good template within the twilight zone. The method we develop makes no use of membrane protein specific information, so may be applied to proteins more generally.

CHAPTER 5

The use of correlated substitutions for fold recognition

5.1 Introduction

The previous two chapters demonstrated how the local structural environment of a protein can be used to improve alignment for comparative modelling. Implicit in those chapters was the assumption that a template structure could be identified for a target sequence of interest, as without a template no model could be built. However, at present the existence of a suitable template is often only discovered after the target protein's structure has been experimentally

determined [Jaroszewski *et al.*, 2009].

Fold recognition (the identification of a template) becomes markedly more difficult when the sequence identity between the template and target falls within the twilight zone ($< 30\%$ identity). Recently, several methods have been developed that detect evolutionary constraints in a protein structure from sequence. These methods are independent of sequence identity, and so have great potential for fold recognition, a potential which we explore in this chapter.

5.1.1 Chapter overview

The recent development of 'direct information' algorithms has led to large improvements in the accuracy with which spatially close positions, 'contacts', in protein structures can be predicted [Ekeberg *et al.*, 2013; Jones *et al.*, 2012; Lapedes *et al.*, 1999; Morcos *et al.*, 2011]. These algorithms work by detecting correlated substitutions: pairs of positions in a protein structure that do not mutate independently of each other. This lack of independence suggests the positions may be undergoing compensatory substitutions, and so may be in close spatial proximity [Altschuh *et al.*, 1988; Göbel *et al.*, 1994; Korber, 1993; Neher, 1994; Shindyalov *et al.*, 1994; Taylor and Hatrick, 1994].

Several groups have performed *ab initio* protein modelling based on correlated substitutions. Greatest success has been achieved in the modelling of membrane proteins, partly due to the large size of membrane protein sequence families, and partly because constraints on membrane positioning can be used to filter out correlated substitutions that cannot correspond to structural contacts [Hopf *et al.*, 2012; Nugent and Jones, 2012]. Models have also been made for soluble proteins: Marks *et al.* [2011] achieved 7 top-ranked predictions with TM-score > 0.5 on a set of 14 soluble protein domains, and Sukowska *et al.* [2012] made models of 7 soluble protein domains with RMSDs to the native structure of $2.4-5.6$Å. Rather than directly building models, contact predictions have also been used to augment a method that selects the most accurate model from a set of 'ideal forms' [Sadowski *et al.*, 2011; Taylor *et al.*, 2012].

As with the above methods, we use correlated substitutions to infer the structure of protein

sequences. However, rather than making *ab initio* models, we identify templates for use in more accurate comparative modelling. To understand how correlated substitutions might be used to identify templates, it is useful to distinguish between 'spatial' and 'homology' information in correlated substitutions. For example, consider a homodimer with correlated substitutions between two positions on opposite sides of the dimer interface. As these two positions are not contacts in the monomer, they contain little spatial information with which to construct an *ab initio* model; however, a good template might possess a similar correlation between structurally-equivalent positions, meaning that the correlated substitutions contain homology information useful for comparative modelling.

Previous studies have been limited to investigating the spatial information in correlated substitutions. They have reached no clear consensus as to whether correlated substitutions are more common in a particular type of secondary structure or range of solvent accessibility [Chakrabarti and Panchenko, 2010; Choi *et al.*, 2005], although correlations are more common between residues on the same face of a helix (especially when separated by a single turn), and between adjacent residues on the same side of a β-strand [Little and Chen, 2009]. An excess of correlated positions are found at functionally important sites [Chakrabarti and Panchenko, 2010], and in regions where mutations are associated with disease [Kowarsch *et al.*, 2010].

In contrast to the above studies, we investigate the homology information available in correlated substitutions by reference to the SCOP protein structure hierarchy [Murzin *et al.*, 1995]. SCOP is a protein structure classification system with four levels – family, superfamily, fold and class. We use SCOP classifications as a proxy for relatedness, where proteins of the same family are most closely related to each other, and proteins of the same fold may not be related at all.

We find that correlated substitutions do contain homology information: closely related pairs of proteins share a greater number of highly-correlated substitutions than distantly related pairs. Moreover, we find cases where a correlated substitution does not correspond to a contact in a protein of interest, but does correspond to a contact in a related protein. In these cases the

average amount of correlation falls as the relationship between the two proteins becomes more distant. Correlated non-contacts such as these are undesirable for *ab initio* approaches, but convey useful information for template selection in comparative modelling.

By comparing correlated substitutions in one structure with equivalent positions in multiple related structures, we can also begin to understand what differentiates a contact undergoing correlated substitution from one that does not. We show that in closely related structures, contacts that undergo correlated substitution are more conserved than uncorrelated contacts, but in distantly related structures there is no difference between the two sets. This observation helps explain why the number of correlated substitutions is generally much lower than the number of contacts.

These insights into the homology information available in correlated substitutions are used to guide the development of a fold recognition algorithm, 'FORECAST' (FOld REcognition by Contact Adjusted STatistics), which identifies suitable templates for comparative modelling using only correlated substitutions and predicted secondary structure. On a test set of 645 Pfam families [Punta *et al.*, 2012], we demonstrate that FORECAST possesses a score threshold above which almost all hits make good templates. We then use FORECAST to assign SCOP fold annotations to 100 Pfam families for which our automated SCOP assignment proved impossible. For 73/100 cases, FORECAST's annotation is supported by conventional fold-recognition methods. In several of the remaining cases the FORECAST prediction can be structurally validated, and in two further cases the top hit of FORECAST shares a CATH topology [Sillitoe *et al.*, 2013] with the top hit of another method – demonstrating the detection of a distant structural link between SCOP folds from sequence alone.

5.1.2 Algorithms to detect correlated substitutions

The 'direct information' approach to detecting correlated substitutions is related to simpler 'mutual information' approaches. Here the two approaches are outlined and contrasted.

Mutual information ('MI') is one of the oldest approaches to detecting correlated substitu-

tions between two columns of a multiple-sequence alignment [Korber, 1993]. The MI between amino acid type i in column x and amino acid type j in column y is

$$MI(x,y) \quad = \quad \sum_{i,j}^{20} f(x_i, y_j) \log \frac{f(x_i, y_j)}{f(x_i) f(y_j)} \tag{5.1}$$

where $f(x_i)$ is the probability of observing amino acid i in column x. This is the Kullback-Leibler divergence between the joint distribution of amino acids in the columns and the estimate of the joint distribution assuming that the columns are independent.

Pairs of columns in a multiple sequence alignment that have a high MI contain a greater signal of correlated substitution than expected by chance. Nevertheless, MI is subject to biases arising from the non-independence of sequences in an alignment (phylogenetic bias), and the differing degrees of variability within alignment columns. For example, the maximum MI involving a column that contains n residue types is $\log n$ [Martin *et al.*, 2005].

Phylogenetic biases can be mitigated by refining estimates of $f(x_i)$ and $f(y_j)$ through sequence weighting or the adding of pseudocounts [Jeong and Kim, 2012] – techniques applied for the same purpose in sequence search programs such as PSI-BLAST (see Section 1.7.1). Several methods build on MI by attempting to subtract column-specific noise from the resulting predictions. For example, the MIp measure [Dunn *et al.*, 2008] applies an average product correction to MI

$$MIp(x,y) \quad = \quad MI(x,y) - \frac{\langle\ MI(x,a)\ \rangle_a\ \langle\ MI(a,y)\ \rangle_a}{\langle\ MI(a,b)\ \rangle_{a,b}}, \tag{5.2}$$

where $\langle\ f\ \rangle_x$ is the average of f taken over x. This correction accounts for the fact that the mean value of MI is column dependent. Others have improved on this measure by accounting for the column-specific variance of MI [Brown and Brown, 2010; Little and Chen, 2009].

Even if noise were completely removed, there would still be a need to distinguish causative

correlations from transitive correlations. Causative correlations originate from evolutionary constraints acting on a protein sequence, whereas transitive correlations are by-products of these causative correlations. For example, if columns A and B undergo correlated substitution, and columns B and C undergo correlated substitution, then on average a transitive correlation will be observed between columns A and C, even though this pair is not subject to evolutionary constraints.

Mutual information methods estimate whether a pair of residues is under evolutionary constraint by using only the sequence information in the corresponding two columns of a multiple sequence alignment (Equation 5.1). In contrast, direct information methods calculate correlated substitutions using all the columns in a multiple sequence alignment, and improve on MI by better distinguishing between causative and transitive correlations. At present a disadvantage of direct information methods is that they require a large number of sequences.

Two direct information methods are PSICOV [Jones *et al.*, 2012] and mfDCA [Morcos *et al.*, 2011]. Both methods define the covariance $C^{i,j}_{x,y}$ between amino acid type i in column x and amino acid type j in column y as:

$$C^{i,j}_{x,y} = f(x_i, y_j) - f(x_i)f(y_j). \tag{5.3}$$

The covariance matrix C for a multiple sequence alignment with m columns is of size $21m \times 21m$. The inverse matrix $\Theta = C^{-1}$ is of the same size, and each element is related to the partial correlation between the corresponding amino acids and columns. For example, the partial correlation between amino acid type i in column x and amino acid type j in column y is given by:

$$\rho^{i,j}_{x,y} = -\frac{\Theta^{i,j}_{x,y}}{\sqrt{\Theta^{i,i}_{x,x}\Theta^{j,j}_{y,y}}}. \tag{5.4}$$

A partial correlation is the correlation that remains when the correlations attributable to all other columns and pairs of amino acids have been discounted. PSICOV and mfDCA differ in how they estimate Θ, but until here are effectively the same algorithm. They now differ in how they relate the elements of Θ to a score.

In the case of PSICOV, the score $S(x, y)$ between two columns is simply the absolute sum of the 20×20 partial covariances between amino acids in those columns

$$S(x, y) \quad = \sum_{i,j=1}^{20} |\Theta_{x,y}^{i,j}|. \tag{5.5}$$

This is then adjusted using the average product correction (Equation 5.2) to attempt to correct for background effects such as phylogenetic bias.

In mfDCA the final score is a modified version of MI (equation 5.1) where the joint distribution $f(x_i, y_j)$ is replaced by an effective distribution $f^{\text{eff}}(x_i, y_j)$. This effective distribution depends on the partial correlations. It is based on a maximum entropy formulation, and is constrained to have the correct marginal distributions e.g. $\sum_{j=1}^{20} f^{\text{eff}}(x_i, y_j) = f(x_i)$.

5.1.3 The SCOP and Pfam classification systems

In this chapter, the SCOP classification of protein structures is used to provide a proxy for the relatedness of two proteins, and correlated substitutions are predicted for families taken from the Pfam classification of protein sequences. These are now described in turn.

The Structural Classification of Proteins (SCOP) database [Murzin *et al.*, 1995] is a hierarchical clustering of publicly-available protein structural domains. The hierarchy has 4 levels called 'class', 'fold', 'superfamily', and 'family'. Proteins in the same family tend to share $\geq 30\%$ sequence identity and can be assumed to be homologous. Proteins in the same superfamily, but different families, are likely the result of divergent evolution, yet can still usually be grouped by sequence-based methods. At the fold level proteins share an arrangement of secondary struc-

ture, and their evolutionary relationship is less clear. The SCOP database is manually-curated. Assigning proteins of different superfamilies to the same fold often requires human judgement. An example of the SCOP naming system is shown in Figure 5.1.

