

Biologically Inspired *De Novo* Protein Structure Prediction.



Saulo Henrique Pires de Oliveira

Department of Statistics

University of Oxford

This dissertation is submitted for the degree of
Doctor of Philosophy

Pembroke College

October 2015

This thesis is dedicated to my family. None of this would have been possible without your guidance, patience, love, and support. Thank you.

Acknowledgements

First and foremost, I would like to acknowledge my supervisor, Professor Charlotte Deane. I cannot put into words how crucial her support, patience, guidance, and friendship have been during the past four years. I have learned so much from Professor Deane, both academically and personally. I am truly thankful for her utmost commitment to being a great supervisor.

I would like to acknowledge my industrial supervisor Dr. Jiye Shi, for all the insights, ideological contributions, and guidance throughout my doctorate.

I would also like to thank my colleagues from the Oxford Protein Informatics Group (OPIG) for the many interesting discussions and their contributions to my work and personal life. In particular, I would like to thank Anthony Bradley for all the insightful conversations and Nicholas Pearce for his friendship and support.

I want to acknowledge my family and friends whose support has been fundamental throughout every step of this journey. I am so grateful for all their love, support, and care. I need to thank my parents for believing in me and convincing me to come to Oxford, even when the prospectus of finding funding was grim. I would also like to thank my wife, Mo, for being the bastion that kept me going when the going got rough.

A big *thank you* is also in order for all the staff members from the Statistics Department, Pembroke College, and the Doctoral Training Centre. Their contribution throughout my doctorate has been very valuable. Special thanks to Beverley Lane for for all her support, administrative and otherwise.

Finally, I would like to acknowledge my funding bodies, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Ciência Sem Fronteiras (CsF), UCB Pharma and the Engineering and Physical Sciences Research Council (EPSRC).

Abstract: Fragment-based approaches are the current standard for *de novo* protein structure prediction. These approaches rely on accurate and reliable fragment libraries to generate good structural models. We demonstrate that fragments presenting different predominant predicted secondary structures should be treated differently during fragment library generation. Using this information, we developed Flib and shown that it generates fragment libraries with higher precision and coverage than two other methods.

We explored co-evolution to identify pairs of residues that are in contact, which were then used to improve model generation. We performed a comparative analysis of nine methods in terms of their precision and their usefulness to *de novo* structure prediction. Our results show that metaPSICOV stage 2 produces the most accurate predictions and that metaPSICOV stage 1 generates the best modelling results. In general, contact predictors are good at identifying contacts between β -strands and bad at identifying contacts between α -helices. We also show that the ratio of satisfied predicted contacts can be used to assess whether correct models were generated for a given target.

We also investigated whether the biological process of cotranslational protein folding, the notion that proteins fold as they are being synthesized, can be used to improve *de novo* protein structure prediction. Our tool for this investigation is SAINT2. SAINT2 differs from conventional fragment-assembly approaches as it is able to perform predictions sequentially from N to C-terminus, starting with a small peptide that is extended as the simulation progresses (SAINT2 Cotranslational). SAINT2 is also able to generate decoys in a standard non-sequential fashion (SAINT2 *In Vitro*). We compared SAINT2 Cotranslational to SAINT2 *In Vitro* and shown that SAINT2 Cotranslational generally produces better answers, generating an individual decoy between 1.5 to 2.5 times faster than SAINT2 *In Vitro*. Our results suggest that biologically inspired structure prediction can improve search heuristics and final model quality.

Table of contents

List of figures	xv
List of tables	xxi
1 Introduction	1
1.1 Life, as we know it	1
1.2 Proteins	3
1.2.1 Amino Acids	4
1.2.2 Peptide Bond	6
1.2.3 Hydrogen Bonds	6
1.2.4 Torsion Angles and Ramachandran Plot	8
1.2.5 Residue Conformation and Rotamers	10
1.3 Protein Sequence (Primary Structure)	12
1.3.1 Sequence Alignment	13
1.4 Secondary Structure	18
1.4.1 α -helices	20
1.4.2 β -strands and β -sheets	21
1.4.3 Relationship between Secondary Structure and the Ramachandran Plot	22
1.5 Protein Structure (Tertiary Structure)	23
1.5.1 The Protein Data Bank (PDB)	26

1.5.2	Protein Structure Determination	26
1.5.3	Structure Alignment (Superposition)	30
1.6	Quaternary Structure	32
1.7	Protein Classification	32
1.7.1	Globular Proteins and Membrane Proteins	32
1.7.2	Pfam	33
1.7.3	SCOP	34
1.8	Protein Structure Prediction	36
1.8.1	A Historical Perspective of Protein Structure Prediction	37
1.8.2	Template-Based Modelling	40
1.8.3	Template-free Modelling (<i>de novo</i> Protein Structure Prediction)	42
1.8.4	Predicted Secondary Structure	43
1.8.5	Predicted Torsion Angles	45
1.8.6	Heuristics	45
1.8.7	Scoring Function	46
1.8.8	Model Quality Assessment	47
1.8.9	Model Refinement	48
1.9	Translation and Protein Synthesis	49
1.9.1	The Genetic Code: codons, mRNA, and tRNA	49
1.9.2	Ribosome and the Machinery for Protein Translation	51
1.10	Biologically Inspired <i>de novo</i> Protein Structure Prediction	53
2	Building a better fragment library for <i>de novo</i> protein structure prediction	55
2.1	Introduction	55
2.1.1	Building Fragment Libraries	57
2.1.2	Building a Better Fragment Library	60
2.2	Materials and Methods	62

2.2.1	Data Sets	62
2.2.2	Homologue Identification	67
2.2.3	Template Databases	68
2.2.4	Fragment Scores	69
2.2.5	Classifying Fragments According to Secondary Structure	71
2.2.6	Fragment Extraction with Flib	71
2.2.7	Validation	74
2.2.8	Model Generation	76
2.3	Results	76
2.3.1	The Rationale Behind Flib	76
2.3.2	Building Better Template Databases	76
2.3.3	Building a Better Scoring Scheme for Fragments	80
2.3.4	Improving Fragment Extraction	81
2.3.5	Enriching Fragment Libraries	82
2.3.6	Protein Threading Hits Library	83
2.3.7	Secondary Structure Classes Determine Fragment Quality	83
2.3.8	Cross-comparison Between Flib and Other Software	87
2.3.9	Effect of Homologues on Fragment Library Quality	92
2.3.10	Model Generation/Protein Structure Prediction	95
2.4	Discussion	98
3	<i>De novo</i> Protein Structure Prediction Assisted by Co-evolution	101
3.1	Introduction	101
3.2	Methods	108
3.2.1	Contact Definitions	108
3.2.2	Mean Field Direct Coupling Analysis (mfDCA)	108
3.2.3	Protein Sparse Inverse COVariance Analysis Program (PSICOV)	110

3.2.4	Pseudo-likelihood Maximization Direct Coupling Analysis (plmDCA)	111
3.2.5	Consensus Methods	113
3.2.6	Secondary Structure-Specific Methods	115
3.2.7	Number of Predictions Considered	115
3.2.8	Data Set	115
3.2.9	Validation Metrics	116
3.2.10	Model Generation	116
3.2.11	SAINT2: Contact Potential	116
3.3	Results	118
3.3.1	Comparing state-of-the-art contact predictors	118
3.3.2	Determining the Best Contact Predictor to Assist in <i>de novo</i> Protein Structure Prediction	129
3.3.3	Satisfied Predictions Contacts Correlate with Model Quality	133
3.3.4	Assessing the Usefulness of Predicted Contacts to Model Quality Assessment	135
3.3.5	Searching for evolutionary signal in predicted residue-residue contacts	138
3.4	Discussion	140
4	Biologically Inspired <i>de novo</i> Protein Structure Prediction: Methods	145
4.1	Introduction	145
4.2	Methods	148
4.2.1	Sequence-to-Structure Pipeline	148
4.2.2	Training Data Set	151
4.2.3	Validation	151
4.2.4	The Framework of SAINT2	152
4.2.5	Peptide/Protein Representation within SAINT2	152
4.2.6	SAINT2 Run Modes and Initialization	152

4.2.7	Fragment Replacements	154
4.2.8	Mover	156
4.2.9	Score	157
4.2.10	Residue-specific All-atom conditional Probability Discriminatory Function	158
4.2.11	Solvation	159
4.2.12	Lennard-Jones Potential	161
4.2.13	Orientation Potential	162
4.2.14	Predicted Secondary Structure Potential	164
4.2.15	Predicted Torsion Angle Potential	165
4.2.16	Radius of Gyration Potential	165
4.2.17	Predicted Contact Potential	166
4.2.18	Normalization of SAINT2 Score Components and Initial Scoring Weights	168
4.2.19	Optimizing Scoring Weights	169
4.2.20	Strategy	171
4.2.21	Selection and Refinement	171
4.2.22	Performance	171
4.2.23	Critical Assessment of protein Structure Prediction	174
4.3	Results	175
4.3.1	CASP	178
4.4	Discussion	178
5	Biologically Inspired <i>de novo</i> Protein Structure Prediction	183
5.1	Introduction	183
5.2	Methods	188
5.2.1	Data Sets	188

5.2.2	Decoy Generation	190
5.2.3	Validation	190
5.2.4	Signatures of Cotranslational Protein Folding	191
5.3	Results	194
5.3.1	Determining the Number of Decoys Required by SAINT2	194
5.3.2	Impact of Sequentiality on the Quality of Models Generated	199
5.3.3	Improvement Cotranslational is Consistent in a Protein Family	203
5.3.4	Exploring Signatures of Cotranslational Protein Folding	204
5.4	Discussion	206
6	Future Work	209
6.1	Building even better fragment libraries	209
6.2	Exploring co-evolution information to improve model generation	211
6.3	Improving SAINT2	212
6.4	Exploring other cotranslational features in <i>de novo</i> protein structure prediction	213
	References	215

List of figures

1.1	Representation of the structure of an amino acid.	4
1.2	The 20 canonical amino acids.	5
1.3	Representation of a peptide bond reaction.	7
1.4	Definition of a dihedral angle.	8
1.5	Schematic representation of torsion angles.	9
1.6	Representation of backbone atoms and bonds used to define torsion angles.	10
1.7	A Ramachandran plot.	11
1.8	Representation of χ dihedral angles.	12
1.9	Schematic representation of an α -helix.	20
1.10	Schematic representation of an β -sheet.	21
1.11	Difference between parallel and anti-parallel β -sheets.	22
1.12	Cartoon representation of a protein structure.	24
1.13	Representation of a funnel-shaped conformational energy landscape.	25
1.14	A diffraction pattern from an X-Ray crystallography experiment.	27
1.15	Photo of lysozyme crystals.	28
1.16	Electron density maps for protein structures with two different resolutions.	29
1.17	Cartoon representation of two superimposed protein structures.	30
1.18	Schematic representation of the hierarchical organization of SCOP.	35

1.19	Growth of the number of known protein sequences and structures available on the public domain.	37
1.20	Schematic representation of translation.	50
1.21	Codons and the amino acids they encode for.	51
2.1	A structural fragment.	56
2.2	Representation of a fragment library.	57
2.3	Determining Flib's score cutoffs.	73
2.4	Difference between good and bad fragments.	75
2.5	The rationale behind Flib.	77
2.6	Effect of different template databases on fragment library quality.	78
2.7	Effect of SCOP-specific template databases on fragment library quality.	79
2.8	Correlation between fragment RMSD and three different scores.	80
2.9	Comparison between Flib's random extraction and exhaustive extraction methods.	82
2.10	Effect of protein threading hits on fragment library quality.	84
2.11	Analysis of the impact of fragments extracted from protein threading hits on the precision within each Secondary Structure fragment class.	85
2.12	Relationship between secondary structure class (SS-Class) and fragment quality.	86
2.13	Comparison between HHFrag, NNMake and Flib for the PDB-representative validation set.	88
2.14	Precision of fragment libraries generated using NNMake, HHFrag, and Flib separated by SS Class for the PDB-representative validation set.	89
2.15	Comparison between HHFrag, NNMake and Flib for the CASP validation set.	90
2.16	Precision of fragment libraries generated using NNMake, HHFrag, and Flib separated by SS Class for the CASP validation set.	91