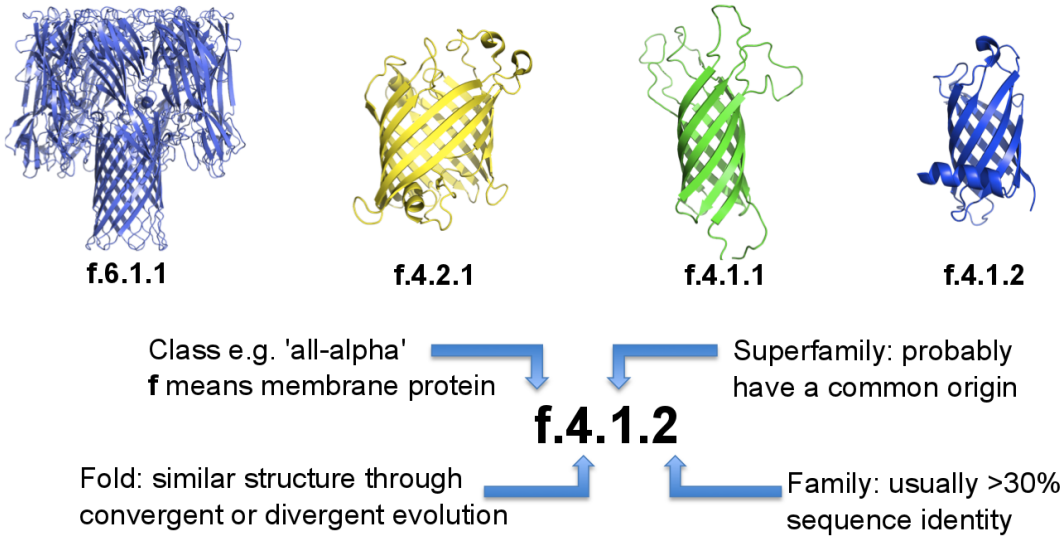


Figure 5.1: Example SCOP annotation

The Pfam database considers sequence instead of structure. In this database sequences are grouped using Hidden Markov Model (HMM)-sequence searches [Punta *et al.*, 2012] . There are two different Pfam sets which differ in how the HMM searches are seeded. Pfam A is the smaller set, and is considered more accurate. Pfam B is larger but less curated. Each Pfam entry includes a multiple sequence alignment and an HMM. The sequence-based nature of Pfam leads to groupings that do not correspond neatly to structural domains. The four groupings are listed below along with their approximate interpretation:

1. Domain: a sequence fragment found as part of many apparently unrelated sequences
2. Repeat: a sequence that only forms a stable structure if present in multiple copies
3. Motif: a short fragment
4. Family: a sequence compounded of the above units

5.2 Methods

5.2.1 Datasets

The Pfam 26.0 release used here contains 13672 families. These are filtered, processed and divided into a training set of 634 families, a test set of 645 families, and a prediction set of 1569 families as follows:

1. For each family a representative sequence is selected (which will not usually have an associated structure). This is the sequence with the greatest number of match states in the initial HMM, and with $> 80\%$ of its amino acids corresponding to match states.

2. A sub-alignment about this sequence is extracted by deleting columns where the representative sequence has a gap, or an amino acid corresponding to an insert state in the initial HMM.

3. PSICOV is run on each of these alignments. Families with fewer than 500 sequences are discarded (leading to a loss of 9012 families). Families with fewer than $\max(100, subLen)$ effective sequences (an estimate of the number of non-redundant sequences in the alignment) are also discarded (a further loss of 1812 families). Here $subLen$ denotes the length of the sub-alignment.

4. The Pfam database provides a list of fragments of PDB structures that are present in each Pfam family. If a PDB fragment overlaps a SCOP domain by $> 60\%$ the length of the larger of the two, and is $> 60\%$ the median length of a sequence in the Pfam family, then SCOP annotations from the fragment are assigned to the family.

5. If no SCOP annotation can be assigned, the family is placed into the prediction set. This set is a mixture of Pfam families that span many SCOP domains or a fraction of a domain, families that include PDB structures that are not in the SCOP database, and families with no associated structure.

6. The remaining families are partitioned into training and test sets. No family in the test set shares the same SCOP superfamily as a protein in the training set, and vice versa.

Steps 1 and 2 are designed to extract a maximal-sized alignment with comparatively few gaps. Step 3 aims to omit Pfam families containing too few sequences for a signal of correlated substitution to be detected. The training and test sets allow the use of the SCOP gold standard to compare fold recognition on proteins at different levels of structural similarity.

A representative structural domain from SCOP 1.75 is assigned to each of the training and test set families. The assignment is performed by taking the top hit from searching each Pfam family against the ASTRAL-1.75 database using HHsearch. When a sequence in the Pfam alignment has $> 70\%$ sequence identity over the full length of the representative structure its contact predictions are mapped onto the structure. Mappings exist for 544/634 and 535/645 of the training and test set alignments respectively.

5.2.2 Subsets of data for specific tests

The conservation of correlated substitutions, and the conservation of contacts undergoing correlated substitution, are studied using subsets of the $1079 = 544 + 535$ Pfam alignments for which it is possible to map correlated substitutions onto representative structures. The existence of these mappings allows us to identify equivalent positions by structure alignment. In both cases, correlated substitutions are obtained by running PSICOV v1.10 with the option '-d 0.03'. The subsets are selected as follows.

For assessing conservation of correlated substitutions, three subsets of Pfam alignments are constructed according to the SCOP hierarchy. The family set is composed of 67 pairs of Pfam alignments with the same SCOP family, taking at most one pair per family. The superfamily set is composed of 122 pairs with the same superfamily but different families, taking one pair per superfamily. Similarly, the fold set is composed of 65 pairs with the same fold but different superfamilies, taking one pair per fold. A fourth set is obtained by randomly pairing alignments from the fold set.

We define two residues as being in contact if their $C_\beta - C_\beta$ distance is < 8Å [Horner *et al.*, 2008]. If either residue lacks a C_β atom the C_α atom is used instead. The use of C_α atoms is not confined to glycine, as many PDB files contain residues without resolved C_β atoms. For assessing the conservation of contacts, Pfam alignments are filtered to those with at least 4 structures of the same SCOP family sharing $< 40\%$ sequence identity AND at least 4 structures in different families but the same superfamily AND at least 4 structures in different superfamilies. Of those Pfam alignments satisfying these criteria, at most one is selected per SCOP fold. This results in a set of 18 different Pfam alignments possessing different folds (Table 5.1).

Care has to be taken when comparing the conservation of contacts across structures, as rigid-body alignment can fail to recognise locally conserved contacts in the presence of a global change in conformation. To avoid this issue we only compare conservation at positions in the representative domain of the Pfam family that are aligned to $> 75\%$ of the related structures. A second difficulty is that pairwise structure alignment of a set of proteins often produces inconsistent assignments of equivalent residues – that is, the same contact in two structures may be mapped onto different pairs of residues in the representative domain. We use the Fr-TM-align program [Pandit and Skolnick, 2008], which has been found to produce more consistent alignments than several alternatives [Sadowski and Taylor, 2012].

5.2.3 The FORECAST fold recognition algorithm

FORECAST compares a query multiple sequence alignment with a library of structures, and scores each structure by its probability of having the same fold as the query. The algorithm uses only predicted secondary structure from PSIPRED [Jones, 1999b] and correlated substitutions from PSICOV. No direct use is made of amino acid sequence. The structure library is a non-redundant version of the ASTRAL 1.75A dataset. This contains 11211 structures corresponding to domains in the SCOP 1.75A release such that no two structures share more than 40% sequence identity. Structures containing only C_α atoms are excluded, as are those that consist of more than one chain.

Table 5.1: Details of the 18 Pfam families used to assess conservation of contacts identified by correlated substitution

Pfam ID	SCOP	Representative	# Structures			% Residues	# Contacts	
	annotation	domain	Fam	Sfam	Fold	assessed	Correlated	Uncorrelated
PF00046	a.4.1.1	d2e1oa1	15	17	13	63.2	6	25
PF09279	a.39.1.7	d1qasa1	9	9	4	37.2	0	31
PF07686	b.1.1.1	d1mjuh1	43	5	26	53.4	13	131
PF00419	b.2.3.2	d2uy6b1	9	6	9	54.7	8	155
PF08207	b.34.5.2	d1ueba1	4	4	15	57.1	2	65
PF01423	b.38.1.1	d1mgqa_	8	7	4	56.8	2	95
PF01336	b.40.4.1	d1c0aa1	5	12	14	48.1	12	114
PF00908	b.82.1.1	d2ixca1	5	16	5	36.4	20	170
PF00215	c.1.2.3	d1eixa_	7	7	28	71.0	21	365
PF00072	c.23.1.1	d1dbwa_	24	7	13	78.0	21	140
PF00075	c.55.3.1	d1rila_	6	8	6	38.8	10	99
PF00240	d.15.1.1	d1z2ma2	30	8	14	67.1	3	53
PF02136	d.17.4.2	d1gy7a_	7	31	6	43.8	1	100
PF00076	d.58.7.1	d1x4ea1	68	6	35	76.4	13	108
PF02800	d.81.1.1	d1k3ta2	9	5	4	22.5	7	69
PF01590	d.110.2.1	d3c2wa1	6	5	8	31.8	11	84
PF00008	g.3.11.1	d1tpga1	20	6	15	26.8	1	15
PF01096	g.41.3.1	d1tfia_	7	4	14	36.0	1	29
						Total	**152**	**1848**

In overview, FORECAST divides correlated substitutions identified by PSICOV into two sets according to their score. The higher scoring correlations are augmented by secondary structure prediction to generate a predicted 'contact map', which is then aligned to the real contact map of each potential template structure using the Al-Eigen program [Di Lena *et al.*, 2010]. A contact map for a protein of length L is an $L \times L$ symmetric matrix with binary entries corresponding to whether a contact exists (1) or does not exist (0) between the indexed positions. The lower scoring correlations are used to assess the significance of this alignment. The scores from this correlated substitution phase are supplemented by a score from the alignment of secondary structure elements. These stages are now described in more detail, in the order of their execution.

5.2.4 Generation of a predicted contact map for alignment

Correlated substitutions for each Pfam alignment are obtained by running PSICOV v.1.05 with default options.[1] The output of the program is a PPV score, which is an estimated probability that the correlated substitution indicates a contact. This score is a monotonic rescaling of the raw correlation score. Correlations with PPV scores of > 0.4 are treated as predicted contacts used for contact map alignment. Thresholds in the range $0.3, 0.35 \ldots 0.5$ were tried with little clear effect, so the choice of 0.4 is arbitrary. The value of the PPV threshold represents a compromise: decreasing the threshold allows alignment to be made with more correlated substitutions, but simultaneously reduces the number of correlated substitutions used to assess alignment significance.

Correlated substitutions above the threshold are refined by incorporating secondary structure predictions from PSIPRED. The refinement rules are as follows:

- Each residue in a helix may not contact any residues in the same helix other than the 4 residues to either side.

[1] This is an older version than that used in Section 5.2.2, reflecting the order in which the results were obtained.

- Each residue in a strand may not contact any residues in the same strand other than the two adjacent residues along the protein backbone.

- The $n + 2^{th}$ residue of a strand may not be in contact with residues $1 \dots n$ before the strand. Symmetrically a residue $n - 2$ positions before the end of a strand may not be in contact with residues $1, 2 \dots n$ positions after the strand.

This final requirement enshrines the fact that β-strands run straighter than typical protein backbones, such that the n^{th} residue after the start of a strand will typically be further away in space from the start of a strand than is the n^{th} residue before the start of the strand.

5.2.5 Contact map alignment

There is, to our knowledge, no program that aims to align predicted contact maps with real contact maps. However, several algorithms exist for aligning two sets of real contact maps. Here we use the Al-Eigen program [Di Lena *et al.*, 2010]. Al-Eigen makes use of the fact that a contact map C of size $n \times n$ can be decomposed into a series of n eigenvectors $\mathbf{v_i}$ each of length n such that

$$C = \sum_{i=1}^{n} \lambda_i \mathbf{v_i} \otimes \mathbf{v_i} \tag{5.6}$$

where λ_i is the eigenvalue of the i^{th} eigenvector. In what follows we choose to order the vectors such that $|\lambda_1| \geq |\lambda_2| \geq \dots$ Alignment is made using the same standard dynamic programming approach as in sequence alignment, but with a unique scoring function. The score for matching column a from a contact map decomposed into eigenvectors $\mathbf{v_i}$ with eigenvalues λ_i^v and column b from a contact map decomposed into eigenvectors $\mathbf{u_i}$ with eigenvalues λ_i^u is

$$S_{a,b} = \sum_{i=1}^{t} \sqrt{|\lambda_i^v||\lambda_i^u|}(v_i)_a(u_i)_b. \tag{5.7}$$

One complication is that the sign of the entry of each eigenvector is related to the sign of

the corresponding eigenvalue, yet equation 5.6 is not altered if $\mathbf{v_i} \rightarrow -\mathbf{v_i}$. To circumvent this problem, an alignment is made for every possible permutation of signs of the first t eigenvectors. To keep execution time low alignments are tried for $t = 1, 2 \ldots 6$ and only the best of all these alignments is reported. The best alignment is that which maximises the number of overlapping contacts.

The gap penalty for these alignments is the smaller of 0 and the smallest value of $S_{a,b}$ reported for the current value of t and permutation of eigenvector signs. The total execution time is usually less than one second.

5.2.6 Assessment of the significance of an alignment

An outline of FORECAST's assessment procedure is shown in Figure 5.2. The Al-Eigen alignment allows the contact map of a candidate structure (top-centre in black and white) to be related to predictions from PSICOV (bottom-centre, shown as a contact map but with PPV scores rather than binary entries). After discarding positions with no PSICOV predictions, every remaining pair of positions is compared to the equivalent pair in the candidate structure. The associated PPV score is assigned to one of three lists: a list of scores that neither support nor oppose the hypothesis that the candidate structure has the same fold as the query alignment, a list of scores that support this hypothesis, and a list of scores that oppose this hypothesis. The three lists are populated as follows:

A score neither supports nor opposes the hypothesis if it involves a column that is matched to an indel by Al-Eigen OR corresponds to a contact inserted during secondary structure based refinement OR involves two columns separated by fewer than 5 positions (e.g. columns i and $i + 4$).

A score supports the hypothesis if it was not used in the alignment AND a corresponding contact is observed in the candidate structure. Note that this definition excludes all scores with $PPV > 0.4$.

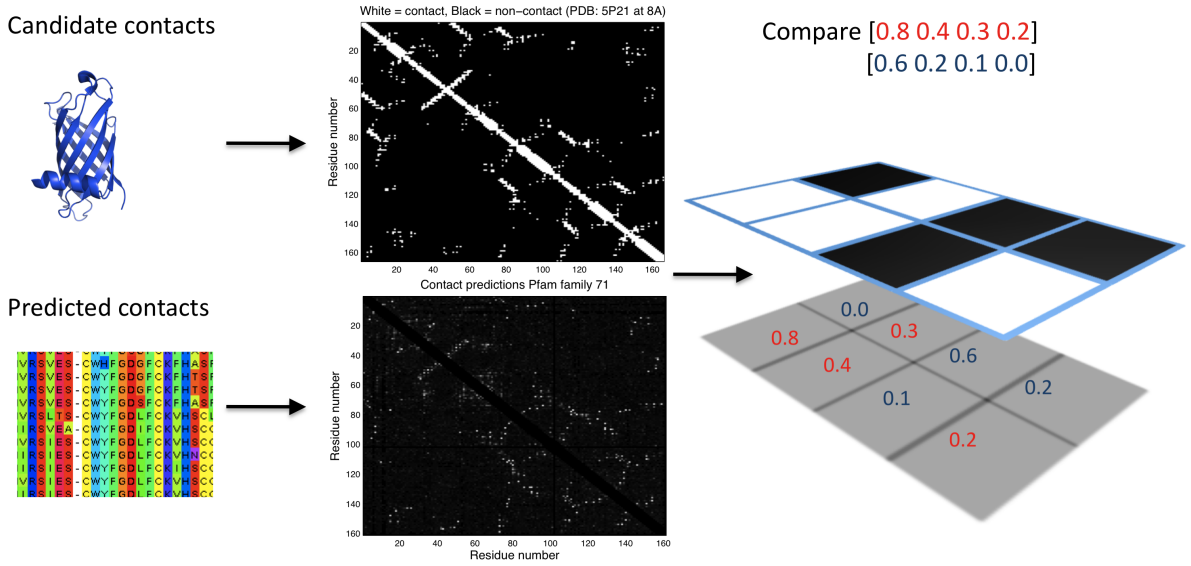


Figure 5.2: Outline of the FORECAST alignment assessment procedure. A contact map from a candidate structure is aligned with a contact map from a set of predicted contacts. Correlated substitutions are divided into two sets depending on whether they correspond to contacts (red) or non-contacts (blue) in the aligned structure. If the candidate has the correct fold, the red set of numbers should be larger on average than the blue set.

A score opposes the hypothesis if it belongs to neither of the above groups AND no corresponding contact is observed in the candidate structure. Note that this includes some scores used in the Al-Eigen alignment.

A Wilcoxon rank-sum test is used to estimate the probability that the list of supporting scores tends to contain higher values than the list of opposing scores. Typically the supporting list is an order of magnitude smaller in size than the opposing list, and the total number of scores is $\sim 10^4$, permitting the probability to be approximated by a normal distribution, and the z-score of this distribution to be reported.

5.2.7 Final fold-level prediction

The above-described fold recognition method produces a list of z-scores for each query – one for each candidate structure. As expected, the distribution of z-scores for a given query or candidate structure is approximately normal. However, certain queries and candidate structures are 'sticky': they have systematically larger mean or variance in their z-scores. To make all scores comparable, we rescale the z-scores for each structure to have zero mean and unit standard deviation. We similarly rescale the z-scores for each query. The final score for a match between a query q and a candidate structure s is the sum of these two independent rescalings of the original z-score, and a secondary structure term (described in the next section; the weighting of 0.15 was optimised on our training set):

$$score(q,s) = \frac{z_{qs} - \langle z_{qs} \rangle_q}{\sqrt{\langle z_{qs}^2 \rangle_q - \langle z_{qs} \rangle_q^2}} + \frac{z_{qs} - \langle z_{qs} \rangle_s}{\sqrt{\langle z_{qs}^2 \rangle_s - \langle z_{qs} \rangle_s^2}} + 0.15 \times sstruc(q,s) \tag{5.8}$$

where z_{qs} is the z-score for query q and candidate structure s, and e.g. $\langle z_{qs} \rangle_s$ denotes the average value of z over all candidate structures for query q.

A measure of certainty that the top-ranked candidate shares the same fold as the query is calculated by summing the difference between the scores of the top-ranked candidate structures of the same fold and the score of the next highest-ranked candidate structure with a different fold. For example, if the top two candidate structures had fold 'a.4.' with scores 4.3 and 4.1, but the third candidate had fold 'a.5' with score 3.8 then the final value would be: $(4.3 - 3.8) + (4.1 - 3.8) = 0.8$. Lower-ranking candidate structures possessing these folds would be ignored.

Although the scoring system is heuristic, adding the scores roughly corresponds to assuming that the scores of candidate structures of the same fold are independent. Taking the difference between top-ranked scores turns a measure of compatibility between the query and candidate structure into a measure of discrimination between different candidate structures.

5.2.8 Secondary structure element alignment

Our implementation of secondary structure element alignment is similar to that of [McGuffin *et al.*, 2001], which was itself evaluated at the task of identifying known folds with no sequence-detectable neighbours [McGuffin and Jones, 2002]. We divide PSIPRED predictions for the query, and DSSP strings for the candidate structure, into elements of secondary structure. Two elements a and b are aligned with the score:

$$S_{a,b} = \begin{cases} \min(\text{len}(a), \text{len}(b)) & \text{if } a \text{ and } b \text{ have the same secondary structure} \\ 0 & \text{if } a \text{ is strand and } b \text{ is helix or vice versa} \\ \frac{1}{2}\min(\text{len}(a), \text{len}(b)) & \text{otherwise} \end{cases} \tag{5.9}$$

There are no gap penalties. The optimum alignment is that which maximises the sum of the above score over all aligned elements. The score of the optimum alignment is expressed as a percentage of the maximum score obtained by aligning either of the secondary structure strings with itself. This gives the $sstruc(q, s)$ of Equation 5.8.

5.2.9 Comparator methods

We compare contact-based fold recognition with methods based on structure alignment (TM-align, Zhang and Skolnick [2005]), HMM-HMM alignment augmented by predicted secondary structure (HHsearch, Söding [2005]), and structural alphabet alignment (fastSCOP, Tung and Yang [2007]).

The TM-score from TM-align is normalized by a sequence length (Equation 1.1). We chose this to be the maximum of the lengths of the query and candidate structure, as this led to most effective fold recognition.

HHsearch outputs three scores for each match of a query with a fold: the raw score for sequence based alignment, the raw score for the alignment of the secondary structure, and a

probability for the match to be correct (expressed as a percentage). We use this last measure throughout the text.

We also experimented with combining HHsearch and FORECAST by adding the maximum HHsearch probability for a query to have a particular fold, $P(q, \text{fold of } s)$, to all FORECAST candidate structures with that fold. HHsearch typically reports only a handful of possible folds, so for most candidate structures $P(q, \text{fold of } s) = 0$. This measure is combined with a rescaling of the FORECAST score (Equation 5.8) to give Equation 5.10. The scaling factor of 25 was optimised on our training set.

$$combined(q, s) = \max(P(q, \text{fold of } s)) + 25 \times score(q, s) \tag{5.10}$$

5.3 Results

5.3.1 Is there a signal of homology in correlated substitutions?

Two general schemes can be suggested for recognising good templates for comparative modelling from correlated substitutions. In the first scheme, two sets of correlated substitutions are compared where one set is associated with a known structure. This scheme requires distantly related proteins to share signals of correlation. The second scheme exploits the large degree of overlap between correlated substitutions and contacts by comparing one set of correlated substitutions with the contacts of a potential template. It requires distantly related proteins to conserve contacts that are identified by correlated substitutions.

To test whether these conditions are met we used a data set of 1079 sequence families from the Pfam database. We assigned a SCOP annotation to each of our families, allowing us to distinguish pairs of families according to their level of relatedness. Each family was also associated with a structure, allowing us to identify equivalent positions in two families through structure alignment. Conservation of contacts was assessed by aligning this reference structure with many related structures gathered from a non-redundant ($< 40\%$ sequence identity) subset

of the ASTRAL Compendium [Chandonia *et al.*, 2004].

The extent of correlated substitution between two positions was determined using the PSICOV program [Jones *et al.*, 2012]. In this section we refer to positions with $PPV > 0.5$ as being 'highly correlated' (a threshold also used in Nugent and Jones [2012]).

Scheme 1: Conservation of correlated substitutions

Distantly related proteins did not share many correlated substitutions, suggesting that fold recognition by comparing sets of correlated substitutions will prove difficult. Among pairs of Pfam alignments sharing the same SCOP family only 23% of highly correlated ($PPV > 0.5$) positions in one family were highly correlated in the other. This fraction decreased to 16% for proteins in the same SCOP superfamily, and 12% for proteins in the same fold (Table 5.2). Details of the selection of pairs of proteins are given in Section 5.2.2. These levels were higher than those obtained by randomly shuffling the PPV scores ($< 3\%$ in all cases), or randomly pairing proteins in different folds (7%).

Table 5.2: The number of highly correlated positions ($PPV > 0.5$) shared between pairs of proteins of the same SCOP family, superfamily, and fold, and between randomly paired different folds.

Level	Pairs	Number of correlations in			Percentage
		Smallest	Largest	Both	
Family	67	1331	2079	406	23.4
Superfamily	122	1756	3132	337	16.1
Fold	65	451	893	63	12.3
Random	65	167	423	13	7.2

The number of correlations is aggregated over all the pairs at each level. Among each pair the number of correlated substitutions present only in the protein with fewest correlated substitutions is added to the 'smallest' column, and the number of correlated substitutions only present in the other protein is added to the 'largest' column. The number of shared correlations is added to the 'both' column. Percentages are calculated as $100 \times \text{both}/(\text{smallest} + \text{both})$. As the relationship between the proteins becomes more distant the proportion of shared correlated substitutions decreases.

Scheme 2: Conservation of contacts identified by correlated substitution

The conservation of contacts identified by correlated substitution was assessed on a set of 18 Pfam families (listed in Table 5.1, selected as described in Section 5.2.2). We found as expected that more distantly related proteins were less likely to share a contact regardless of whether or not that contact was undergoing highly correlated substitution (Figure 5.3). However, among proteins in the same family or superfamily, contacts subject to high levels of correlated substitution ($PPV > 0.5$, black boxes) were generally more conserved than those that were not (blue boxes). There was no statistically significant difference at the fold level ($p > 0.01$, Wilcoxon rank-sum test). The same conclusions held when the conservation threshold described in Section 5.2.2 was lowered from 75% to 60%, and on a larger set of 32 Pfam families (obtained by loosening the cut-offs described in Section 5.2.2 to require only 3 structures per SCOP level), except that on the larger set the difference in fold level conservation was also significant.

In many cases a correlated substitution corresponded to a contact that was absent in the most closely related structure, but present in a more distantly related structure. These cases are a good example of the homology information that can be present in correlated substitutions. We classified these according to the lowest level of the SCOP hierarchy at which the contact was observed. The top 10% of such correlations first appearing as contacts at the family level were stronger (average PPV score of 0.306) than those first appearing at the superfamily level (0.202) or fold level (0.187). This ordering was robust as the percentage was changed.

Summary

The picture that emerges from these results is of correlated substitutions representing structurally conserved contacts in a set of closely related proteins. More distantly related proteins share a decreasing number of correlated substitutions. A simple interpretation of this is that the evolutionary constraints that give rise to correlated substitutions may strengthen or weaken over time. Even after a correlation has ceased to represent a structural contact, the signal of that correlation may remain detectable, decaying over time.