2.17	Distribution of RMSDs to the native structure of all fragments as generated by NNMake and Flib.	93
2.18	Effect of Homologues on fragment library quality.	94
2.19	TM-Score of the best decoy as generated by Flib+SAINT2 and by NN-Make+SAINT2.	96
2.20	Distribution of RMSDs to the native structure of the best fragment per target position as generated by each of NNMake, HHFrag, and Flib.	97
3.1	Schematic representation of contacts inferred from coevolution.	102
3.2	Average precision of nine contact predictors for our PDB-Representative Validation data set.	118
3.3	Precision of the top $L/10$, $L/5$, $L/2$ and L predicted contacts as output by nine predictors for our PDB-Representative Validation data set.	119
3.4	Precision and proportion of short-range and long-range predicted contacts output by nine different contact predictors.	120
3.5	Estimate of the number of sequences required to produce precise contact predictions.	122
3.6	Correlation between precision of predicted contacts and number of MSA sequences or protein length.	122
3.7	Precision of contact predictions by secondary structure type.	124
3.8	Proportions of predicted contacts for each secondary structure type.	125
3.9	Precision of top L predicted contacts according to target's SCOP class.	126
3.10	Precision of top $L/10$ predicted backbone hydrogen bond contacts as output by nine contact predictors.	127
3.11	Spread of the top L correct predicted contacts as output by nine contact predictors for 41 proteins in our PDB-Representative Validation data set.	128

3.12	TM-Score Best obtained using contact predictions of seven different predictors as distance constraints in SAINT2.	130
3.13	Difference in TM-Score Best of the best possible model and the best model produced using each of seven contact predictors.	131
3.14	TM-Score Best according to target's SCOP class obtained using contact predictions of seven different predictors as distance constraints in SAINT2 .	132
3.15	Relationship between satisfied predicted contacts and model quality.	133
3.16	Relationship between satisfied short-range and long-range predicted contacts and model quality.	134
3.17	Relationship between the spread of satisfied predicted long-range contacts and model quality	135
3.18	Test cases to ascertain effectiveness of long-range contact predictions to assist in model quality assessment.	136
3.19	Proportion of satisfied long-range predicted contacts for correct models compared to remaining models.	137
3.20	TM-Score Top 5 of all decoys against subset of decoys that satisfy at least 29% of predicted long-range contacts.	138
3.21	Comparison of the best model generated by SAINT2 using the top ten and the bottom ten correct contact predictions output by metaPSICOV Stage 2. .	139
4.1	Schematic Representation of Our Sequence-to-Structure Modelling Pipeline.	149
4.2	The framework of the Cotranslational mode of SAINT2.	153
4.3	Representation of the three modes implemented within SAINT2.	154
4.4	Representation of an extrusion step of SAINT2 Cotranslational mode. . . .	155
4.5	Representation of a move step of SAINT2 Cotranslational mode.	156
4.6	Solvation scores for Alanine and Asparagine given for each of the exposition bins defined in SAINT2.	160

4.7	Representation of the Lennard-Jones Potential	162
4.8	Schematic representation of the angles used to compute the Orientation score.	163
4.9	Distribution of the radii of gyration for all proteins in the PISCES Training set.	166
4.10	Regression based on the scaling law of the radius of gyration for all proteins in the PISCES Training set	167
4.11	Heatmap of the $\Delta_{TM-Score}$ for the scoring weight tests performed on the 43 proteins in our Training Data Set.	170
4.12	Run time profile of each score component of SAINT2 before performance optimization.	173
4.13	Run time profile of each score component of SAINT2 after speed optimization.	173
4.14	Run times for an individual run of SAINT2 and ROSETTA for the 43 proteins in our Training Data Set.	175
4.15	Comparison between the TM-Score Best of SAINT2 against ROSETTA for the 43 proteins in our Training Data Set.	176
4.16	Native structure and the best model generated by SAINT2 for CASP11 target T0836.	179
5.1	Estimate of the number of decoys required by SAINT2 to produce a correct answer and a "best" model.	195
5.2	Correlation between the number of decoys required by SAINT2 to produce a correct answer and three sequence-based features.	196
5.3	Correlation between the number of decoys required by SAINT2 to produce a "best" model and three sequence-based features	197
5.4	Relationship between fragment library precision/coverage and the number of decoys required to produce a correct answer using SAINT2.	198
5.5	Relationship between precision of predicted contacts and the number of decoys required to produce a correct answer using SAINT2.	199

5.6	Comparison of the TM-Score Best and TM-Score Top 5 of SAINT2 Cotranslational against SAINT2 <i>In vitro</i>	200
5.7	Comparison of the TM-Score Best and TM-Score Top 5 of SAINT2 Cotranslational against SAINT2 Reverse.	201
5.8	Run times for an individual run of SAINT2 Cotranslational and SAINT2 <i>In vitro</i> for the 41 proteins in our Validation Data Set.	202
5.9	Δ TM-Score Best and Δ Average TM-Score for nine proteins belonging to the Pfam family <i>TatD related DNAses</i> as generated by SAINT2.	203
5.10	Relationship between four structural signatures for cotranslational protein folding and the Δ TM-Score Best obtained for 41 proteins in our Validation Data Set.	205

List of tables

2.1	Training Data Set	62
2.2	PDB-Representative Validation Data Set	65
2.3	Flib Score Cutoffs for Random Extraction	72
4.1	The orientation bins used to classify residue pairs for the Orientation Potential	163
4.2	Initial Scoring Weights for the Score Components in SAINT2	169
4.3	CASP11 Results	178
5.1	<i>TatD related DNAses</i> Data Set.	189

Chapter 1

Introduction

1.1 Life, as we know it

Life is a phenomenon that requires organized information to be self-replicating and adaptable, subject to selective pressure during its replication cycles. The information, in this case, is expressed and contained in the genetic material, usually macromolecules of deoxyribonucleic acid (DNA). The information in the genetic material can be seen as a blueprint for the construction of other macromolecules, which in turn are responsible for the organization, replication, and preservation of the genetic material itself.

Other than DNA, two classes of macromolecules are actively involved in the processes of life. The information contained in the DNA is converted via a complex cell machinery into ribonucleic acid (RNA), in a process known as transcription. RNA molecules are active participants in a myriad of cellular processes, therefore being fundamental for the preservation/replication of the information and, thus, to the sustenance of life. Another intracellular process converts a specific type of RNA (referred to as messenger RNA or mRNA, for short) into proteins, which is known as translation. Proteins mediate the vast majority of known biological interactions, from DNA replication and organization to transcription and translation.

These two conversion steps (transcription and translation) comprise what is known as the central dogma of biology: DNA is transcribed to RNA, which is translated to proteins. Although the conversion is not necessarily unidirectional (i.e. RNA can be reversely transcribed into DNA), the flux of information dictates our perception of life as we know it.

The final component required for life is the action of natural selection. Natural selection is a phenomenon that emerges from the efficiency of a set of living systems at preserving and replicating their genetic information. Biological information can be lost due to many causes. The loss of information is just as important for natural selection as the replication of information. The balance between how much information is lost and how much information is preserved/replicated at a given generation is known as fitness. Natural selection therefore arises as a property of this system, in which the information of fitter organisms will propagate over generations whereas the information of less fitter organisms will decay and ultimately be extinguished/extinct.

The sources for loss of information can change (i.e. changes in the environment), which drives living organisms to become adaptable in order to cope with these changes. Adaptability is achieved via changes to the information that may occur during its preservation/replication, thus ensuring divergence amongst organisms of a same species. To this extent, "higher fitness" and "more adaptable" are often used interchangeably. The drive for adaptation is known as selective pressure and is another emerging property of the system. Selective pressure can be seen as the set of all causes of loss of information at a given time.

Proteins and RNA and, more importantly, the concerted action of several different proteins and RNA confer observable properties to organisms. The set of all observable properties of an organism is known as phenotype. The phenotype is the target of natural selection; the set of observable properties of an organism are the features that can be selected for or against. The genetic material, which is inherited, is known as the genotype.

Here, it is important to highlight that phenotypic variations can also occur due to the action of environmental factors capable of controlling gene transcription. This is known as epigenetics. For instance, in eukaryotes, the genetic material is tightly packed and not necessarily accessible to the transcription machinery. Proteins can modify and control the packing of chromatin in response to external stimuli, thus leading to phenotypic changes that are not encoded by the genetic sequence. Under the light of epigenetics, DNA can be seen not only as a storage for information, but also as another control mechanism for transcription.

Evolution is the notion that the genetic information of all species change over generations to become more adaptable to the selective pressure. It is impossible to define life without considering this process of preservation, replication, and adaptation under selective pressure. Evolution is therefore fundamental for defining and shaping our understanding of life.

We have used our understanding of biology as inspiration to improve the prediction of protein structures *de novo*. In this thesis, we describe several analyses in which we investigated whether different biological features can be used to address the current limitations of protein structure prediction. These biological features were based on concepts from structural biology, evolution, biophysics, and molecular biology, all of which will be described in detail over the next sections.

1.2 Proteins

Proteins are involved in a majority of biological processes. For that reason, understanding the way proteins interact and function is paramount for the elucidation and understanding of life. In this section, we will describe proteins, their composition, their structure and how they are classified/grouped according to their sequence, structure, and function.

1.2.1 Amino Acids

Proteins are biological macromolecules comprised of organic compounds known as amino acids. By definition, an amino acid is a compound that presents two functional groups (an amine and a carboxylic acid) and an organic side chain. Non-side chain atoms are referred to as main chain. Amino acids that present the side chain bound to the α -carbon (C- α), as seen in Figure 1.1, are known as α -amino acids. In this case, the α -carbon is a chiral center, and amino acids are enantiomeric, presenting L and D-oriented isomeric forms.

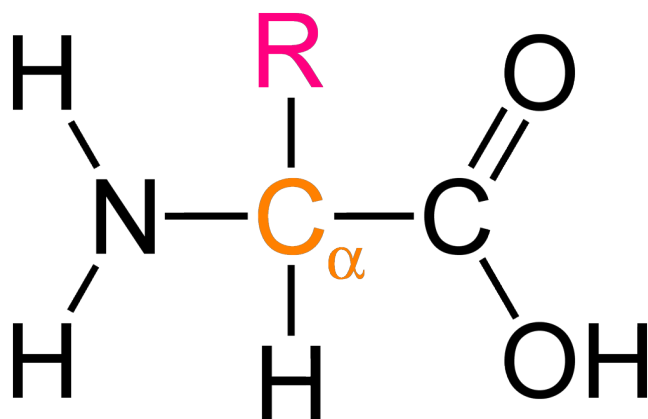


Fig. 1.1 Representation of the structure of an amino acid. The amine group can be seen on the left and the carboxylic acid can be seen on the right. In this particular example, an L- α -amino acid, the organic substituent (represented by the letter R) is bound to the α -carbon and the isomer is L-oriented.¹

With few exceptions, proteins are comprised of 20 distinct L- α -amino acid types (Figure 1.2) (Young and Schultz, 2010). For the sake of simplicity, the term amino acid will now be used to describe these 20 canonical L- α -amino acids. As seen on Figure 1.2, amino acids can be represented by a three-letter code or a one-letter code. They can also be grouped according to their physico-chemical properties. There is a clear distinction between hydrophilic (polar) and hydrophobic (apolar) amino acids, as their propensity to be exposed to or hidden from water is suggested to coordinate protein folding (Agashe et al., 1995).

¹Image by Benjah-bmm27 via Wikimedia Commons.