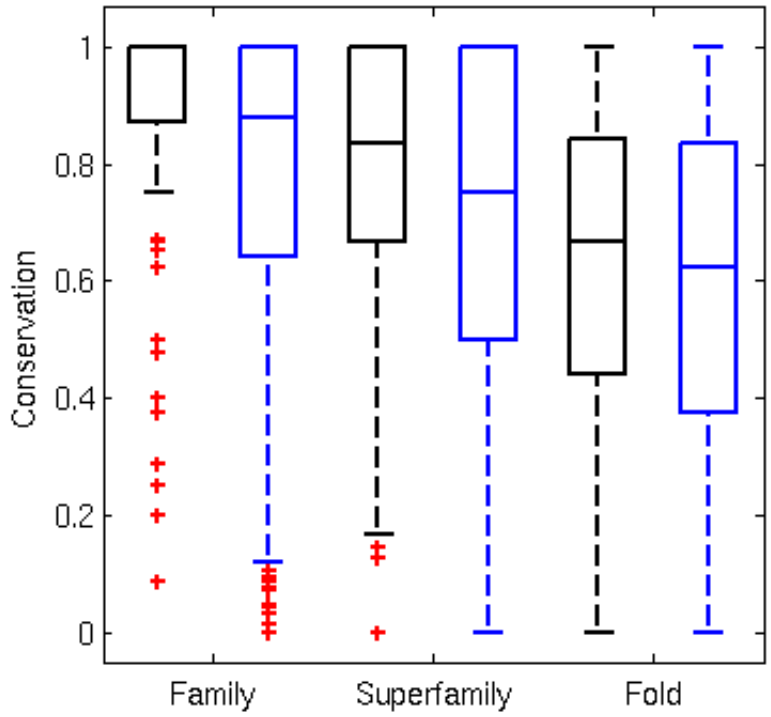


Figure 5.3: Contacts in 18 protein structures are divided into two sets depending on whether they are undergoing a high level of correlated substitution ($PPV > 0.5$, black boxes) or not (blue boxes). Contacts undergoing correlated substitution are more conserved at the family and superfamily levels than contacts which are not. At the fold level there is no detectable difference.

All fold recognition methods decrease in accuracy as more distantly related pairs of proteins are matched. As correlated substitutions encode information about structure they have the potential to perform distant fold recognition with the high accuracy of a structural alignment program. The above results suggest that realising this potential will be challenging. As contacts identified by correlated substitutions are more conserved than the signal of correlated substitution itself, scheme 2 above appears the more promising approach. We next develop this scheme into a fold recognition technique.

5.3.2 Fold recognition using correlated substitutions

Motivated by the above results, we developed a fold recognition algorithm, 'FORECAST' (FOld REcognition by Contact Adjusted STatistics), that compares a set of correlated substitutions with contacts from potential templates. In the previous section this comparison was aided by the use of a structural alignment to establish equivalent positions between the two proteins, but this is impossible in fold recognition where one set of sequences has no known structure. To circumvent this problem, we divided correlated substitutions into a high scoring set ($PPV > 0.4$) and a low scoring set. We constructed a contact map from a combination of the high scoring set and contacts inferred from predictions of secondary structure, and aligned this map with that of a potential template using the contact map alignment program Al-Eigen [Di Lena *et al.*, 2010].

The consistency of this alignment with the hypothesis that the template has the correct fold was then assessed using the low scoring set. The ability to extract a strong structural signal from low-scoring correlated substitutions is significant, as they have generally been excluded from structure prediction approaches [Marks *et al.*, 2011]. Assessment using correlated substitutions alone was effective, but often failed to distinguish the true fold from folds with similar topology but different secondary structure. The addition of secondary structure to the scoring removed this degeneracy (see Section 5.2.8 for scoring details).

Prediction of correlated substitutions from direct information requires a large number of sequences: current opinion holds that the effective number of sequences must be greater than the length of the alignment. We found that sequence families that violated this rule almost always received low scoring predictions, so it is likely that no particular care needs to be taken in preparing input predictions for FORECAST: poor input produces no output. This is a useful behaviour that stands in contrast to *ab initio* model building where poor data will still produce a model whose quality must be separately assessed.

The soluble protein models with a TM-score > 0.5 made by Marks *et al.* [2011] used ≥ 10000 sequences, whereas FORECAST predictions supported by HHsearch are found for sequence

families as small as our 500 sequence cut-off ($PF04456$, aligned length of 90 residues). This reduction in the number of required sequences results in a large increase in the number of sequence families amenable to prediction. Only 321 Pfam 26.0 families contain ≥ 10000 sequences, whereas 4660 families possess ≥ 500 sequences.

To demonstrate the use of correlated substitutions for fold recognition, we assigned SCOP folds (e.g. 'b.1.') to sequence families in the Pfam A database. Our FORECAST algorithm was developed on a training set of 634 Pfam families for which there was already a known SCOP assignment. A similarly constructed test set of 645 Pfam families served as a benchmark. Predictions were then made on a further 1569 Pfam families for which automated SCOP assignment proved impossible. Sequence families in Pfam are labelled as 'domain', 'family', 'repeat' or 'motif' and do not necessarily correspond to structural domains. In principle it is not possible to assign a fold to a repeat or motif, as they form only a fraction of the structure of interest. Nevertheless, they were kept in all the sets to test the robustness of our method to false positives.

Fold recognition can be attempted at several levels of difficulty. Consider attempting fold recognition on a Pfam alignment with real SCOP annotation b.1.18.21. At the easiest 'family level' matches are possible to candidate structures with the same annotation. At the superfamily level all structures with SCOP annotation b.1.18.21 are excluded, and at the fold level all structures with SCOP annotation b.1.18.x are excluded, where x is any number.

On our training set, HHsearch, a state of the art method, returned a top hit with the correct fold for $\sim 80\%$ of queries at the superfamily level and $\sim 35\%$ of queries at the fold level, consistent with the findings of others [Yang *et al.*, 2011]. A structure alignment algorithm, TM-align [Zhang and Skolnick, 2005], achieved $\sim 60\%$ accuracy for its top-hit at the fold level. This 25% increase in recognition accuracy from using structural information suggests that structural constraints from correlated substitutions will be most effective at fold level recognition. In what follows, unless otherwise stated, only fold level recognition is discussed.

5.3.3 Fold recognition benchmarking

In assessing fold recognition performance it is important to recognise that real-life queries may have novel folds. To simulate this we removed all fold recognition results from our test set with the same superfamily as the query. After this, approximately half of our test set could be matched at the fold level, and half had no possible match (acted as 'novel' folds). The number of folds that were correctly identified in the top 1, 5, and 10 hits of various methods are shown in Table 5.3. FORECAST achieved 35 correct top hits without the secondary structure component of its score, but substantially benefited from the addition of secondary structure information. For comparison, randomly assigned scores typically led to < 10 correct top hits.

HHsearch recognised the correct fold nearly twice as often as FORECAST on our test set (107 vs 64 top hits), but was less able to distinguish when it had done so. This is apparent in Figure 5.4 where FORECAST initially produces more true positives for a given number of false positives, despite having fewer true positives from which to choose. When the two approaches were combined they proved complementary to each other: a marginal increase in the number of top hits with the correct fold (112 vs 107) was accompanied by much greater discrimination (red line, Figure 5.4). When the contribution of secondary structure was removed from the FORECAST score, the same improvements were seen (22 true positives for 5 false positives, 64 true positives for 40 false positives) showing that the correlated substitutions produced the increased discrimination.

We plotted the fold recognition scores from FORECAST and HHsearch against the TM-score of their top hits to the representative structure for each query.[1] The results are shown by the blue and purple points in Figure 5.5. A point is blue if the top hit has an incorrect SCOP fold, and otherwise is purple. A perfectly performing method would have an approximately equal number of blue and purple points (all possible fold level matches are made) clearly separated

[1]The assignment of representative structures to queries is described in Section 5.2.1, briefly it is performed by taking the top HHsearch hit at the family level (i.e. using an unfiltered database). The TM-score was calculated using fr-TM-align over the region of the representative structure that HHsearch determined to be homologous to the query.

Table 5.3: Performance of different methods on our test set, assessed as the number of true folds identified in the top-scoring 1, 5, and 10 folds. The numbers are out of a possible 278 matches.

Method description	Queries with true fold in		
	Top 1	Top 5	Top 10
FORECAST (no *sstruc*)	35	75	93
Secondary structure	53	110	127
FORECAST	64	116	140
HHsearch †	107	*112*	*113*
HHsearch + FORECAST	112	152	166

† The results list of HHsearch was sometimes filled before producing 5 or 10 predicted fold types. This means that the italicised 'Top 5' and 'Top 10' figures are not representative of the power of the method.

along the x-axis (good fold recognition), and with many of these points sitting above the line at TM-score 0.5 (many top hits could form a basis for a structural model).

For comparison, Figure 5.5 also includes points for correctly identified top hits at the superfamily level (green) and correctly identified top hits at the family level (yellow). Most yellow points are occluded by green points, highlighting that some superfamilies are no more distantly related to each other than some families.

The fold recognition score from FORECAST depends on how compatible a structure is with constraints predicted by correlated substitutions. Hence, the top hit for a given query should tend to have a high TM-score to the native structure. This is apparent in the left panel of Figure 5.5 where almost no hits have TM-scores of < 0.2 and where correctly identified SCOP assignments are always made to structurally similar top hits (the lowest non-blue point has a TM-score of ≈ 0.4). This also gives rise to bands of sfam/fold/novel matches: the most structurally similar match across superfamilies is more similar than the most structurally similar match across folds. As the scoring function of HHsearch is based on sequence properties, its banding of sfam/fold/novel matches is less clear, and the link between recognition score and TM-score is weaker (right panel, Figure 5.5).

Crucially, it is possible to choose a score threshold for FORECAST (vertical black line, left

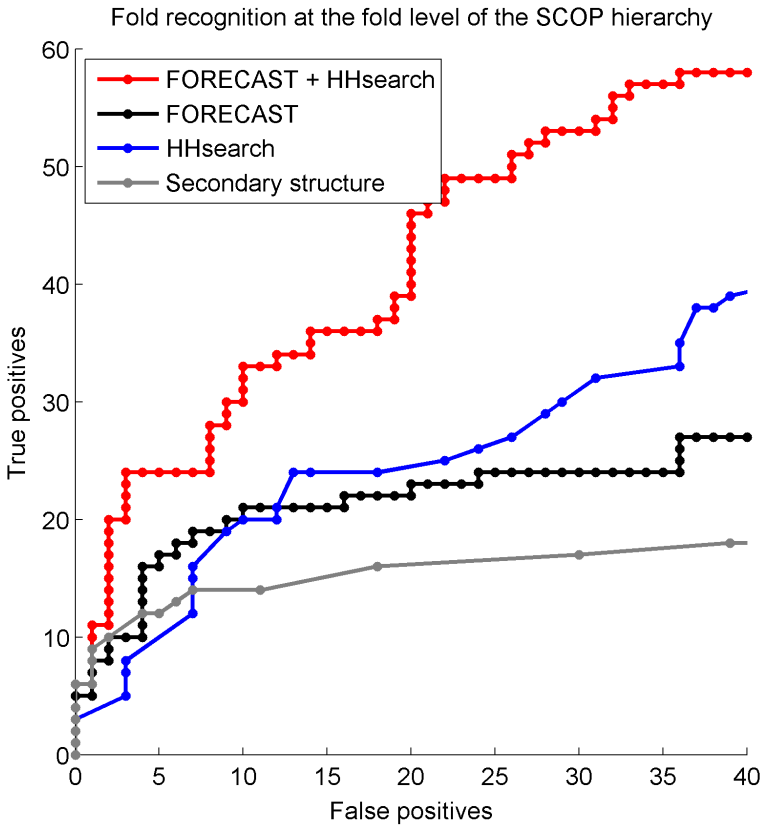


Figure 5.4: True positives vs false positives for each method as the fold recognition threshold for a 'positive' score is decreased. There are 278 potential true positives in the test set, but each method matches only a fraction of these. In conjunction with the 'Top 1' column of Table 5.3 this shows the ability of a method to discern when its top hit is correct.

panel, Figure 5.5) such that almost any hit scoring above the threshold could form the basis of a good structural model. This threshold, used in the next section, corresponds to a FORECAST score of 3.5, and led to 16 correct top hits for 5 false positives at the fold level.

5.3.4 Novel fold predictions

We attempted to assign SCOP folds to 1569 Pfam families for which our automated SCOP assignment failed (automated assignment is described in step 4 of Section 5.2.1). These were a mixture of families with known structures but no SCOP annotation, and those without any

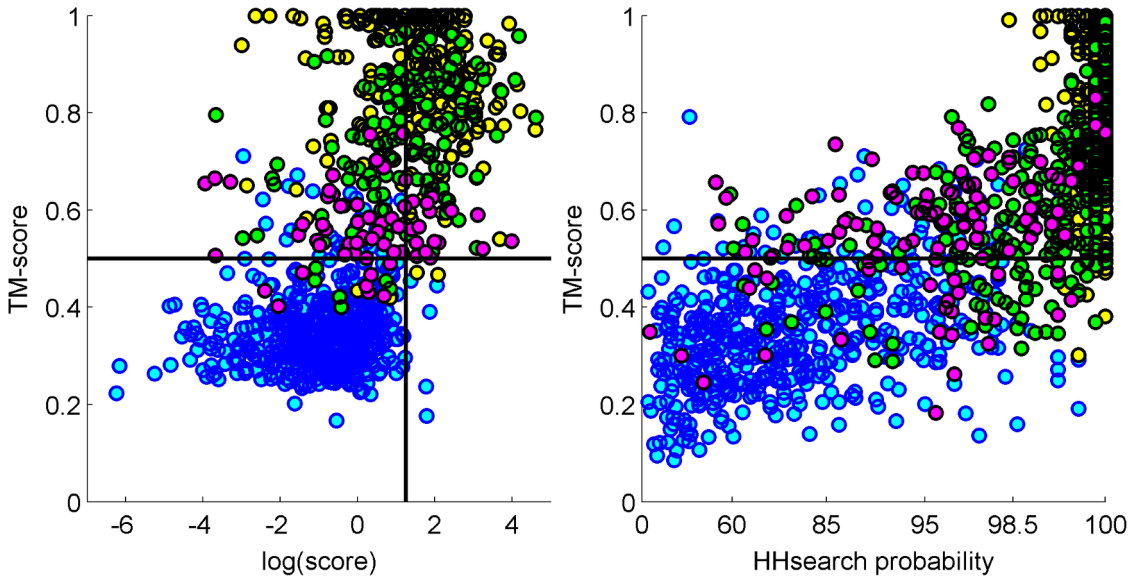


Figure 5.5: Relationship between the fold-recognition score and the structural similarity of the top hit to the representative structure for FORECAST (left) and HHsearch (right). Each point represents the top hit of a fold recognition query. Points are colour-coded by the SCOP relationship of the hit to the query: yellow points have the same family, green the same superfamily, purple the same fold, and blue a different fold. Points above the horizontal black line at TM-score 0.5 represent hits that could be used as templates for comparative modelling.

structure in the Pfam alignment. FORECAST annotated 57/688 and 43/881 of these respectively with a score higher than 3.5 – the threshold shown by the vertical black line in Figure 5.5. All Pfam families were also annotated by HHsearch, and those with a known structure were annotated with fastSCOP [Tung and Yang, 2007].

The majority (73/100) of FORECAST's annotations agreed with those of the other methods. Of the differences, 8/27 were caused by varied assignments of folds that consist of a small number of long helices (e.g. 'a.7.' (spectrin repeat-like), 'a.38.' (HLH-like), 'f.3.' (light-harvesting complex subunits), 'f.17.' (transmembrane helix hairpin), 'f.21.' (Heme-binding four-helical bundle), and 'f.23.' (single transmembrane helix)) and a further 5/27 were short fragments ($\leq$ 40 amino acids). Correct fold assignment is not important for these cases as the fold provides little information beyond that which can be obtained from secondary structure

prediction, or programs that detect transmembrane helices such as TMHMM [Krogh *et al.*, 2001]. The remaining 14 differing annotations are listed in Table 5.4, and some are discussed below.

Table 5.4: Pfam families of unknown fold that are assigned different folds by FORECAST and other methods

Pfam ID	Known Structure?	No. seqs	FORECAST Fold	FORECAST Score	fastSCOP & HHsearch Fold	fastSCOP & HHsearch HH Score	Notes
PF02465	No	1401	a.55	4.0	b.1.	77.3	
PF03382	No	2106	c.10.	8.0	c.15.	4.0	[1]
PF05506	No	803	b.1.	4.4	b.6.	88.0	
PF08808	No	930	b.2.	3.5	b.102.	11.2	
PF13173	No	2479	c.26.	6.7	c.37.	99.7	[2]
PF13437	No	3035	b.1.	4.4	f.46.	99.8	
PF13477	No	1348	c.23.	4.7	c.87.	99.3	[1,2]
PF13690	Yes	655	d.58.	7.9	d.252.	99.9	[5]
PF00271	Yes	60138	c.47.	5.0	c.37.	99.9	[5]
PF04500	Yes	869	b.40.	5.8	g.79.	96.9	[1]
PF06969	Yes	3507	a.4.	7.5	c.1.	99.5	[4]
PF07677	Yes	712	d.58.	5.3	b.2.	100.0	[3]
PF08445	Yes	640	b.40.	4.4	d.108.	99.8	[5]
PF13418	Yes	3138	g.41.	3.6	b.68/69.	98.9	[5]

[1] Described in main text
[2] Top hits identified by FORECAST and other methods share a CATH topology
[3] Secondary structure connectivity and overall topology is identical, but the top hits differ in secondary structure type
[4] The disagreement between the methods appears to be a result of misclassification by SCOP. The structure associated with this Pfam family consists of two domains, the larger of which is of fold c.1. HHsearch's full list of hits confirms that the smaller domain, which is the subject of this Pfam family, has fold a.4.
[5] Presumed false positive prediction from FORECAST

Individual examples show the value of the method. For example, FORECAST predicted sequence family $PF03382$ to be of fold 'c.10.' (Leucine-rich repeat), a suggestion also provided in the Pfam metadata, presumably from an expert analysis of the sequence motif. No structure is associated with the family, and whilst HHsearch also returns 'c.10.' as a hit, the probability

of this being correct is estimated at just 4%.

A second example is provided by family $PF04500$, the FLYWCH zinc-finger domain. This family has a single NMR structure for which no SCOP annotation has been assigned. fastSCOP returned no hits for this structure, and the best match from HHsearch was structurally dissimilar. The top match from FORECAST had superfamily 'b.40.4' (nucleic acid-binding proteins), and like the NMR structure possessed a tight five-stranded β-barrel (Figure 5.6), albeit with different chain connectivity.

Sometimes a disagreement between FORECAST and HHsearch revealed a potential evolutionary link between different folds. For example, $PF13477$ was matched by FORECAST to PDB code 1TOI chain A and SCOP fold 'c.23.', and by HHsearch to PDB code 1V4V chain A and SCOP fold 'c.87.' . Both proteins share a common region of > 100 residues that can be aligned to 3.2Å RMSD, with much of the difference due to the packing of a single helix against a different face of the protein structure (Figure 5.6). This is not a novel discovery – the proteins share a CATH topology – but here it is detected from sequence alone.

5.4 Discussion

We have explored the extent to which correlated substitutions can be used as a tool to detect protein homology. Our analysis centres on the use of the SCOP hierarchy to roughly partition proteins by relatedness: proteins in the same SCOP family mostly share a sequence-detectable link, which is generally more recent than that shared by proteins within the same superfamily. It is not clear whether proteins in the same fold are related by convergent or divergent evolution. As correlated substitutions often correspond to spatially proximate pairs of positions in protein structures, it is reasonable to infer a structural significance to these substitutions, and to expect similar structures to share many of the same correlations regardless of the family/superfamily/fold distinction.

We found this assumption to be incorrect: closely related proteins share significantly more

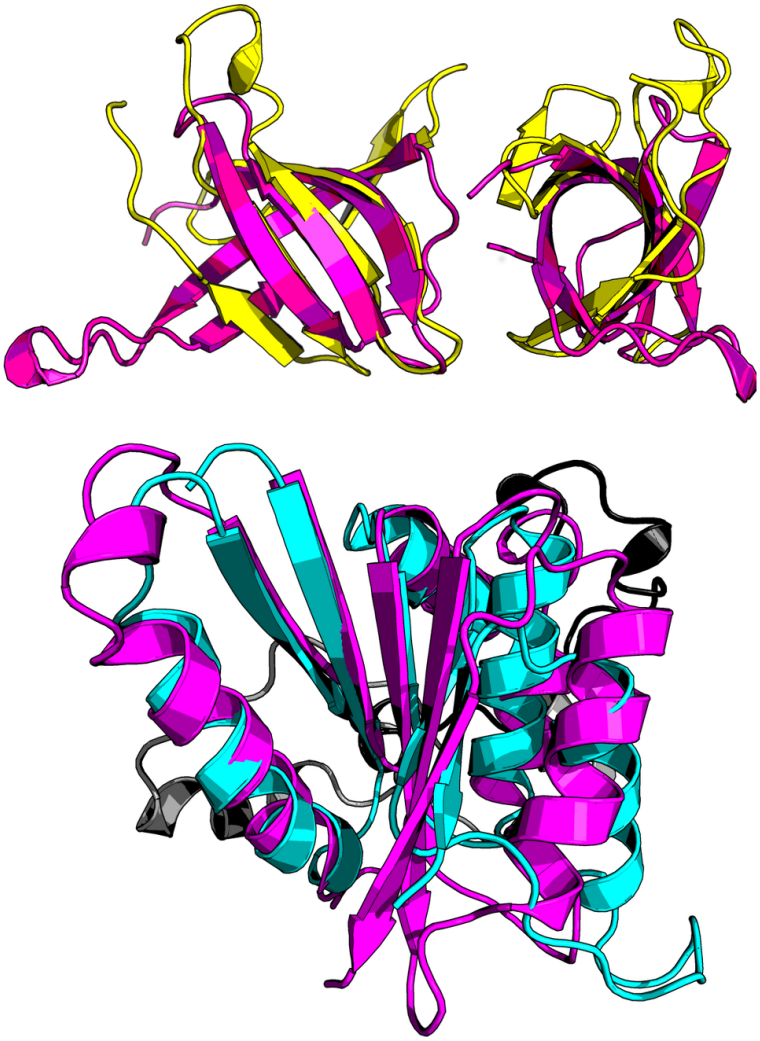


Figure 5.6: The FLYWCH zinc-finger domain **Top** (yellow) returns no fastSCOP hits, and a structurally dissimilar top hit from HHsearch. The RSCB PDB structural similarity tool shows at most 15% sequence identity with known structures (accessed 23/06/13). The top hit from FORECAST (pink) forms a similar barrel of 5 β-strands including a substructure of 3 strands. An additional strand is matched, but with different connectivity. **Bottom** Matches from FORECAST (magenta and black) and HHsearch (cyan and grey) for Pfam family $PF13477$. The structures are from different SCOP folds, but the relevant region only differs in the positioning of a single helix (black and grey), and in the addition of a terminal helix to one of the structures (not shown). The connectivity of the remaining 4 helices and 5 strands is conserved, and sequence identity is $\sim 16\%$. The structures share a CATH topology.

correlated substitutions than distantly related ones. However, this does not mean that correlated substitutions have no structural significance: contacts between positions undergoing correlated substitution are significantly more conserved within a protein family and superfamily than other contacts. There is also evidence of shifts in the set of correlated substitutions over time, with weak signals found at positions that correspond to contacts in a protein of the same superfamily or fold, but not in the extant protein.

Whilst the fold and function of a protein may necessitate a certain set of correlated substitutions, these results suggest that at least some correlations are determined by a protein's ancestry. This observation could form the basis of a molecular clock capable of establishing the divergence times of proteins with little sequence identity.

The decaying number of correlated substitutions that are shared over time also has implications for fold recognition. Conventional profile-profile alignment approaches are highly effective at recognising proteins of the same SCOP family and superfamily, so any competing approach must be capable of recognising more distant relationships. We suggested two ways of using correlated substitutions to find such relationships: comparison of two sets of correlated substitutions, and comparison of correlated substitutions from a query with contacts in a known structure. We found that only 12% of highly correlated substitutions were shared between proteins of the same SCOP fold, suggesting that comparing two sets of correlations is unlikely to be fruitful. We implemented the second approach in a program called 'FORECAST'.

FORECAST relies on the fact that correlated substitutions in a query will tend to correspond to contacts in a candidate structure even when a different set of contacts in the candidate is undergoing correlated substitution. The main advantage over conventional approaches is a stronger link between the fold recognition score and the structural similarity of the match. This allows a score threshold to be defined above which almost every hit could serve as a template for modelling. Or put another way, the false positive rate is extremely low. This comes at the cost of recall: even at the SCOP family level many queries return no significant matches.

Applying FORECAST to a set of 2848 Pfam families we achieved successful fold recognition

with as few as 500 sequences – the smallest family in our set. This is an order of magnitude fewer than previously used to build models with TM-scores > 0.5 from correlated substitutions. We attribute this to our use of all the available correlated substitution information rather than selecting highly correlated substitutions.

In fact, FORECAST's scoring function makes almost no use of highly scoring correlated substitutions, which has implications for the development of algorithms that detect correlated substitutions. Currently these algorithms are assessed by the number of their highest scoring correlations that correspond to contacts (e.g. Monastyrskyy *et al.* [2013]). This form of assessment favours algorithms that are useful for *ab initio* structure prediction, but which may prove ill-suited to FORECAST. Instead our use of a Wilcoxon rank-sum test requires that a correlated substitution of strength x is marginally more likely to correspond to a contact than is a correlated substitution of strength $x - \delta x$.