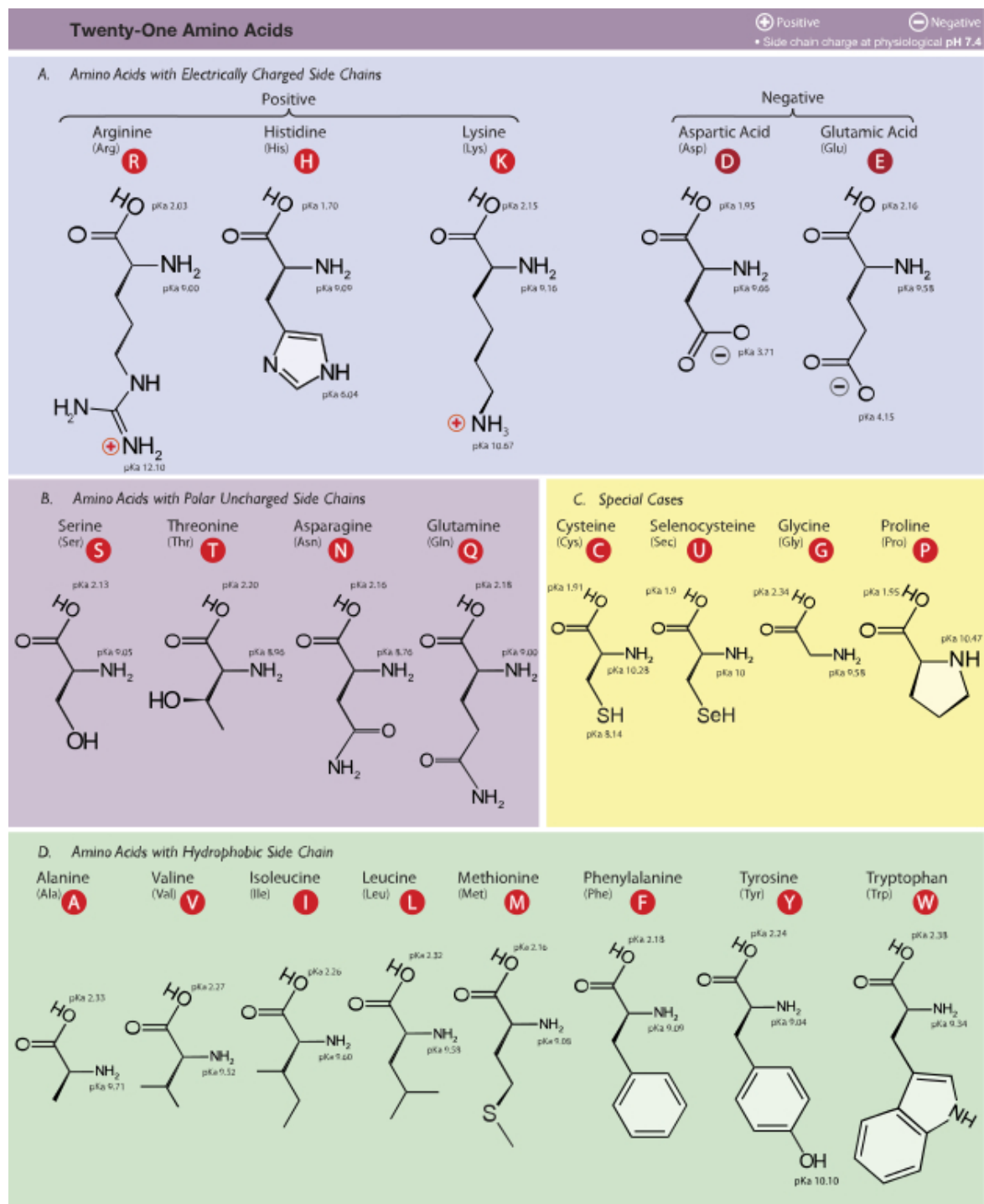


Fig. 1.2 The 20 canonical amino acids that compose proteins, separated according to their physico-chemical properties. A non-canonical amino acid, selenocysteine, is also shown.²

²Image by Dancojocari under Creative Commons (CC BY-SA 3.0 - <http://creativecommons.org/licenses/by-sa/3.0/>), via Wikimedia Commons.

Five amino acids (D,E,H,K,R) are charged at physiological pH, which is particularly important given that electrostatics are one of the dominant forces in protein folding (Dill, 1990). Refer to Section 1.5 for more details. Three amino acids are unique in terms of their physico-chemical properties: cysteine is capable of forming covalent disulfide bonds with other cysteines (known as disulfide bridges), glycine does not present a side chain, and proline is the only amino acid that cannot act as a hydrogen bond donor, given its unique cyclic side chain.

1.2.2 Peptide Bond

Two amino acid molecules can react to form a covalent bond known as a peptide bond; the carboxyl group of one amino acid reacts with the amine group of another (Figure 1.3).

This reaction generates a peptide bond and releases a molecule of water. After the formation of a peptide bond, the remaining atoms of an amino acid are commonly referred to as a residue. An individual amino acid can form up to two peptide bonds, using two of its functional groups (amine and carboxyl). A polypeptide is a linear chain of multiple amino acid residues, linked together by peptide bonds. Main chain atoms in a polypeptide chain are referred to as the backbone. A polypeptide chain presents two termini. The N-terminus is the end of the chain in which the amine group of the terminal residue is not forming a peptide bond. Analogously, the C-terminus is the opposite end of the chain, in which the carboxyl group of the terminal residue is not forming a peptide bond. Residues in a polypeptide are numbered sequentially from 1 to L , where L is the length of the chain, 1 corresponds to the N-terminal residue and L corresponds to the C-terminal residue.

1.2.3 Hydrogen Bonds

Hydrogen bonds are an important interaction in proteins. In this context, a hydrogen bond refers to a weak dipole-dipole attraction that occurs when a hydrogen atom covalently bound

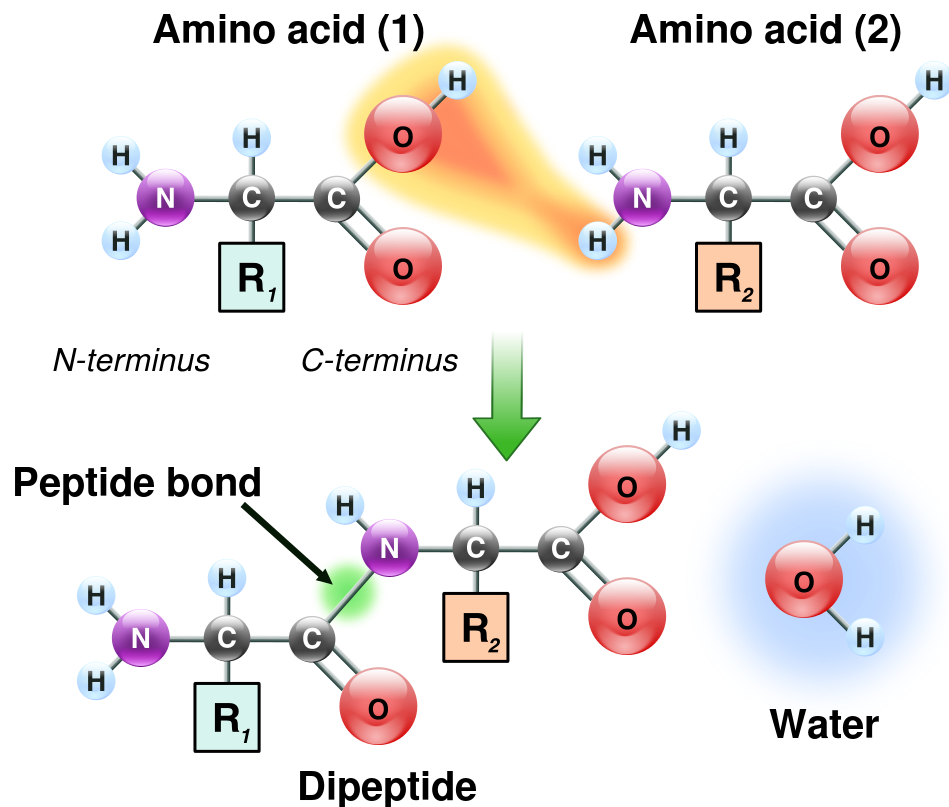


Fig. 1.3 Representation of a peptide bond reaction, in which the amine and carboxyl groups of two different amino acids react, forming a peptide bond and releasing a molecule of water.³

to a strongly electronegative atom (i.e. N,O,F) is in proximity to a lone pair of electrons from another electronegative atom (Desiraju and Steiner, 2001). The atom bound to the hydrogen is called a hydrogen bond donor, whereas the electronegative atom that contributes with the lone pair of electrons is called a hydrogen bond acceptor. Hydrogen bonds are stronger than other type of dipole-dipole interactions, yet not as strong as ionic or covalent bonds.

As seen in Figure 1.3, protein residues present an amide as a result of the peptide bond. The amide is a four-atom functional group -C(=O)NH- , in which both a hydrogen bond donor (N) and acceptor (O) are present. The only exception is proline, which does not present a hydrogen bond donor because of the existence of a bond between its cyclic side-chain and its main chain nitrogen.

³Image by GYassineMrabetTalk, via Wikimedia Commons.

Given the geometry of peptide bonds, a protein residue's amide is incapable of forming a hydrogen bond with itself. However, proteins present multiple peptide bonds/amides tightly packed together, as proteins tend to be highly compact (Lobanov et al., 2008a). Multiple residues present polar side chains which can also form hydrogen bonds. The aggregate effect of all the hydrogen bonds in a protein is one of the forces that define a protein's three-dimensional structure (refer to Section 1.5) (Dill, 1990).

1.2.4 Torsion Angles and Ramachandran Plot

The four main chain amide atoms -C(=O)NH- form a plane which is referred to as the amide plane. For any given α -carbon (C_α) in a protein (with the exception of the two terminal residues), one can use these planes to define three dihedral angles ϕ , ψ , and ω (torsion angles). Torsion angles are used to describe the conformation of the main chain (backbone) atoms of a protein. If the torsion angles are known, the position of backbone atoms can be closely approximated using ideal bond lengths and angles.

A dihedral angle is defined as the angle between two planes. It can be computed using the position of four atoms as seen in Figure 1.4. Torsion angle values range from -180° to 180° , and assume the value of 0° when the orientation of the atoms around the bond is *cis* (Figure 1.4).

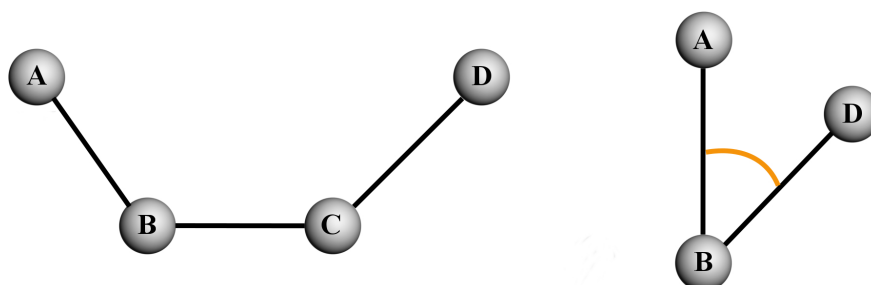


Fig. 1.4 A dihedral angle is defined as the angle formed between two planes rotated around a bond. Dihedral angles are defined to be 0° when the orientation of the atoms around the bond is *cis*.

Each of the three torsion angles is defined by a different set of atoms and around rotations of different bonds (Figures 1.5, 1.6). The ϕ angle of residue i is defined around the rotation of the N- C_α bond and using the atoms C_{i-1}, N, C_α, C (A,B,C,D as shown in Figure 1.4). In this case, $i-1$ is used to indicate atoms of the previous residue in the polypeptide chain (one residue closer to the N-terminus). The ψ angle is defined around the rotation of the C_α -C bond and using the atoms N, C_α, C, N_{i+1} . The ω angle is defined around the rotation of the C-N bond using the atoms $C_{\alpha i-1}, C, N, C_\alpha$. The peptide bond presents a partial double-bond character, which forces two successive C_α s and the C,O,N, and H atoms between them to lay on a plane. Therefore, there is little to no variation in ω , which adopts values of approximately 180° (or -180°).

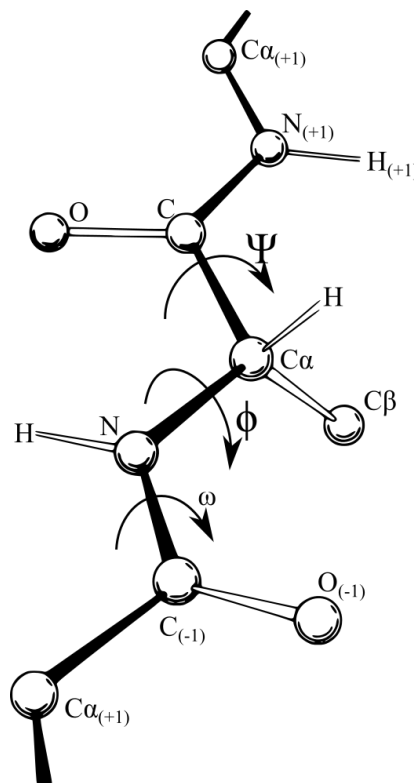


Fig. 1.5 Schematic representation of each of the planes and rotating bonds used to define the three torsion angle ϕ , ψ , and ω .⁴

⁴Image by Dcrjsr, vectorised by Adam Rędzikowski (CC BY 3.0 - <http://creativecommons.org/licenses/by/3.0>), via Wikimedia Commons.

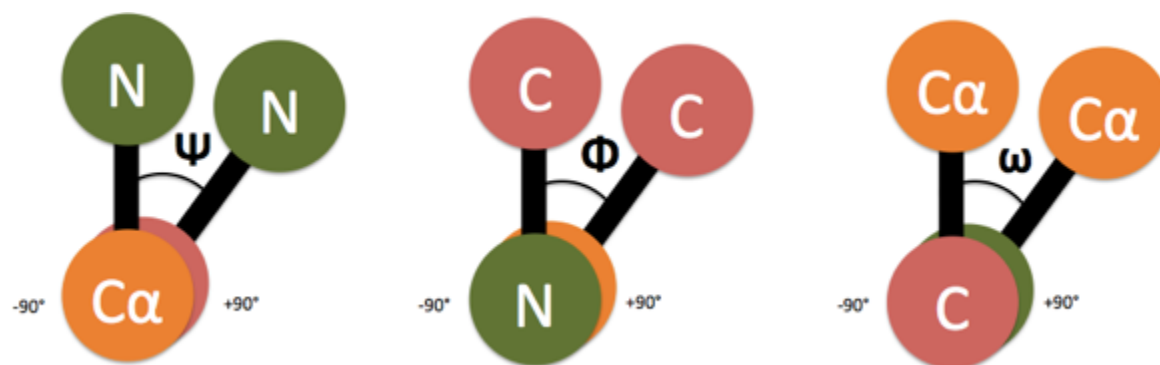


Fig. 1.6 Representation of the set of four atoms and the respective bonds used to define the three torsion angle ϕ , ψ , and ω .⁵

Given that ω is approximately constant, the set of all pairs of torsion angles ϕ, ψ can be used as one-dimensional descriptors of a protein's structure (refer to Section 1.5). One way to visualize these descriptors is by means of a Ramachandran plot (Ramachandran et al., 1963), as seen in Figure 1.7.

The Ramachandran plot can be used to determine if a given protein conformation is native-like, based on the observed torsion angles of all known protein structures and their distribution on the plot. The diagram is also useful to ascertain the torsion angle preferences of each amino acid. There is a relationship between the Ramachandran plot and secondary structure; certain intervals/areas of the Ramachandran plot correspond to specific secondary structure types (refer to Section 1.4.3). Certain Ramachandran plot areas are rarely populated due to steric constraints imposed by side chains.

1.2.5 Residue Conformation and Rotamers

As seen in the previous section, the torsion angles ϕ, ψ, ω can be used to describe the conformation of the backbone of a protein. However, to fully describe a residue's conformation (the approximate position, in space, of its atoms) it is necessary to consider the position of the side chain. In a similar fashion to the main chain, side chain atoms/bonds can be used to

⁵Image by Eric Vallabh Minikel under Creative Commons (CC BY-SA 4.0 - <http://creativecommons.org/licenses/by-sa/4.0>), via CureFFI.org

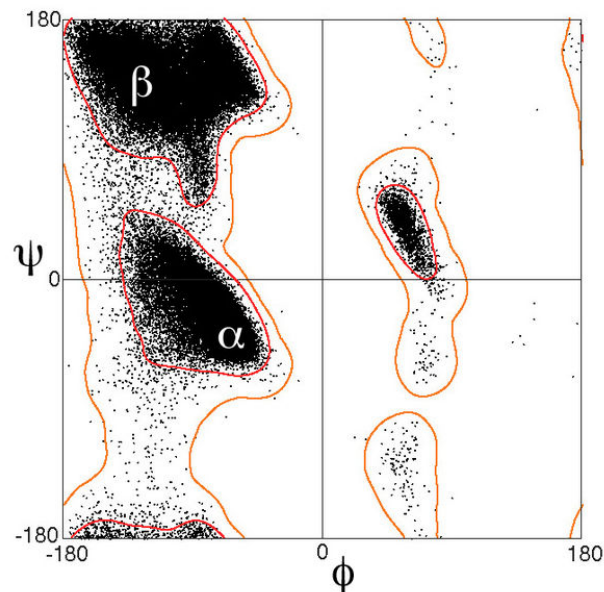


Fig. 1.7 A Ramachandran plot can be used to visualize the torsion angles (ϕ, ψ) of a protein. Each point represents a protein residue. Horizontal and vertical lines correspond to 0° values for ϕ and ψ , respectively. Certain areas of the plot (contour lines) correspond to specific secondary structure types.⁶

define a set of dihedral angles, referred to as χ_i , where i is a numeric identifier (Figure 1.8). Note that the side chain is different for different amino acids and the numbers of side chain dihedral angles will depend on the size of the side chain.

Due to proteins being tightly compact, there are some energy barriers to rotations about single bonds of residues' side chains. For that reason, residues are thought to have discrete conformational states. These low energy side chain conformations are known as rotamers (Dunbrack and Karplus, 1993; McGregor et al., 1987; Ponder and Richards, 1987; Shapovalov and Dunbrack, 2011). Rotamer libraries, such as the one described in Shapovalov and Dunbrack (2011), describe frequencies, mean dihedral conformations and their standard deviation and are extensively used in protein structure determination and prediction (refer to Sections 1.5.2 and 1.8, respectively).

⁶Image by Dcrjsr under Creative Commons (CC BY 3.0 - <http://creativecommons.org/licenses/by/3.0>), via Wikimedia Commons.

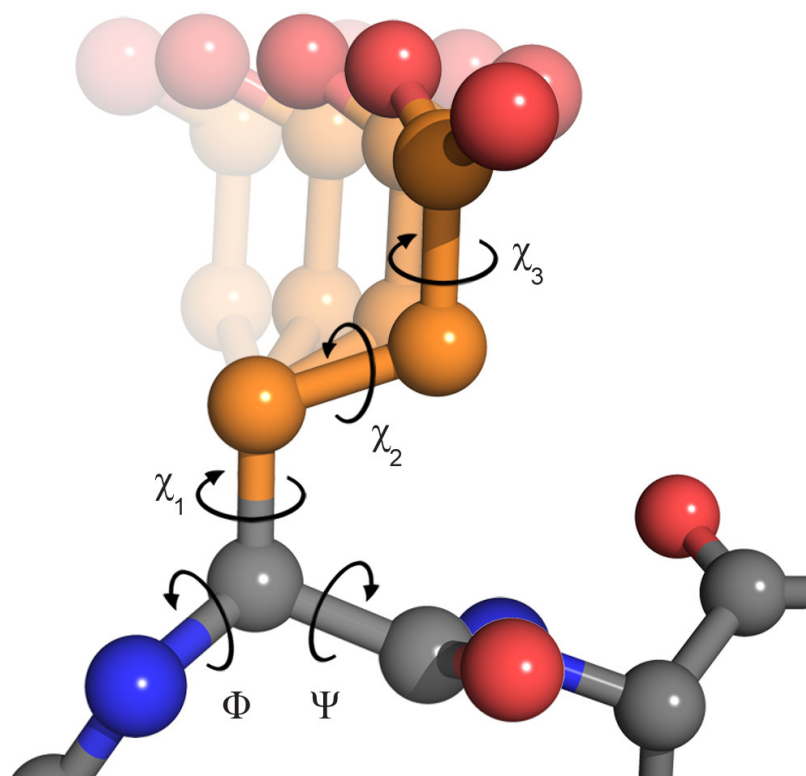


Fig. 1.8 Representation of chi dihedral angles for two different residues in a polypeptide. Residues with longer side chains need to be described by a larger number of χ angles.⁷

1.3 Protein Sequence (Primary Structure)

The amino acid composition of a protein is known as protein sequence or primary structure. A protein sequence is represented with the one-letter code for each of the protein's residues, starting from the N-terminus. There is a belief that all the information required to obtain a protein's structure is contained in its sequence (Anfinsen, 1972), which implies that given a protein's sequence, we should be able to predict its structure (refer to Section 1.8 for more details). Other than their relevance for protein structure prediction, protein sequences are useful for protein classification (Section 1.7), homology detection, and functional characterization.

The number of protein sequences available in the public domain is growing exponentially (Kahn, 2011). Most protein sequences originate from genome projects/genetic sequencing

⁷Reproduced from Harder et al. (2010) under Creative Commons (CC BY 4.0 - <http://creativecommons.org/licenses/by/4.0>), via BioMed Central.

and can be manually curated or predicted from coding DNA sequences (known as automatic annotation). One of the main repository for protein sequence information is the UniProt (Bairoch et al., 2005). The data in UniProt is subdivided into two databases: a manually curated database (UniProtKB/Swiss-Prot) and an automatically annotated database (UniProtKB/TrEMBL). UniProtKB/Swiss-Prot presents highly curated non-redundant data, which is based on literature. At the moment, UniProtKB/Swiss-Prot presents approximately 550,000 protein sequences. The process of manual curation is not fast enough to accommodate the large amount of sequencing data that is being generated. For that reason, UniProtKB/TrEMBL was conceived to provide a database of automatically annotated protein sequences. As of August 2015, UniProtKB/TrEMBL has approximately 50 million sequences.

1.3.1 Sequence Alignment

Sequence alignments have broad applications in bioinformatics and are at the core of many applications, from functional annotation to protein structure prediction. Sequence alignments are used to ascertain the similarity between two or more protein sequences. This is particularly useful to detect homology between proteins. Homology can be seen as an evolutionary relationship in which a pair of homologous proteins (referred to as homologues) share a common evolutionary ancestry and tend to present similar structures and function.

The problem of aligning two sequences can be solved in $O(n \times m)$ using a dynamic programming approach, where m and n are the lengths of the two sequences. Alignments can be performed in a global (full sequence) or local (two sub-sequences that present high similarity) fashion. Global alignments are usually used when the similarity between the sequences is high and they have similar lengths. The general global alignment approach is the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). This routine can be divided into two steps, the computation of a dynamic programming matrix, which contains the best

scores for all possible sub-alignment paths, and a trace back step in which the dynamic programming matrix is used to determine the best possible alignment.

Local alignments are used when it is more meaningful to ascertain local similarities between the two sequences, such as when the two sequences are not sufficiently similar or when their lengths drastically differ. The general approach for local alignment is the Smith-Waterman algorithm (Smith and Waterman, 1981). This method is analogous to the Needleman-Wunsch, with differences in the way the dynamic programming matrix is computed and traced back.

Several measures can be used to establish homology between two aligned protein sequences. One of these measures, sequence identity, refers to the number of identical amino acids (matches) in both aligned sequences. When sequence identity is sufficiently high (>40%), a homology relationship can be established. However, the evolutionary signal becomes noisy in what is known as the twilight zone, between 20% and 35% sequence identity (Rost, 1999). Proteins presenting less than 20% sequence identity are less likely to be evolutionarily related and more advanced techniques are required in such cases to fully ascertain homology.

Another commonly used measure to assess homology is sequence similarity. Sequence similarity takes into account how similar the two sequences are both in terms of their identical residues and in terms of residues with similar physico-chemical properties. Other than sequence identity and similarity, most alignment software also output an alignment score. This score is highly dependent on the length of the aligned (sub)sequences and the scoring scheme that was used to generate the alignment and, without a proper statistical treatment, is not suited to establish homology.

BLAST (Camacho et al., 2009), a commonly used tool for local sequence alignment, takes a protein sequence as input and searches this target sequence against all available sequences in a given database. One of the main applications of BLAST is to detect homologues. BLAST

provides an E-value, which describes the expectation to find an alignment with that particular score given the parameters used in the alignment. A p-value (probability) can be extrapolated from this expectation, which provides a quantitative assessment of the significance of a given alignment. For E-values < 0.01 , the P-value and E-value are nearly identical and, for the sake of simplicity, only E-values are reported by BLAST.

Substitution Matrices

In order to perform a sequence alignment, it is necessary to define a scoring system that describes the cost of mismatching a given amino acid for another. These substitution costs are described in a substitution matrix. Different substitution matrices can be employed depending on the similarity between the two sequences, and depending on the environment in which these residues occur. BLOSUM (Henikoff and Henikoff, 1992) and PAM (Dayhoff and Schwartz, 1978) are two commonly used substitution matrices.