The results from FORECAST provide orthogonal information to that present in sequence profiles. The top hits from FORECAST and HHsearch sometimes belonged to different SCOP folds with sequence identity far below the 'twilight' zone, but high structural similarity (in two cases with the same CATH topology). These hits represent a detection of links between folds without the use of structural information. Additionally, even a naive linear combination of the scores from HHsearch and FORECAST substantially improved the discrimination of top hits with the correct fold.

Several promising approaches to improving FORECAST can be identified. One approach would be to develop heuristics to better divide the correlated substitutions into alignment and assessment sets. A certain minimal number of correlations are required for accurate alignment, yet our use of a fixed threshold occasionally leads to an empty alignment set. Another observation is that not all correlations provide independent information for the purposes of alignment: identifying pairs of positions that contain redundant alignment information and placing one of these into the assessment set would be a more effective use of the data.

A second approach would be to expand the range of structural information available to

FORECAST. This could most simply be done by increasing the size of the structural library. A natural extension of the method is to replace the binary contact maps of individual structures with contact profiles, where each entry specifies the fraction of related structures in which that contact occurs.

CHAPTER 6

Context and future directions

Each results chapter of this thesis includes a short discussion of the work it contains and suggestions for how it may be developed. The intent here is not to repeat that information, but instead to give an opinion of the place of each method in the wider context of bioinformatics techniques.

6.1 A future for ESSTs

Although the first two chapters of this thesis dealt exclusively with membrane proteins, the techniques developed in them have broader applications. More than 20 years after their introduction, environment specific substitution tables (ESSTs) are the exception rather than the rule in sequence alignment. The limited popularity of the ESST idea has several root causes, which we now address in turn:

Difficulty of annotation: In Chapters 3 and 4 environment annotations required a protein

structure, yet in many bioinformatics applications, no structures are known for the sequences under study.

Comment: Although environments have historically been defined from structure this is not a fundamental requirement. Since 1992 the per-residue accuracy of sequence-based secondary structure prediction has increased from $< 70\%$ to 80%. Prediction methods for solvent accessibility have also improved. Such predictions could be used to select an appropriate ESST for a position in a sequence for which no related structure is known. Alternatively, ESSTs could be built for predicted environments, or averaged over environments depending on the confidence value of a prediction.

Sequence profiles: The extreme case of an ESST arises in a sequence profile, where each position in a protein family forms a separate environment with its own scoring system. What additional benefit can a less specific environment confer?

Comment: In principle any method that uses a general substitution table such as BLOSUM62 could benefit from using ESSTs. General substitution tables are an intrinsic part of the scoring system for most programs that use sequence profiles.

Lack of structure data: Historically ESSTs have been made from sets of aligned structures. There is much less structural data than sequence data.

Comment: In Chapter 3 we showed that substitution tables built from sequence alignments closely resemble those built from structure alignments. In that chapter we also developed methods for assessing the quality of ESSTs, which could be applied when the quality of the underlying sequence alignment is uncertain.

6.2 Comments on different alignment methodologies

MP-T was developed to demonstrate that membrane specific information could improve alignments for comparative modelling. However, it also provided a test bed for the comparison of

profile-profile and multiple sequence alignment methods, albeit on the unusual case of membrane proteins. On our test set, the multiple sequence alignment methods generally misaligned fewer residues than the profile-profile methods.

In the recent CASP10 structure modelling competition, the majority of groups used profile based tools such as PSI-BLAST, FUGUE, HHsearch, and RaptorX. Why are multiple sequence alignment programs not more widely used for comparative modelling?

There are several reasons, one of which is convenience. Profile alignment lends itself to sequence search whereas multiple sequence alignment does not; all of the above-mentioned programs can perform fold recognition and alignment in a single step. A second reason is accuracy. Profile alignment programs often construct their own profiles, presumably in a way that complements the alignment algorithm, whereas there are no clear guidelines as to how sequences should be selected to give accurate multiple sequence alignments. This problem is discussed in the next section. Profile methods also typically include structural information whereas MSA programs typically do not; in regions of low sequence identity this additional information can be used to improve alignment quality, as we found when aligning transmembrane domains.

In principle, it should be possible to make a more accurate target-template alignment with a multiple sequence aligner than with a profile-profile aligner as, in addition to the information present in profiles, a multiple sequence alignment method can make use of inferred evolutionary relationships between sequences.

6.3 Multiple sequence alignment for comparative modelling

Multiple sequence aligners are designed to perform well on difficult input. For example, the common BALiBASE benchmark includes reference sets of alignments containing long insertions or terminal extensions. However, for target-template alignment robustness to input is not necessary, as the user can freely include and exclude sequences from the alignment in order to

maximise accuracy. The question then is how to select homologs for a given program.

One lesson learned from developing MP-T is that small changes in homolog selection can drastically alter alignment accuracy. The homolog selection procedure used in MP-T has a great deal of room for improvement, and it is likely that large increases in accuracy can be obtained this way. For example, in some cases the procedure in Chapter 4 is too strict in its length cut-offs, leading to the use of few sequences in the alignment. In cases where fewer than 50 sequences are selected for alignment, the selection script can be modified to extract only the portions of a candidate sequence that are deemed homologous by PSI-BLAST (Henry Wilman, personal communication). This allows more homologs to pass the length cut-off, and results in a sizeable improvement in alignment accuracy (Table 6.1). It should be noted that this improvement is not the result of great efforts to refine homolog selection – the threshold of 50 sequences is an arbitrary choice, and some alignments are still made with too few sequences.

Table 6.1: Performance of MP-T and MSAProbs when the homolog selection procedure is changed from that in Chapter 4.

	TM domain		Full alignment	
Method	F_M	F_D	F_M	F_D
MP-T (default)	66.3	65.8	69.6	70.1
MP-T (more homologs)	**66.6**	**66.4**	**69.9**	**70.5**
MSAProbs (default)	62.5	62.1	68.2	69.0
MSAProbs (more homologs)	62.5	62.3	67.8	68.7

It is likely that the optimal homolog selection procedure varies between programs – the above procedure does not lead to increased accuracy when applied to MSAprobs (Table 6.1). Other methods tested in this thesis such as ClustalΩ, MAFFT, and MUSCLE could become more competitive with profile-profile aligners if they also had homolog selection optimised for them.

6.4 Future improvements to MP-T

As noted above, the greatest improvements to MP-T will result from refinements to the way that homologs are selected. Nevertheless, this section considers possible algorithmic developments.

Improvements to MP-T suggested in Chapter 4 included the incorporation of sequence weighting, use of a modern consistency criterion (based on conditional probabilities calculated by the forward-backward algorithm), and an iterative refinement step. Sequence weighting is perhaps easiest to implement, but is unlikely to lead to large increases in accuracy as overly similar sequences are already removed during homolog selection. Likewise the high F_M scores of MP-T may depend on the use of the T-Coffee consistency criterion, so changes here have uncertain benefit.

A promising avenue for improvement lies in incorporating more information from experimental PDB files into the alignment process. For example, at present only the structured residues in a PDB file are used in alignment, as only these can be annotated by iMembrane and JOY, and only these can be used to build a comparative model. However alignment may be improved by the incorporation of residues that are present in the native protein but absent in the experimental structure.

Direct use could also be made of the coordinate data in PDB files. In some cases MP-T produces biologically unrealistic alignments that require a single amino acid in the target to span many angstroms in the template. PDB coordinate data would allow these to be detected, and gaps to be repositioned in order to suggest a more plausible alignment.

MP-T already allows multiple structural annotations to be included in its alignments. However, it could follow the example of PROMALS3D and T-Coffee in allowing PDB structures to be aligned by a structure alignment program such as TM-align, and assigning this alignment high confidence in the resulting consistency calculation. In this way, sequence alignment would be informed by more accurate structure alignment.

Finally, an early stage in the MP-T algorithm is the transfer of structural annotation from

the template to all other sequences via pairwise alignment. At this stage, PSI-BLAST searches have already been made to find sequences homologous to the target and template. These PSI-BLAST results are the input required for PROMALS, so potentially the transfer of annotation between the target and template could be made with a PROMALS alignment rather than a less accurate MP-T pairwise alignment. More accurate annotation transfer should improve alignment.

6.5 New applications for MP-T

Although MP-T was developed to align membrane proteins, the algorithm is easily adapted to any set of environment annotations and ESSTs. Preliminary work has been carried out on aligning soluble proteins using environments that are a combination of accessible surface area and either secondary structure or a structural alphabet. At the time of writing homolog selection cut-offs are the primary stumbling block, and no attempt has been made at incorporating environment specific gap penalties. Nevertheless, results are promising, with MP-T retaining a high F_M value, although F_D is several percentage points behind PROMALS.

6.6 A future for FORECAST

The FORECAST fold-recognition technique serves to prove that structural information can be extracted from even low-scoring and noisy correlated substitutions. At present it is complementary to, but not competitive with the best conventional fold-recognition methods. In the near future, fold recognition and alignment methods will make increasing use of correlated substitutions, and FORECAST serves as a yardstick for the gains that can be realised from the incorporation of this information.

In some sense FORECAST has already reached the limits of its utility by being applied to all amenable Pfam families in Pfam 26.0, although as sequencing efforts continue more Pfam families will grow large enough for the method to be applied. Hopefully, as direct information

methods become more powerful and require less sequence information, FORECAST-style algorithms will also improve. However, caution is warranted in view of the way by which direct information methods are currently assessed: as it is impossible to validate a correlation, but it is easy to validate a contact, methods are deemed to perform better if a greater fraction of their higher-scoring correlations are contacts. This may cause problems if higher-scoring correlations are detected at the expense of lower-scoring correlations.

6.7 Final remarks

In this thesis, two of the challenges facing comparative modelling at low sequence identities have been investigated. First, by investigating the evolutionary pressures acting at different local environments of membrane proteins, an alignment method, MP-T, was developed that improved membrane protein modelling accuracy. This method can be used via the web, and is also incorporated in the web-based Memoir membrane protein modelling pipeline. Second, investigation of signals of correlated substitution showed how they might be incorporated in fold recognition.

References

Alexandrov, N. and Shindyalov, I. (2003). PDP: protein domain parser. *Bioinformatics*, **19**(3), 429–430. 7

Almén, M. S., Nordström, K. J. V., Fredriksson, R., and Schiöth, H. B. (2009). Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC biology*, **7**(1), 50. 33

Altschuh, D., Vernet, T., Berti, P., Moras, D., and Nagai, K. (1988). Coordinated amino acid changes in homologous protein families. *"Protein Engineering, Design and Selection"*, **2**(3), 193–199. 96

Altschul, S. (1990). Basic Local Alignment Search Tool. *Journal of Molecular Biology*, **215**(3), 403–410. 19, 20

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**(17), 3389–3402. 20, 50, 77

Altschul, S. F., Gertz, E. M., Agarwala, R., Schäffer, A. A., and Yu, Y.-K. (2009). PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic acids research*, **37**(3), 815–24. 21

Arnold, K., Bordoli, L., Kopp, J., and Schwede, T. (2006). The SWISS-MODEL workspace: a web-based environment for protein structure homology modelling. *Bioinformatics (Oxford, England)*, **22**(2), 195–201. 30

Bahr, A., Thompson, J. D., Thierry, J. C., and Poch, O. (2001). BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic acids research*, **29**(1), 323–6. 71

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 Research*, **28**(1), 235–242. 8, 50

Boucher, Y., Kamekura, M., and Doolittle, W. F. (2004). Origins and evolution of isoprenoid lipid biosynthesis in archaea. *Molecular microbiology*, **52**(2), 515–27. 40

Bretscher, M. and Munro, S. (1993). Cholesterol and the Golgi apparatus. *Science*, **261**(5126), 1280–1281. 40

Bretscher, M. S. (1972). Asymmetrical Lipid Bilayer Structure for Biological Membranes. *Nature*, **236**(61), 11–12. 40

Brown, C. A. and Brown, K. S. (2010). Validation of coevolving residue algorithms via pipeline sensitivity analysis: ELSC and OMES and ZNMI, oh my! *PloS one*, **5**(6), e10779. 99

Browne, K. A., Blink, E., Sutton, V. R., Froelich, C. J., Jans, D. A., and Trapani, J. A. (1999). Cytosolic Delivery of Granzyme B by Bacterial Toxins: Evidence that Endosomal Disruption, in Addition to Transmembrane Pore Formation, Is an Important Function of Perforin. *Mol. Cell. Biol.*, **19**(12), 8604–8615. 38