The BLOcks SUbstitution Matrix (BLOSUM) is derived from frequencies of amino acids as computed on very conserved protein family regions. The amino acid frequencies in these very conserved regions are used to assign a log-odds score of a given amino acid being substituted by another. BLOSUM matrices are based on local alignments and numbered according to the percentage of identity in the alignments, where a higher number indicates higher sequence identity. For instance, BLOSUM62 is computed from sequences that do not share more than 62% sequence identity.

Point Accepted Mutation (PAM) matrices are extrapolated from phylogenetic information of global sequence alignments between proteins that share high sequence identity ($> 85\%$). The matrix PAM1 is calculated directly from the frequencies in the alignments, whereas other PAM matrices can be computed using PAM1. The numbering of PAM matrices indicates evolutionary distance, i.e. nearly identical sequences should be aligned with PAM1 and PAM matrices with higher numbers should be used as sequences become more dissimilar.

The frequencies of amino acids can vary according to the environment in which these amino acids are inserted in. For instance, a polar amino acid is more likely to be observed on the surface of a protein as opposed to its core. For that reason, a polar amino acid is also more likely to be substituted by another polar amino acid if it sits on the protein's surface, yet more likely to be substituted by an apolar amino acid if it is buried. Environment-specific substitution matrices were created based on this notion. These scoring matrices are constructed in a similar way to BLOSUM, but the frequencies are computed for each environment. Therefore, each environment is described by a different substitution matrix where each matrix describes a propensity of a given amino acid to be substituted by another in that particular environment.

Environment-specific substitution matrices have been developed to consider the secondary structure (refer to Section 1.4) of amino acids as environments (Choi and Deane, 2010; Overington et al., 1992; Shi et al., 2001). These environment-specific substitution matrices quantify the propensity of amino acids substitutions to occur within each of the secondary structure types, thus incorporating secondary structure information when performing sequence alignment.

Multiple Sequence Alignments

Three or more protein sequences that present an evolutionary relationship can be aligned to create a multiple sequence alignment (MSA). MSAs are used to establish conserved regions between a number of protein sequences and can be used in a wide range of structural bioinformatics applications such as functional characterization and secondary and tertiary protein structure prediction. The computation of an MSA is an NP-Hard problem (refer to Section 1.8.6 for more details). By definition, if a problem H is said to be NP-Hard, every decision problem that cannot be solved in polynomial time using a deterministic Turing

machine can be reduced to H. Therefore, an NP-Hard problem is at least as hard as any problem in NP.

An approximation commonly used to align multiple sequences is known as progressive or tree-based MSA. It requires the computation of a guide tree (Hogeweg and Hesper, 1984), which describes the similarity between each of the sequences to be aligned. Once the guide tree is computed/obtained, the MSA is built beginning with a pairwise sequence alignment of the most similar sequences, and incrementally aligning new sequences according to the guide tree. Progressive methods tend to propagate errors; an error at any stage of the alignment will be propagated in every subsequent alignment. Other heuristics attempt to address this issue at the cost of performance (i.e. by revisiting the initial alignments as the routine progresses). Nonetheless, the progressive method, implemented in software such as Clustal ω (Sievers et al., 2011), is one of the most widely used MSA generation approach.

Profile Hidden-Markov Models

A profile Hidden-Markov Model (profile HMM) is a compact representation of an MSA, in which the probabilities of observing each of the 20 amino acids are described for every sequence position. Profile HMMs also describe the probabilities of observing insertions and deletions along the alignment, which distinguishes profile HMMs from sequence profiles. Profile HMMs define three states: match, insertion, and deletion. These states are hidden, i.e. given a set of observations (the residues in a protein sequence), the state for each of these observations is unknown.

Protein sequences can be aligned to profile HMMs leading to the computation of a log-odds score. In this case, the profile HMM is used to obtain a probability of observing the target sequence conditional to that particular profile. A log-odds score can be defined as the log of the ratio between the conditional probability and the probability of the null model (the product of the frequencies of the amino acids in the target sequence).

Another class of methods consists in generating a profile HMM for the target sequence and aligning that profile HMM to other profile HMMs, usually pre-computed for a sequence database. This approach, implemented in methods such as HHBlits (Remmert et al., 2012), is the most sensitive methods for homology detection (Söding, 2005; Söding and Remmert, 2011). Two profiles HMMs can be aligned using a log-sum-of-odds score, which can be computed using the Viterbi algorithm (Viterbi, 1967).

1.4 Secondary Structure

Given the high propensity for hydrogen bond formation between protein backbone amides, there are several local structures (atomic arrangements) that are commonly observed amongst all proteins (Kabsch and Sander, 1983a). These patterns reflect local arrangements of protein residues and are referred to as Secondary Structure (SS). Secondary structure is traditionally defined based on the patterns of hydrogen bond formation between protein backbone atoms and their geometric features (Kabsch and Sander, 1983a).

A widely accepted definition of secondary structure types is given by DSSP (Kabsch and Sander, 1983a), a Dictionary of Protein Secondary Structures. Note that DSSP is the name used both for the database of protein secondary structures and the name of the DSSP software, which is used to calculate SS.

DSSP defines eight different types of SS:

1. **Turn (T)**: Turns define a single hydrogen bond from the CO of residue i to the NH of residue $i + j$, where $j = 3, 4, 5$. The residue separation can be used to further specify the turn as 3-turn, 4-turn, or 5-turn. Residues are only classified as T if they are not part of a helix. Helices are defined by at least three consecutive turns.
2. **3_{10} -helices (G)**: These helices are defined by three or more consecutive 3-turns and, therefore, present a hydrogen bonding pattern between residues i and $i + 3$.

3. **α -helices (H)**: These helices are defined by four or more consecutive 4-turns and, therefore, present a hydrogen bonding pattern between residues i and $i + 4$.
4. **Π -helices**: These helices are defined by five or more consecutive 5-turns and, therefore, present a hydrogen bonding pattern between residues i and $i + 5$.
5. **Isolated β -bridge (B)**: bridges are defined when two triplets of consecutive residues $(i - 1, i, i + 1)$ and $(j - 1, j, j + 1)$ form hydrogen bonds following a specific pattern. Bridges can be parallel or anti-parallel depending on which atoms are involved in the hydrogen bonding pattern. A parallel bridge is defined by two patterns: when two specific hydrogen bonds are formed from CO of $i - 1$ to NH of j (1) and from CO of j to NH of $i + 1$ (1), or when two hydrogen bonds are formed from CO of $j - 1$ to NH of i (1) and from CO of i to NH of $j + 1$ (2). Analogously, an anti-parallel bridge is defined by two patterns: when two specific hydrogen bonds are formed from CO of i to NH of j (1) and from CO of j to NH of i (2), or when two hydrogen bonds are formed from CO of $i - 1$ to NH of $j + 1$ (1) and from CO of $j - 1$ to NH of $i + 1$ (2). An isolated β -bridge is assigned when only one residue is involved in the β -bridge.
6. **β -strand (E)**: when two or more consecutive residues are involved in β -bridges, residues are considered to be part of a β -strand. Strands can be parallel or anti-parallel and can form a β -sheet.
7. **Bend (S)**: Bends are defined as regions with high curvature, in which the curvature is computed by the angle formed between $C_\alpha(i - 2)\hat{C}_\alpha(i) - C_\alpha(i + 2)$. A residue is assigned to be a bend whenever this angle exceeds 70° .
8. **Loop (C,"-" or " ")**: Loop residues, also referred to as coil, are residues that do not present a defined secondary structure, i.e. not categorized in any of the previously described SS types.

The three most common secondary structure types are, in order of their prevalence, loops, α -helices and β -strands (Zhang and Zhang, 1999). α -helices and β -strands will be described in more detail over the next sections.

1.4.1 α -helices

Right-handed α -helices represent one of the two most prevalent secondary structure types observed in all known protein structures (Zhang and Zhang, 1999). Given the particular hydrogen bonding pattern of the residues in α -helices, these helices present a well-defined geometry.

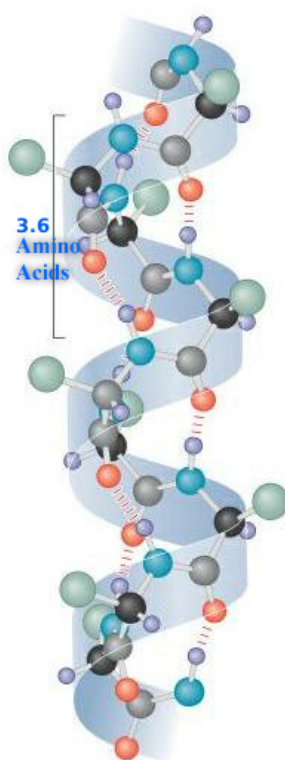


Fig. 1.9 Schematic representation of an α -helix. The hydrogen bonding pattern of α -helices confer a particularly well-defined geometry, in which a full turn is completed every 3.6 residues and leading to a rise of approximately 5.4Å.⁸

⁸This image is a work of the National Institutes of Health, part of the United States Department of Health and Human Services. As a work of the U.S. federal government, the image is in the public domain.

A full turn of an α -helix is completed every 3.6 residues, leading to a rise (distance between successive C- α s in the helix) per turn of approximately 5.4Å. α -helix residues can form up to two backbone hydrogen bonds, i.e. a given α -helical residue i can form hydrogen bonds with residues $i - 4$ and $i + 4$. A schematic representation of an α -helix is provided in Figure 1.9.

1.4.2 β -strands and β -sheets

Given their patterns of hydrogen bonding, β -strands are partially extended residue segments. A fully extended residue segment is one that presents torsion angles (ϕ, ψ) of 180° for all of its residues. As can be seen in Section 1.4.3, the torsion angles of β -strand residues vary significantly from the fully extended conformation. Each partially extended strand can be connect to other β -strands via several hydrogen bonds, forming a pleated β -sheet (Figure 1.10). The pleated character of the sheet is conferred due to the tetrahedral geometry of carbon atoms.

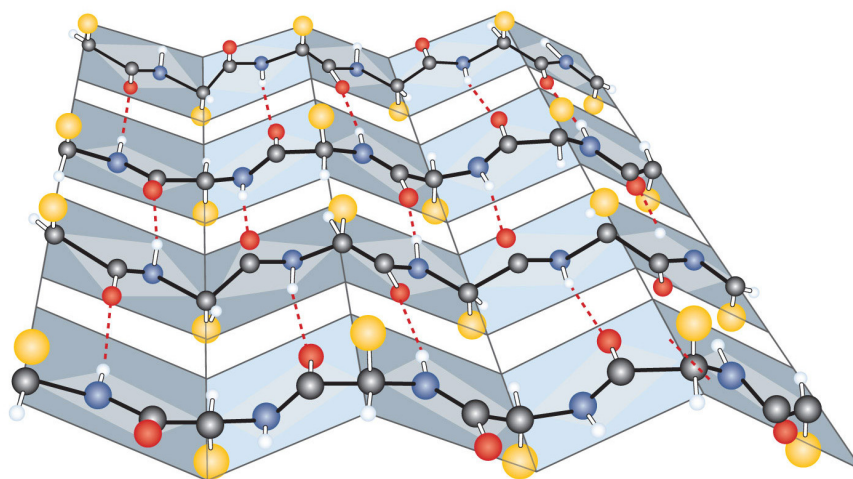


Fig. 1.10 Schematic representation of an anti-parallel β -sheet. The hydrogen bonding pattern of β -strands causes each strand to be partially extended. The pleats occur due to the tetrahedral geometry of Carbon.⁹

⁹Image by Anonymous Author under Creative Commons (CC BY-NC-SA 3.0 - <http://creativecommons.org/licenses/by-nc-sa/3.0>).

Depending on the type of β -bridge formed between the strands, the β -sheet can be parallel or anti-parallel (Figure 1.11). Given that polypeptides present a directionality from N-terminus to C-terminus, β -strands are often represented by arrows starting at the most N-terminal residue of the strand and pointing towards the most C-terminal residue of the strand. The simplest way to visualize the difference between parallel and anti-parallel sheets is to consider that in a parallel beta-sheet all the arrows point towards the same direction (i.e. the beginning of the first strand interacts with the beginning of the second and so forth), whereas in anti-parallel sheets alternating strands point to opposite directions (i.e. the beginning of the first strand interacts with the end of the second).