Bryson, K., McGuffin, L. J., Marsden, R. L., Ward, J. J., Sodhi, J. S., and Jones, D. T. (2005). Protein structure prediction servers at University College London. *Nucleic acids research*, **33**(Web Server issue), W36–8. 13

Chakrabarti, S. and Panchenko, A. R. (2010). Structural and functional roles of coevolved sites in proteins. *PloS one*, **5**(1), e8591. 97

Chandonia, J.-M., Hon, G., Walker, N. S., Lo Conte, L., Koehl, P., Levitt, M., and Brenner, S. E. (2004). The ASTRAL Compendium in 2004. *Nucleic acids research*, **32**(Database issue), D189–92. 114

Chandran, V., Fronzes, R., Duquerroy, S., Cronin, N., Navaza, J., and Waksman, G. (2009). Structure of the outer membrane complex of a type IV secretion system. *Nature*, **462**(7276), 1011–5. 38

Chang, J.-M., Di Tommaso, P., Taly, J.-F., and Notredame, C. (2012). Accurate multiple sequence alignment of transmembrane proteins with PSI-Coffee. *BMC Bioinformatics*, **13**(Suppl 4), S1. 71

Choi, S. S., Li, W., and Lahn, B. T. (2005). Robust signals of coevolution of interacting residues in mammalian proteomes identified by phylogeny-aided structural analysis. *Nature genetics*, **37**(12), 1367–71. 97

Choi, Y. and Deane, C. M. (2010). FREAD revisited: Accurate loop structure prediction using a database search algorithm. *Proteins*, **78**(6), 1431–1440. 30, 89

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978). A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, **5**(Suppl 3), 345–352. 47

Di Lena, P., Fariselli, P., Margara, L., Vassura, M., and Casadio, R. (2010). Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, **26**(18), 2250–2258. 107, 108, 117

Do, C. B., Mahabhashyam, M. S. P., Brudno, M., and Batzoglou, S. (2005). ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, **15**(2), 330–40. 23, 24

Dong, C., Beis, K., Nesper, J., Brunkan-Lamontagne, A. L., Clarke, B. R., Whitfield, C., and Naismith, J. H. (2006). Wza the translocon for E. coli capsular polysaccharides defines a new class of membrane protein. *Nature*, **444**(7116), 226–9. 38

Dunn, S. D., Wahl, L. M., and Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics (Oxford, England)*, **24**(3), 333–40. 99

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids.*, volume 17. CUP. 18

Eddy, S. R. (2011). Accelerated Profile HMM Searches. *PLoS computational biology*, **7**(10), e1002195. 20

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, **32**(5), 1792–1797. 23, 46, 77, 79

Edgar, R. C. (2009). Optimizing substitution matrix choice and gap parameters for sequence alignment. *BMC bioinformatics*, **10**(1), 396. 48

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**(19), 2460–1. 77

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, **87**(1), 012707. 96

Elofsson, A. and von Heijne, G. (2007). Membrane protein structure: prediction versus reality. *Annual review of biochemistry*, **76**, 125–40. 34

Eswar, N., Webb, B., Marti-Renom, M. A., Madhusudhan, M. S., Eramian, D., Shen, M.-Y., Pieper, U., and Sali, A. (2007). Comparative protein structure modeling using MODELLER. *Current protocols in protein science / editorial board, John E. Coligan ... [et al.]*, **Chapter 2**, Unit 2.9. 30

Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., and Zhou, Y. (2012). SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, **33**(3), 259–67. 13

Forrest, L., Tang, C., and Honig, B. (2006). On the Accuracy of Homology Modeling and Sequence Alignment Methods Applied to Membrane Proteins. *Biophysical Journal*, **91**(2), 508–517. 45, 71

Fredriksson, R., Lagerström, M. C., Lundin, L.-G., and Schiöth, H. B. (2003). The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints. *Molecular pharmacology*, **63**(6), 1256–72. 36

Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins*, **23**(4), 566–79. 11

Gascuel, O. (1997). BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.*, **14**(7), 685–695. 83

Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins*, **18**(4), 309–17. 96

Gong, S., Worth, C. L., Bickerton, G. R. J., Lee, S., Tanramluk, D., and Blundell, T. L. (2009). Structural and functional restraints in the evolution of protein families and superfamilies. *Biochemical Society transactions*, **37**(Pt 4), 727–33. 54

Gonzalez, M. W. and Pearson, W. R. (2010). Homologous over-extension: a challenge for iterative similarity searches. *Nucleic acids research*, **38**(7), 2177–89. 20

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, **162**(3), 705–708. 18

Gotoh, O. (1995). A weighting system and aigorithm for aligning many phylogenetically related sequences. *Bioinformatics*, **11**(5), 543–551. 26

Grishin, N. (1995). Estimation of the number of amino acid substitutions per site when the substitution rate varies among sites. *Journal of Molecular Evolution*, **41**(5). 73

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, **89**(22), 10915–10919. 45, 47

Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *Journal of Molecular Evolution*, **20**(2), 175–186. 23

Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *Journal of molecular biology*, **233**(1), 123–38. 16

Hopf, T. A., Colwell, L. J., Sheridan, R., Rost, B., Sander, C., and Marks, D. S. (2012). Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, **149**(7), 1607–21. 96

Horner, D. S., Pirovano, W., and Pesole, G. (2008). Correlated substitution analysis and the prediction of amino acid structural contacts. *Briefings in bioinformatics*, **9**(1), 46–56. 105

Hotelling, H. (1933). Analysis of complex statistical variables into principal components. *Journal of Educational Psychology*, **24**(6), 417–441. 54

Hubbard, T. and Blundell, T. (1987). Comparison of solvent-inaccessible cores of homologous proteins: definitions useful for protein modelling. *"Protein Engineering, Design and Selection"*, **1**(3), 159–171. 11, 50

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD: Visual molecular dynamics. *Journal of Molecular Graphics*, **14**(1), 33–38. 39

Jaakola, V.-P., Griffith, M. T., Hanson, M. A., Cherezov, V., Chien, E. Y. T., Lane, J. R., Ijzerman, A. P., and Stevens, R. C. (2008). The 2.6 angstrom crystal structure of a human A2A adenosine receptor bound to an antagonist. *Science (New York, N.Y.)*, **322**(5905), 1211–7. 49

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**(2), 351–67. 30

Jaroszewski, L., Li, Z., Krishna, S. S., Bakolitsa, C., Wooley, J., Deacon, A. M., Wilson, I. A., and Godzik, A. (2009). Exploration of uncharted regions of the protein universe. *PLoS biology*, **7**(9), e1000205. 96

Jeong, C.-S. and Kim, D. (2012). Reliable and robust detection of coevolving protein residues. *Protein engineering, design & selection : PEDS*, **25**(11), 705–13. 99

Jones, D., Taylor, W., and Thornton, J. (1994). A mutation data matrix for transmembrane proteins. *FEBS Letters*, **339**(3), 269–275. 45

Jones, D. T. (1999a). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, **292**(2), 195–202. 13, 80

Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992). A new approach to protein fold recognition. *Nature*, **358**(6381), 86–9. 26

Jones, D. T., Buchan, D. W. A., Cozzetto, D., and Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics (Oxford, England)*, **28**(2), 184–90. 96, 100, 114

Jones, M. (1999b). Polymeric micelles a new generation of colloidal drug carriers. *European Journal of Pharmaceutics and Biopharmaceutics*, **48**(2), 101–111. 105

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, **32**(5), 922–923. 17

Kabsch, W. (1978). A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, **34**(5), 827–828. 17

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**(12), 2577–637. 11

Karlin, S. and Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, **87**(6), 2264–8. 22

Katoh, K. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, **30**(14), 3059–3066. 24

Katoh, K. and Toh, H. (2008). Recent developments in the MAFFT multiple sequence alignment program. *Briefings in bioinformatics*, **9**(4), 286–98. 24, 71, 72

Katoh, K., Kuma, K.-i., Toh, H., and Miyata, T. (2005). MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, **33**(2), 511–8. 24

Kelm, S., Shi, J., and Deane, C. M. (2009). iMembrane: homology-based membrane-insertion of proteins. *Bioinformatics*, **25**(8), 1086–1088. 14, 46

Kelm, S., Shi, J., and Deane, C. M. (2010). MEDELLER: homology-based coordinate generation for membrane proteins. *Bioinformatics*, **26**(22), 2833–2840. 30, 54, 81, 89

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*. 30

Khafizov, K., Staritzbichler, R., Stamm, M., and Forrest, L. R. (2010). A study of the evolution of inverted-topology repeats from LeuT-fold transporters using AlignMe. *Biochemistry*, **49**(50), 10702–13. 71

Korber, B. (1993). Covariation of Mutations in the V3 Loop of Human Immunodeficiency Virus Type 1 Envelope Protein: An Information Theoretic Analysis. *Proceedings of the National Academy of Sciences*, **90**(15), 7176–7180. 96, 99

Kowarsch, A., Fuchs, A., Frishman, D., and Pagel, P. (2010). Correlated mutations: a hallmark of phenotypic amino acid substitutions. *PLoS computational biology*, **6**(9), 13. 97

Krivov, G. G., Shapovalov, M. V., and Dunbrack, R. L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. *Proteins*, **77**(4), 778–95. 30

Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of molecular biology*, **305**(3), 567–80. 123

Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., and Baker, D. (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science (New York, N.Y.)*, **302**(5649), 1364–8. 7

Lapedes, A. S., Giraud, B. G., Liu, L., and Stormo, G. D. (1999). Correlated mutations in models of protein sequences: phylogenetic and structural effects. *IMS Lecture Notes - Monograph Series*, **33**, 236–256. 96

Law, R. H. P., Lukoyanova, N., Voskoboinik, I., Caradoc-Davies, T. T., Baran, K., Dunstone, M. A., D'Angelo, M. E., Orlova, E. V., Coulibaly, F., Verschoor, S., Browne, K. A., Ciccone, A., Kuiper, M. J., Bird, P. I., Trapani, J. A., Saibil, H. R., and Whisstock, J. C. (2010). The structural basis for membrane binding and pore formation by lymphocyte perforin. *Nature*, **468**(7322), 447–51. 38

Lee, B. and Richards, F. (1971). The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, **55**(3), 379–IN4. 11

Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**(13), 1658–1659. 50

Lieberman, R. L. and Rosenzweig, A. C. (2005). Crystal structure of a membrane-bound metalloenzyme that catalyses the biological oxidation of methane. *Nature*, **434**(7030), 177–82. 57

Little, D. Y. and Chen, L. (2009). Identification of coevolving residues and coevolution potentials emphasizing structure, bond formation and catalytic coordination in protein evolution. *PloS one*, **4**(3), e4762. 97, 99

Liu, Y., Schmidt, B., and Maskell, D. L. (2010). MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics*, **26**(16), 1958–64. 23, 24, 71, 77

Lobley, A., Sadowski, M. I., and Jones, D. T. (2009). pGenTHREADER and pDomTHREADER: new methods for improved protein fold recognition and superfamily discrimination. *Bioinformatics*, **25**(14), 1761–7. 71

Lomize, A. L., Pogozheva, I. D., and Mosberg, H. I. (2011). Anisotropic solvent model of the lipid bilayer. 2. Energetics of insertion of small molecules, peptides, and proteins in membranes. *Journal of chemical information and modeling*, **51**(4), 930–46. 15, 33

Long, S. B., Tao, X., Campbell, E. B., and MacKinnon, R. (2007). Atomic structure of a voltage-dependent K+ channel in a lipid membrane-like environment. *Nature*, **450**(7168), 376–82. 9, 39

Lugtenberg, B. and Vanalphen, L. (1983). Molecular architecture and functioning of the outer membrane of Escherichia coli and other gram-negative bacteria. *Biochimica et Biophysica Acta (BBA) - Reviews on Biomembranes*, **737**(1), 51–115. 40

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. *PloS one*, **6**(12), e28766. 96, 117

Martin, L. C., Gloor, G. B., Dunn, S. D., and Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics (Oxford, England)*, **21**(22), 4116–24. 99

McGuffin, L. J. and Jones, D. T. (2002). Targeting novel folds for structural genomics. *Proteins*, **48**(1), 44–52. 112

McGuffin, L. J., Bryson, K., and Jones, D. T. (2001). What are the baselines for protein fold recognition? *Bioinformatics*, **17**(1), 63–72. 112

Miao, Z., Cao, Y., and Jiang, T. (2011). RASP: rapid modeling of protein side chain conformations. *Bioinformatics (Oxford, England)*, **27**(22), 3117–22. 30

Michener, C. D. and Sokal, R. R. (1957). A quantitative approach to a problem in classification. *Evolution*, **11**(2), 130–162. 23