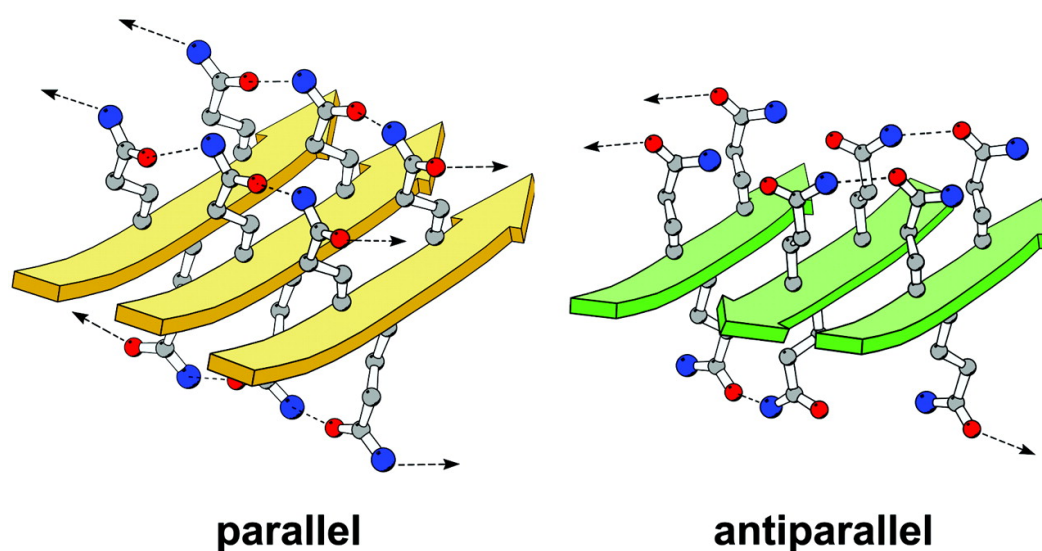


Fig. 1.11 Difference between parallel and anti-parallel β -sheets.¹⁰

1.4.3 Relationship between Secondary Structure and the Ramachandran Plot

As seen in the previous sections, the specific hydrogen bonding patterns observed in different secondary structure types confer a well-defined geometry to protein residues (Kabsch and Sander, 1983a). This geometry is held in place by the stabilizing and aggregate effect

¹⁰Reproduced from Bevivino and Loll (2001), copyright © by the National Academy of Sciences.

of the multiple hydrogen bonds that form secondary structures, which in turn restrict the backbone conformations (Dill, 1990; Kabsch and Sander, 1983a). For that reason, there is a relationship between certain torsion angle intervals and the secondary structure, as can be seen in Figure 1.7. The Ramachandran plot can be used to estimate a residue's secondary structure without resorting to the identification and characterization of hydrogen bonding patterns. Loop residues can occupy both the plot regions designated to α -helices and β -strands.

1.5 Protein Structure (Tertiary Structure)

Protein Structure (or tertiary structure) refers to the three-dimensional arrangement of all atoms of a polypeptide chain (Figure 1.12). Protein structures are formed as a result of several forces that cause these structures to be held in place (Dill, 1990). For instance, the formation of covalent bonds between cysteine residues and the formation of hydrogen bonds can favour specific conformations. Hydrophobicity plays a huge role in conferring a protein its structure, as different residue types will have different preferences for being buried or exposed to solvent. Van der Waals interactions and electrostatics also have an effect on the arrangement of protein atoms. The combination of all the forces that act to determine a protein structure can be used to define an energy associated with a particular protein conformation (Onuchic et al., 1997). There is a consensus that proteins will assume their global minimal energy conformation, which is thought to represent their native state (Anfinsen, 1972). This is known as the thermodynamic hypothesis. However, there are cases where more than one native conformation have been observed (Berkenpas et al., 1995; Govindarajan and Goldstein, 1998). For instance, certain human pathologies are a consequence of misfolding, where proteins are entrapped in alternate conformations capable of aggregating (Thomas et al., 1995).

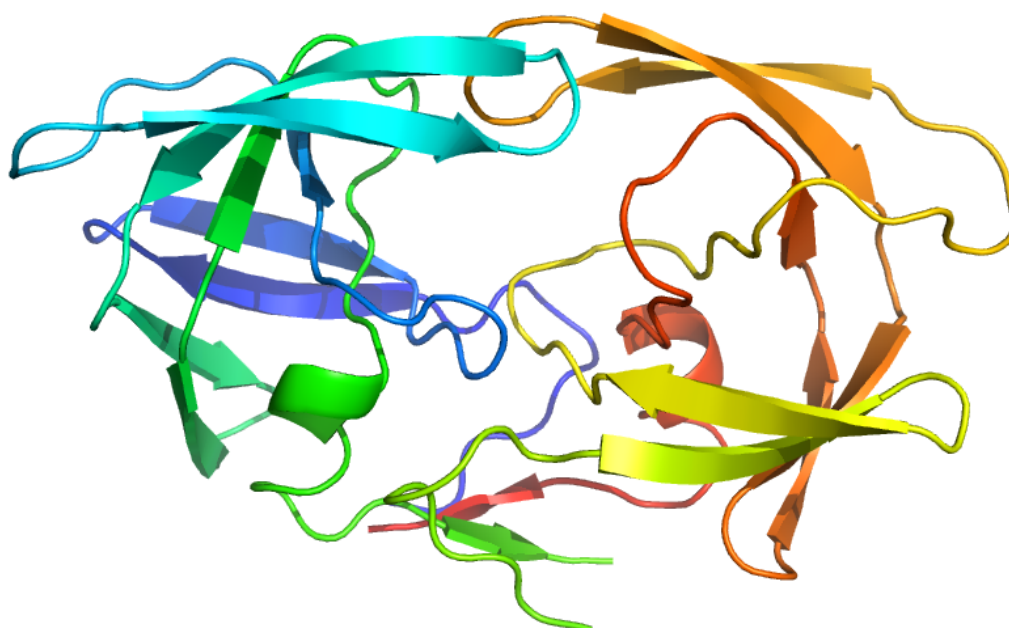


Fig. 1.12 Cartoon representation of the structure of HIV-1 Protease (PDB entry: 1HSG).

The process via which proteins assume their native conformation is known as protein folding. It is important to consider the way in which proteins attain their structure given that the number of possible protein conformations is very large. For a given protein with 100 residues, if each residue can assume up to three possible conformations, this would result in a rough estimate of 3^{100} possible conformations. A significant number of these conformations will lead to physically infeasible atom arrangements as steric clashes are likely to occur. Nonetheless, the number of physically feasible conformations is still extremely large. If a protein were to rapidly sample every single possible conformation, it would take longer than the age of the universe to sample all physically feasible conformations. Yet, most proteins fold in micro to milliseconds (Debès et al., 2013; Hartl and Hayer-Hartl, 2009; Ivankov et al., 2003; Zou et al., 2014). This is known as Levinthal's paradox (Levinthal, 1968).

The proposed solution to Levinthal's paradox is that a protein does not sample every physically feasible conformation during folding (Levinthal, 1968; Zwanzig et al., 1992). The energetics of folding can direct which conformations are sampled, as the abundance of each

conformation follows a Boltzmann distribution and is dependent on the conformation's free energy (Levinthal, 1968; Onuchic et al., 1997). The term energy landscape is often used to describe the variation of energy for different conformations. One of the proposed solutions to Levinthal's paradox suggests that the energy landscape is shaped like a funnel (Figure 1.13) (Bryngelson et al., 1995; Levinthal, 1968), thus directing any starting conformation towards the native state (global energy minimum; bottom of the funnel). It is also suggested that low energy conformation intermediates occur in order to direct protein folding (Bryngelson et al., 1995; Dill and Chan, 1997; Onuchic et al., 1997).

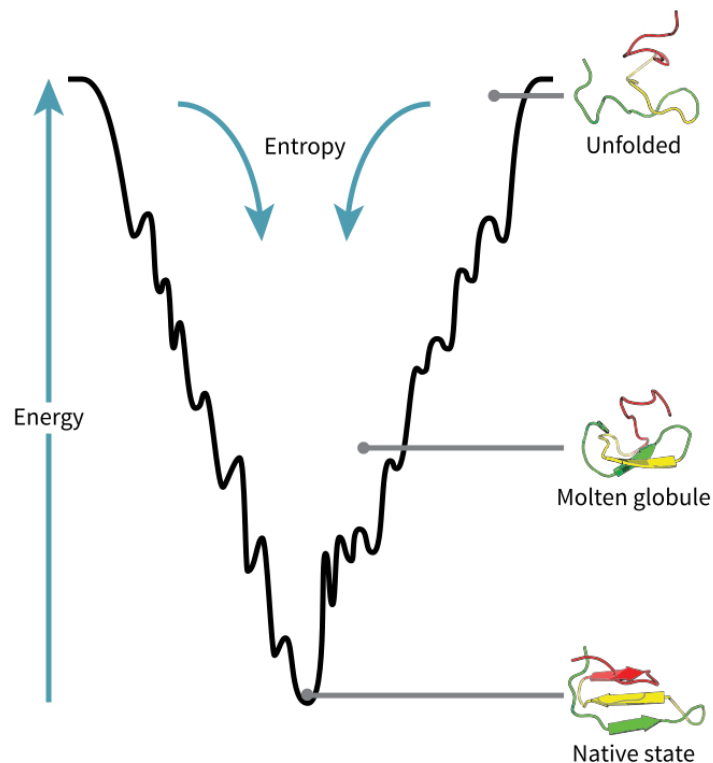


Fig. 1.13 Representation of a funnel-shaped conformational energy landscape. The energetics of folding direct which conformations are sampled, therefore guiding a protein towards its correct structure.¹¹

¹¹Image by Thomas Splettstoesser (www.scistyle.com) (CC BY-SA 3.0 - <http://creativecommons.org/licenses/by-sa/3.0>), via Wikimedia Commons.

1.5.1 The Protein Data Bank (PDB)

The majority of protein structures that have been determined experimentally (Section 1.5.2) are deposited in a publicly available database known as the Protein Data Bank (PDB) (Berman et al., 2000). The most recent build of the PDB (as of September, 2015) contains approximately 110,000 protein structures, which includes the structures of proteins and protein complexes obtained using different experimental techniques.

The PDB collates multiple information about protein structures and how these structures were derived. For each entry, the PDB provides data on the protein sequence, secondary structure, and a list of the atomic coordinates. The structures deposited on the PDB need to fulfil certain quality requirements, but do not need to be complete. Therefore, structures in the PDB may have missing atoms spanning multiple residue segments that could not be resolved due to experimental constraints. The PDB also provides some of the experimental details (remarks) associated with each entry along with any protein classification data (refer to Section 1.7 for more details).

1.5.2 Protein Structure Determination

Currently, the majority ($\approx 89\%$) of the structures available in the PDB have been determined using an experimental technique known as X-Ray crystallography. Of the remaining structures, $\approx 10\%$ have been determined via protein Nuclear Magnetic Resonance spectroscopy (NMR) and fewer than 1,000 structures have been determined using other techniques such as Cryo-Electron Microscopy. The protein structures described in this work have all been determined using X-ray crystallography and, for that reason, this experimental technique shall be discussed in further detail.

Protein X-Ray Crystallography

Protein X-ray crystallography consists of bombarding a protein crystal with a focused beam of X-rays. The electrons in the crystal have a small chance of causing the X-rays to diffract. The diffracting rays can be detected and recorded using a detector to constitute a diffraction pattern (Figure 1.14). Crystals are required for this process because the diffraction of a single molecule is too weak to be measurable. The diffraction signal needs, therefore, to be amplified by using an ordered three-dimensional lattice (a crystal). The resulting diffraction pattern can be used to compute the density of electrons in a crystal by means of Fourier transforms. The electron density, in turn, can be used to determine the atomic position of the crystal constituents (namely, the protein's structure).

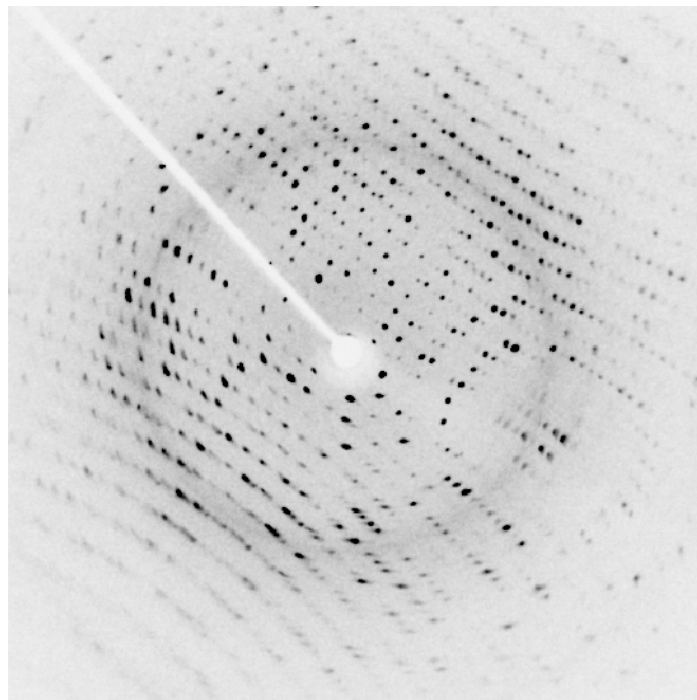


Fig. 1.14 A diffraction pattern obtained by recoding the diffraction of X-rays bombarded onto a crystal.¹²

Given that a protein crystal is required by X-Ray crystallography (Figure 1.15), the first step in this technique is to perform protein crystallization. The target protein is usually

¹²Image by Del45 available in the public domain via Wikimedia Commons.

expressed *in vivo* and purified before crystallization. After purification, the protein solution needs to be concentrated and placed in a crystallization buffer in order for a crystal to form. Different proteins require different sets of conditions in order for crystallization to occur and several proteins fail to crystallize. Crystallization of membrane proteins is particularly challenging as these proteins are usually inserted in a hydrophobic environment (the membrane).

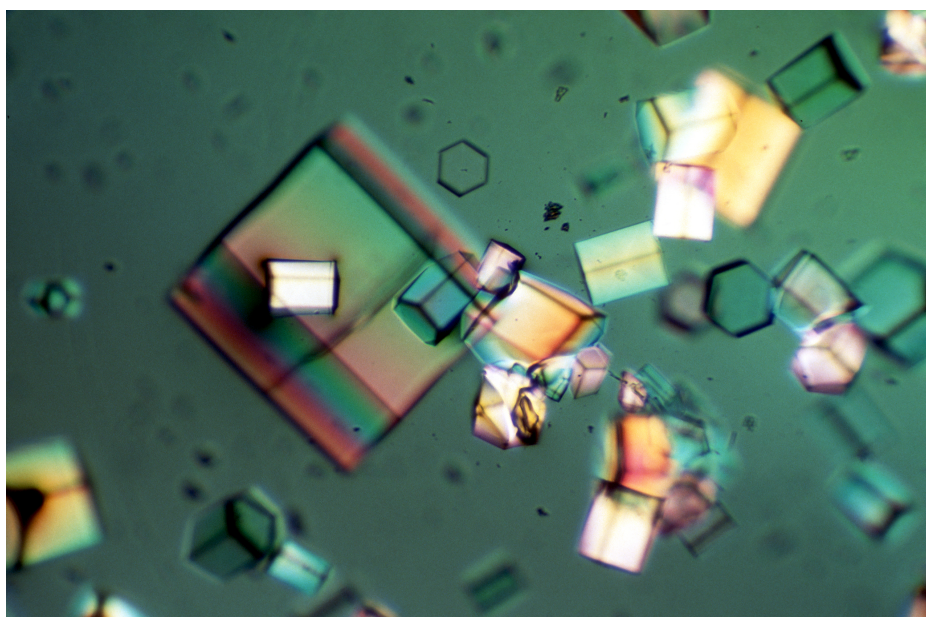


Fig. 1.15 Photo of lysozyme crystals.¹³

The quality of a diffraction pattern can be measured by its resolution. Resolution is a measure of the distance used to distinguish between two observable features in an electron density map. In other words, if two objects (i.e. atoms) are closer than this distance, they appear as a combined density rather than separate objects. Lower resolution diffraction patterns present a lower level of detail and lead to poorer quality models compared to higher resolution data. A commonly used resolution cutoff is 2.0Å (resolution values of less than 2.0Å are considered to be high resolution). The reported resolution for a given protein

¹³Image by CSIRO under Creative Commons (CC BY 3.0 - <http://creativecommons.org/licenses/by/3.0>), via Wikimedia Commons.

structure corresponds to the best resolution of a crystal. Other parts of the crystal are likely to have worse resolution and, thus, a poorer quality.

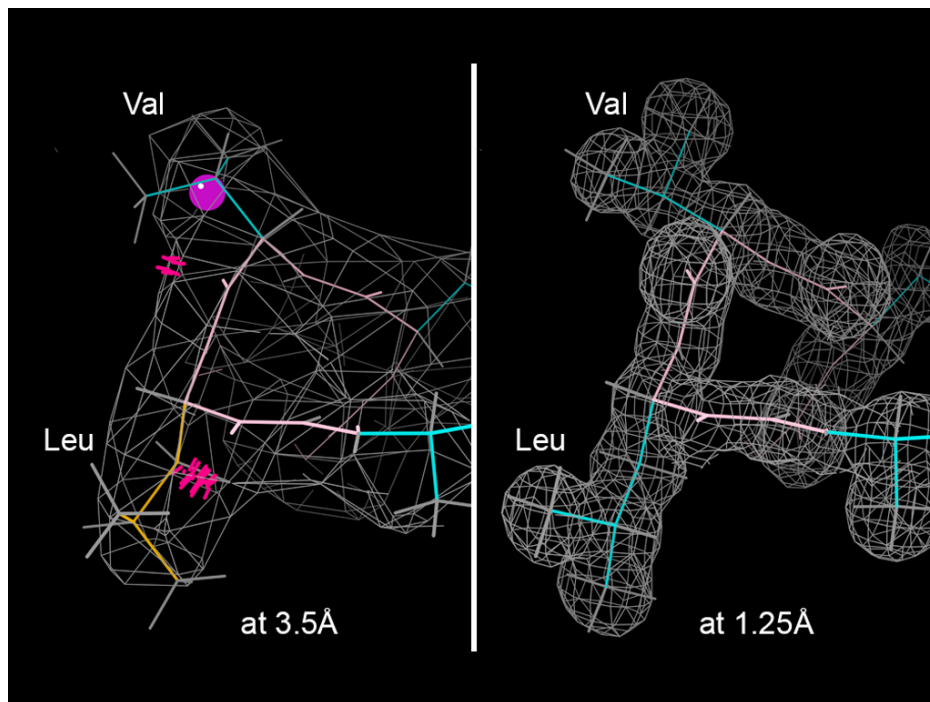


Fig. 1.16 Electron density maps for protein structures with two different resolutions. Lower resolutions lead to higher quality protein structures.¹⁴

Whereas the quality of a crystal is given by the resolution, the quality of a structure model is given by its R-value. The R-value measures the correspondence between a calculated diffraction pattern (generated using the structure model) and the experimentally observed diffraction pattern. Lower values of R-value represent a better fit between simulated and observed diffraction patterns, thus being indicative of better model quality. Another measure used to describe atomic models is the B-factor, which is a measure of uncertainty. There is a B-factor associated with every atom coordinate obtained experimentally, i.e. an uncertainty in regards to the position of each triplet of atomic coordinates. Protein regions with high B-factors are less precise and poorly represented by the structure model.

¹⁴Image by Dcrjsr available under Creative Commons (CC BY 3.0 - <http://creativecommons.org/licenses/by/3.0>), via Wikimedia Commons

1.5.3 Structure Alignment (Superposition)

The similarity between two protein structures can be ascertained by protein structure alignment (superimposition), as shown in Figure 1.17. Structure alignment is particularly useful at detecting protein homology when the sequence similarity between the two proteins is low.

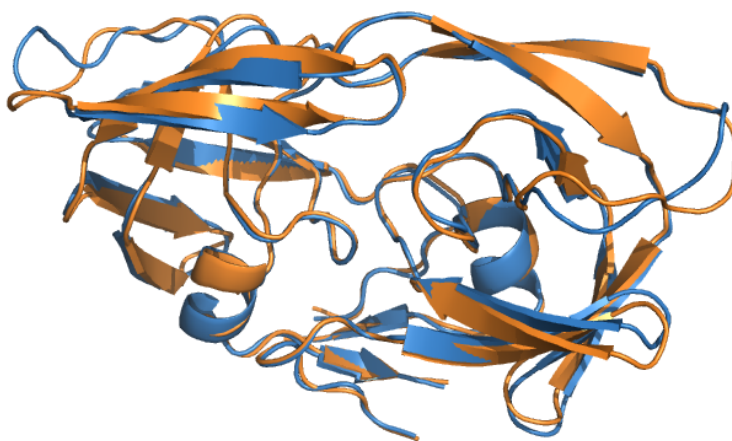


Fig. 1.17 Cartoon representation of the structure of SIV Protease (PDB entry: 1TCW), in blue, superimposed to the structure HIV-1 Protease (PDB entry: 1HSG), in orange.

One of the most common approaches used to align two protein structures relies on a pre-determined mapping between the two proteins (Zhang and Skolnick, 2005). This mapping describes the equivalence between the residues of the two sequences (i.e. sequence alignment). Given a mapping, two structures can be superimposed by rotating one of the structures until a minimal score is obtained. The main score used in this type of alignments is the coordinate root-mean square deviation (RMSD), which is defined as the root-mean square distance between corresponding residue pairs (Kabsch, 1978). The RMSD can be used as a measure of similarity between two superimposed protein structures, but it has limitations. For instance, a small number of local deviations can lead to high (bad) RMSD values (Betancourt and Skolnick, 2001). More importantly, RMSD is length-dependent and a pair of aligned longer proteins will invariantly present higher RMSD than an equally similar pair of aligned shorter proteins (Betancourt and Skolnick, 2001).

Another class of structural alignment methods does not rely on any knowledge of the equivalence between the residues in the two proteins (Zhang and Skolnick, 2005). In these cases, an optimal alignment needs to be estimated as part of the structure superimposition process. A majority of the methods, such as DALI (Holm and Sander, 1993) and TM-Align (Zhang and Skolnick, 2005) fall in this category.

Given the limitations of RMSD, other scores have been developed to ascertain the similarity between two protein structures (Zemla, 2003a; Zhang and Skolnick, 2004). The Template Modelling Score (TM-Score) was developed to address the length dependence of RMSD by scaling the score according to the aligned protein's length. TM-Scores vary from 0 to 1, where a TM-Score below 0.2 corresponds to a randomly chosen unrelated pair of proteins and a TM-Score above 0.5 indicates a high probability that the aligned molecules share the same fold (refer to Section 1.7) (Xu and Zhang, 2010). TM-Score is particularly useful to assess the similarity of proteins with low sequence identity and is the main score used by TM-Align (Zhang and Skolnick, 2005).

The Global Distance Test (GDT-TS) is another measure for protein structure similarity (Zemla, 2003b). The GDT-TS is a measure between the similarity of two structure that share the exact same sequence. For that reason, GDT-TS is commonly used to assess the results of protein structure prediction (Section 1.8). The GDT-TS is calculated based on C_α distances. It is defined as:

$$GDT - TS = \frac{1}{4} \sum_{r=1,2,4,8} Perc(r) \quad (1.1)$$

Where $Perc(r)$ is the percentage of superimposed residue pairs with a C_α - C_α distance $< r\text{\AA}$.

1.6 Quaternary Structure

Quaternary structure refers to the arrangement of two or more polypeptide chains to form a multi-molecular complex. Certain proteins are comprised of only a single polypeptide chain (often referred to as single chain proteins) whereas others are comprised by two or more sub-units, each an independent chain called a monomer, bound in a non-covalent fashion. Proteins comprised of two or more sub-units are referred to as oligomers. Homomers are protein complexes formed by identical monomers and heteromers are complexes formed by different polypeptide chains. All the proteins used in the analyses described in Chapters 2-5 are single chain proteins.

Here, it is necessary to distinguish between a protein chain and a protein domain. A protein domain is a unit that evolves and functions independently of the rest of the protein. A protein chain can contain multiple domains, but domains do not necessarily constitute separate polypeptide chains.