Mirabello, C. and Pollastri, G. (2013). Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics (Oxford, England)*, **29**(16), 2056–8. 13

Mizuguchi, K., Deane, C., Blundell, T., and Overington, J. (1998a). HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci*, **7**(11), 2469–2471. 50

Mizuguchi, K., Deane, C., Blundell, T., Johnson, M., and Overington, J. (1998b). JOY: protein sequence-structure representation and analysis. *Bioinformatics*, **14**(7), 617–623. 11, 80

Mizuguchi, K., Sele, M., and Cubellis, M. V. (2007). Environment specific substitution tables for thermophilic proteins. *BMC bioinformatics*, **8 Suppl 1**, S15. 45, 55

Mokrab, Y. and Mizuguchi, K. (2005). Amino-Acid Substitutions In Membrane Proteins: Applications To Homology Recognition And Comparative Modelling. *BMC Bioinformatics*, **6**(Suppl 3), S9. 45

Mokrab, Y., Stevens, T. J., and Mizuguchi, K. (2010). A structural dissection of amino acid substitutions in helical transmembrane proteins. *Proteins*, **78**(14), 2895–2907. 45

Monastyrskyy, B., D'Andrea, D., Fidelis, K., Tramontano, A., and Kryshtafovych, A. (2013). Evaluation of residue-residue contact prediction in CASP10. *Proteins*. 127

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America*, **108**(49), E1293–301. 96, 100

Müller, T., Rahmann, S., and Rehmsmeier, M. (2001). Non-symmetric score matrices and the detection of homologous transmembrane proteins. *Bioinformatics*, **17**(suppl 1), S182–S189. 45

Muller, T., Spang, R., and Vingron, M. (2002). Estimating Amino Acid Substitution Models: A Comparison of Dayhoff's Estimator, the Resolvent Approach and a Maximum Likelihood Method. *Mol. Biol. Evol.*, **19**(1), 8–13. 47

Murzin, A., Brenner, S., Hubbard, T., and Chothia, C. (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, **247**(4), 536–540. 7, 97, 101

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. *Journal of Molecular Biology*, **48**(3), 443–453. 18

Neher, E. (1994). How Frequent are Correlated Changes in Families of Protein Sequences? *Proceedings of the National Academy of Sciences*, **91**(1), 98–102. 96

Ng, P. C., Henikoff, J. G., and Henikoff, S. (2000). PHAT: a transmembrane-specific substitution matrix. *Bioinformatics*, **16**(9), 760–766. 45

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, **302**(1), 205–17. 23, 71, 74, 77

Nugent, T. and Jones, D. T. (2012). Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. *Proceedings of the National Academy of Sciences of the United States of America*, **109**(24), E1540–7. 96, 114

Oberai, A., Ihm, Y., Kim, S., and Bowie, J. U. (2006). A limited universe of membrane protein families and folds. *Protein science : a publication of the Protein Society*, **15**(7), 1723–34. 36

Overington, J., Donnelly, D., Johnson, M. S., Sali, A., and Blundell, T. L. (1992). Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds. *Protein science : a publication of the Protein Society*, **1**(2), 216–26. 45

Overington, J. P., Al-Lazikani, B., and Hopkins, A. L. (2006). How many drug targets are there? *Nature reviews. Drug discovery*, **5**(12), 993–6. 34

Pandit, S. B. and Skolnick, J. (2008). Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score. *BMC bioinformatics*, **9**(1), 531. 16, 105

Park, Y., Sheetlin, S., Ma, N., Madden, T. L., and Spouge, J. L. (2012). New finite-size correction for local alignment score distributions. *BMC research notes*, **5**(1), 286. 22

Pearson, W. R. (1988). Improved Tools for Biological Sequence Comparison. *Proceedings of the National Academy of Sciences*, **85**(8), 2444–2448. 19

Pei, J. and Grishin, N. V. (2007). PROMALS: towards accurate multiple sequence alignments of distantly related proteins. *Bioinformatics*, **23**(7), 802–8. 27, 71, 72, 77, 79

Pei, J., Kim, B.-H., and Grishin, N. V. (2008). PROMALS3D: a tool for multiple protein sequence and structure alignments. *Nucleic acids research*, **36**(7), 2295–300. 27

Peng, J. and Xu, J. (2010). Low-homology protein threading. *Bioinformatics*, **26**(12), i294–300. 27

Peng, J. and Xu, J. (2011). RaptorX: Exploiting structure information for protein alignment by statistical inference. *Proteins: Structure, Function, and Bioinformatics*, **79 Suppl 1**, n/a–n/a. 27

Pirovano, W., Feenstra, K. A., and Heringa, J. (2008). PRALINE: a strategy for improved multiple alignment of transmembrane proteins. *Bioinformatics*, **24**(4), 492–497. 45, 68, 71

Plyusnin, I. and Holm, L. (2012). Comprehensive comparison of graph based multiple protein sequence alignment strategies. *BMC Bioinformatics*, **13**(1), 64. 83

Punta, M., Coggill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E. L. L., Eddy, S. R., Bateman, A., and Finn, R. D. (2012). The Pfam protein families database. *Nucleic acids research*, **40**(Database issue), D290–301. 7, 98, 102

Rees, D. C., Johnson, E., and Lewinson, O. (2009). ABC transporters: the power to change. *Nature reviews. Molecular cell biology*, **10**(3), 218–27. 36

Remaut, H., Tang, C., Henderson, N. S., Pinkner, J. S., Wang, T., Hultgren, S. J., Thanassi, D. G., Waksman, G., and Li, H. (2008). Fiber formation across the bacterial outer membrane by the chaperone/usher pathway. *Cell*, **133**(4), 640–52. 35

Remmert, M., Linke, D., Lupas, A. N., and Söding, J. (2009). HHomp–prediction and classification of outer membrane proteins. *Nucleic acids research*, **37**(Web Server issue), W446–51. 34

Remmert, M., Biegert, A., Linke, D., Lupas, A. N., and Söding, J. (2010). Evolution of outer membrane beta-barrels from an ancestral beta beta hairpin. *Molecular biology and evolution*, **27**(6), 1348–58. 35

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods*, **9**(2), 173–5. 20, 80

Rosenbaum, D. M., Rasmussen, S. r. G. F., and Kobilka, B. K. (2009). The structure and function of G-protein-coupled receptors. *Nature*, **459**(7245), 356–63. 36

Rost, B. and Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, **232**(2), 584–99. 13

Sadowski, M. I. and Taylor, W. R. (2012). Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics (Oxford, England)*, **28**(9), 1209–15. 105

Sadowski, M. I., Maksimiak, K., and Taylor, W. R. (2011). Direct correlation analysis improves fold recognition. *Computational biology and chemistry*, **35**(5), 323–32. 96

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, **4**(4), 406–425. 23

Sali, A. (1993). Comparative Protein Modelling by Satisfaction of Spatial Restraints. *Journal of Molecular Biology*, **234**(3), 779–815. 30

Sauder, J. M., Arthur, J. W., and Dunbrack, R. L. (2000). Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, **40**(1), 6–22. 81

Schulz, G. E. (2005). The structures of general porins. In R. Benz, editor, *Bacterial and Eukaryotic Porins: Structure, Function, Mechanism*, book part (with own title) 2. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, FRG. 34

Scott, K. A., Bond, P. J., Ivetac, A., Chetwynd, A. P., Khalid, S., and Sansom, M. S. (2008). Coarse-Grained MD Simulations of Membrane Protein-Bilayer Self-Assembly. *Structure*, **16**(4), 621–630. 14, 49

Shi, J., Blundell, T., and Mizuguchi, K. (2001). FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of molecular biology*, **310**(1), 243–257. 27, 45, 46, 51, 62, 70

Shindyalov, I., Kolchanov, N., and Sander, C. (1994). Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *"Protein Engineering, Design and Selection"*, **7**(3), 349–358. 96

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular systems biology*, **7**, 539. 77

Sillitoe, I., Cuff, A. L., Dessailly, B. H., Dawson, N. L., Furnham, N., Lee, D., Lees, J. G., Lewis, T. E., Studer, R. A., Rentzsch, R., Yeats, C., Thornton, J. M., and Orengo, C. A. (2013). New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic acids research*, **41**(Database issue), D490–8. 98

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**(1), 195–197. 18

Söding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**(7), 951–60. 27, 71, 72, 77, 79, 112

Studier, J. A. and Keppler, K. J. (1988). A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular biology and evolution*, **5**(6), 729–31. 23

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007). UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, **23**(10), 1282–8. 8, 77

Sukowska, J. I., Morcos, F., Weigt, M., Hwa, T., and Onuchic, J. N. (2012). Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences of the United States of America*, **109**(26), 10340–5. 96

Tastan, O., Klein-Seetharaman, J., and Meirovitch, H. (2009). The effect of loops on the structural organization of alpha-helical membrane proteins. *Biophysical journal*, **96**(6), 2299–312. 58

Taylor, W. R. (1999). Protein structure comparison using iterated double dynamic programming. *Protein science : a publication of the Protein Society*, **8**(3), 654–65. 16

Taylor, W. R. and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. *"Protein Engineering, Design and Selection"*, **7**(3), 341–348. 96

Taylor, W. R., Chelliah, V., Hollup, S. M., MacDonald, J. T., and Jonassen, I. (2009). Probing the "dark matter" of protein fold space. *Structure (London, England : 1993)*, **17**(9), 1244–52. 7

Taylor, W. R., Jones, D. T., and Sadowski, M. I. (2012). Protein topology from predicted residue contacts. *Protein science : a publication of the Protein Society*, **21**(2), 299–305. 96

Thompson, J. D., Linard, B., Lecompte, O., and Poch, O. (2011). A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS one*, **6**(3), e18093. 81

Tung, C.-H. and Yang, J.-M. (2007). fastSCOP: a fast web server for recognizing protein structural domains and SCOP superfamilies. *Nucleic acids research*, **35**(Web Server issue), W438–43. 112, 122

Tusnády, G., Dosztányi, Z., and Simon, I. (2005). PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. *Nucleic Acids Res*, **33**(Database issue). 50

Tusnády, G. E., Dosztányi, Z., and Simon, I. (2004). Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics (Oxford, England)*, **20**(17), 2964–72. 15

Vieira-Pires, R. S. a. and Morais-Cabral, J. a. H. (2010). 3(10) helices in channels and other membrane proteins. *The Journal of general physiology*, **136**(6), 585–92. 7

Vogt, G., Etzold, T., and Argos, P. (1995). An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *Journal of molecular biology*, **249**(4), 816–31. 2

von Ohsen, N., Zimmer, R., Gascuel, O., and Moret, B. (2001). *Algorithms in Bioinformatics*, volume 2149 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg. 47

Wheeler, T. J. and Kececioglu, J. D. (2007). Multiple alignment by aligning alignments. *Bioinformatics*, **23**(13), i559–68. 83

White, S. H. (2009). Biophysical dissection of membrane proteins. *Nature*, **459**(7245), 344–6. 36

White, S. H. (2013). Membrane proteins of known structure http://blanco.biomol.uci.edu/mpstruc/. 8, 34, 36, 74

Xu, J. and Zhang, Y. (2010). How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics*, **26**(7), 889–95. 17

Yang, Y., Faraggi, E., Zhao, H., and Zhou, Y. (2011). Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, **27**(15), 2076–82. 27, 70, 71, 118

Ye, Y. and Godzik, A. (2003). Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19**(Suppl 2), ii246–ii255. 16

Yildiz, O., Vinothkumar, K. R., Goswami, P., and Kühlbrandt, W. (2006). Structure of the monomeric outer-membrane porin OmpG in the open and closed conformation. *The EMBO journal*, **25**(15), 3702–13. 35

Zemla, A., Venclovas, e., Moult, J., and Fidelis, K. (1999). Processing and analysis of CASP3 protein structure predictions. *Proteins: Structure, Function, and Genetics*, **37**(S3), 22–29. 14

Zemla, A., Venclovas, Moult, J., and Fidelis, K. (2001). Processing and evaluation of predictions in CASP4. *Proteins*, **Suppl 5**, 13–21. 16, 72

Zeth, K. and Thein, M. (2010). Porins in prokaryotes and eukaryotes: common themes and variations. *Biochem. J.*, **431**(1), 13–22. 35

Zhang, H., Zhang, T., Chen, K., Kedarisetti, K. D., Mizianty, M. J., Bao, Q., Stach, W., and Kurgan, L. (2011). Critical assessment of high-throughput standalone methods for secondary structure prediction. *Briefings in bioinformatics*, **12**(6), 672–88. 14

Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**(4), 702–10. 15, 81

Zhang, Y. and Skolnick, J. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, **33**(7), 2302–2309. 16, 54, 74, 112, 118