1.7 Protein Classification

Proteins can be classified according to a number of different criteria: protein sequence, protein structure, homology and function. In the next sections, we will discuss several protein classification schemes/databases.

1.7.1 Globular Proteins and Membrane Proteins

Given the importance of hydrophobicity for conferring a protein's structure (Dill, 1990), one of the principal ways to classify proteins relates to the environment that different proteins occupy. In particular, a distinction can be drawn between globular (soluble) proteins, which exist and function on aqueous environment, and transmembrane proteins, which are inserted in hydrophobic environments such as biological lipid bilayers.

Globular and transmembrane proteins present intrinsically different structures (Stevens and Arkin, 1999). They show different propensities for each amino acid (Ulmschneider and Sansom, 2001) and their synthesis and folding are known to occur via different mechanisms (refer to Section 1.9). Given the experimental limitations relating to crystallization of membrane proteins, a majority of the known structures are of globular proteins. In August 2015, approximately 3,000 of the structures on the PDB correspond to transmembrane protein structures ($\approx 3\%$). Given that globular protein structures are more abundant, all of the analyses described in this work were performed using globular proteins.

1.7.2 Pfam

Pfam (Punta et al., 2011) is a database that classifies protein domains into families. Each family in Pfam is represented by three different features: an MSA of a small set of representative members of the family (seed alignment), an automatically generated MSA containing all detectable sequences that belong to that family (full alignment), and a profile HMM built from the seed alignment. Pfam profile HMMs are built using HMMER3 (Eddy, 2011) and publicly available sequence databases.

The latest release of Pfam (release 28.0) contains 16,230 protein domain families. About 81.5% protein sequences available in the UniprotKB/TrEMBL and UniprotKB/Swiss-Prot databases have at least one match on Pfam. According to a recent study (Kamisetty et al., 2013), more than 50% of the protein families in Pfam do not have a member with a known protein structure available in the PDB.

A given protein sequence may have more than one match on Pfam for two different reasons. As mentioned in Section 1.6, a protein chain may present more than one domains, which could result in two or more Pfam families being assigned to non-overlapping sequence regions of multi-domain proteins. The second reason for multiple Pfam assignments comes from the fact that protein families may present a common evolutionary origin. For those

cases, Pfam families are grouped into clans. A Pfam clan can be defined when overlapping parts of the same protein sequence present a significant match to more than one family. Clans are also identified by profile comparison methods such as HHSearch, which detect similarity between the profile HMMs of two different families. These results are manually curated and a clan is established whenever there is sufficient evidence in literature to support common ancestry between the two families (i.e. same function has been reported).

1.7.3 SCOP

Structural Classification Of Proteins (SCOP) is a database that aims to establish structural relationships between all proteins of known structure (Murzin et al., 1995). According to Murzin et al. (1995), existing tools for protein sequence and structure classification cannot detect all structural and, more importantly, evolutionary relationships between different proteins. All proteins in SCOP have been visually inspected and classified into a tree-like hierarchy. Given the lack of automated ways of classifying proteins in SCOP, the latest build available (release 1.75 from February 2009) contains 38,221 PDB entries, which corresponds to $\approx 40\%$ of all available structures on PDB.

Structural Classification Of Proteins Extended (SCOPe) (Fox et al., 2014) has been conceived to address the lack of coverage of SCOP. SCOPe uses a hybrid between manual curation and automatic classification. The latest build of SCOPe (release 2.05 from February 2015) contains 71,105 PDB entries and presents backwards compatibility to SCOP. The hierarchy of SCOPe is, therefore, the same as the one used in SCOP. The major differences between the two databases are the inclusion of novel structures and new protein classifications.

The prototype for SCOP2 (Andreeva et al., 2014) has also been released describing a reduced number of proteins (995 PDB entries, as of February 2015). Unlike SCOPe, SCOP2 is fundamentally different from SCOP. SCOP2 has shifted from a tree-like hierarchical classification scheme to an unrooted directed acyclic graph, which is suggested to

be a better representation for evolutionary relationships between proteins (Andreeva et al., 2014). Given the small coverage of SCOP2, we have refrained from adhering to its new nomenclature/hierarchy and adhered instead to the hierarchy used in SCOP and SCOPe.

The hierarchy in SCOP was designed to encompass evolutionary and structural relationships. There are several classification levels in SCOP (Figure 1.18). At the top of the hierarchy, proteins are separated according to SCOP classes. SCOP classes are based on a protein's secondary structure content and how this content is distributed. Transmembrane and cell surface proteins make a class of their own, for reasons discussed in Section 1.7.1. Small proteins, which present little to no secondary structure, and multi-domain proteins comprise two other SCOP classes. The remaining four SCOP classes comprise a majority of the PDB entries on the database. The SCOP classes All α and All β correspond to proteins that are predominantly α -helical or β -strands, respectively. The classes $\alpha + \beta$ and α/β present α -helices and β -strands segregated (+) or alternated (/).

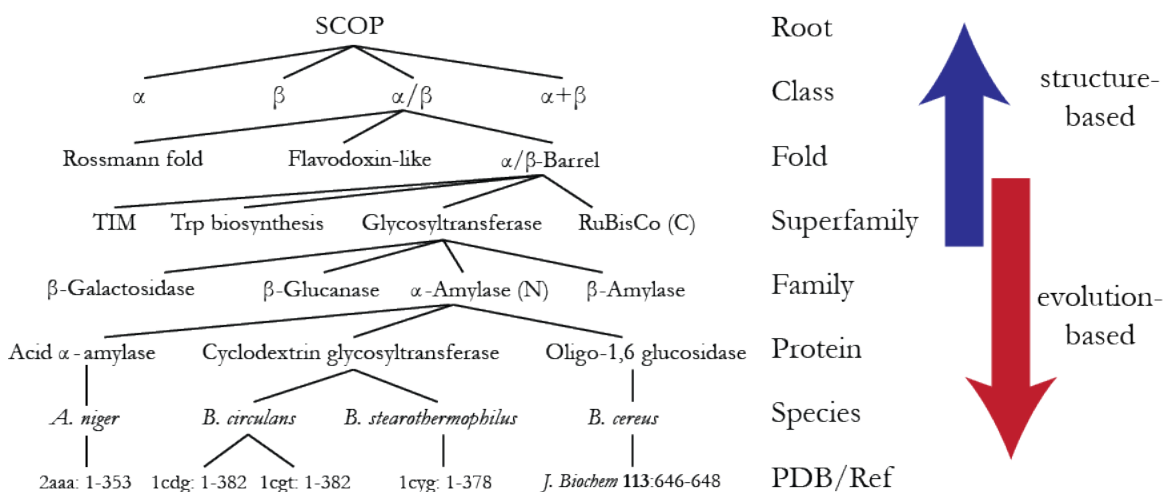


Fig. 1.18 Schematic representation of the hierarchical organization of SCOP.¹⁵

The next level in SCOP's hierarchy is the Fold level. SCOP Folds represent structural similarity that may or may not have a common evolutionary origin. Two proteins are assigned to have the same fold if their secondary structure is arranged in a similar fashion,

¹⁵Image reproduced from Brenner et al. (1996), Copyright © 1996 Published by Elsevier Inc. Permission to reproduce this figure has been granted by the publisher.

conferring a similar protein topology. Proteins in the same fold may differ in terms of the length and conformation of several of their secondary structure elements, as long as their arrangement/topology is maintained.

SCOP Folds are divided into Superfamilies. SCOP Superfamilies can be thought as an analogue to Pfam clans, as proteins are placed in the same SCOP superfamily if they present low sequence identity, but sufficient structure and functional similarities to be thought of as presenting common evolutionary descent. SCOP Superfamilies, in turn, are divided into SCOP Families (not to be confused with Pfam families), where the evolutionary relationship is clear either from sequence or structure information. SCOP families and Pfam families may diverge because the criteria used in the two classifications are significantly different (Pfam uses profile HMMs, whereas SCOP is manually curated based on structural data).

1.8 Protein Structure Prediction

Protein structure determination is expensive and presents technical difficulties for some cases (Terwilliger et al., 2009). The costs for determining protein sequences, on the other hand, are decreasing significantly, and there are fewer limitations to sequence determination compared to structure. Both the number of protein sequences and protein structures are increasing exponentially, but the former is increasing at a much faster pace (Figure 1.19). This is known as the sequence-to-structure gap.

Given the notion that all the information required to determine a protein's structure is contained in its sequence (Anfinsen, 1972), theoretically one should be able to predict a protein's three-dimensional shape given its sequence. Protein structure prediction provides a cheaper alternative to experimental determination and has been proposed as the solution to the sequence-to-structure gap (Schwede, 2013).

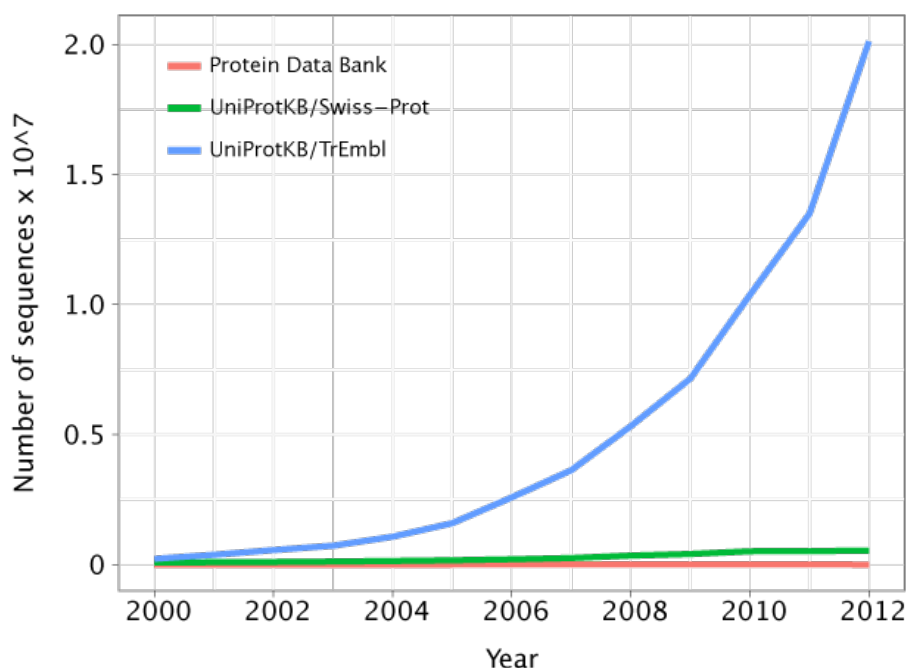


Fig. 1.19 Growth of the number of known protein sequences and structures available on the public domain.¹⁶

1.8.1 A Historical Perspective of Protein Structure Prediction

Some of the first attempts to predict protein structures from their amino acid composition date back from the early seventies (Scheraga et al., 1973; Schulz et al., 1974). These works were based on the concept of a "local structure", which was inferred solely based on the amino acid sequence of peptides. Such developments were fundamental to our understanding of secondary structure and local residue-residue interactions.

It was during the late sixties and early seventies when it was first observed that protein structures are, in general, more conserved than protein sequences (Goldkorn and Schejter, 1976). This feature was exploited by several works that preceded what later became known as homology modelling or template-based prediction (Browne et al., 1969; Kretsinger and Barry, 1975; Rossmann and Argos, 1976).

¹⁶Image reproduced from GORBI: Gene Ontology at Ruđer Bošković Institute under Creative Commons (CC BY-NC-SA 3.0 - <http://creativecommons.org/licenses/by-nc-sa/3.0>)

Yet, at the time, not many protein structures were known and suitable templates were not available for a vast majority of cases. This led to the development of template-free techniques to predict protein structures from sequences. The work by Levitt and Warshel (1975) was one of the pioneering studies in which a near-native structure of a small protein was predicted theoretically without the use of a template. Levitt and colleagues employed an energy minimization approach to infer the fold of a bovine Pancreatic Trypsin Inhibitor (PTI). In their final remarks, the authors state that "such a hierarchical approach might eventually lead to an understanding and simulation of very complicated biological assemblies". Despite their optimism and the many advances in the field since then, the problem of accurate and consistent protein structure prediction still remains unsolved.

The work by Levitt and Warshel (1975) was followed by several other works (Kuntz et al., 1976; Levitt, 1976; Tanaka and Scheraga, 1975, 1976) describing prediction approaches that were based either on the minimization of an energy function (Kuntz et al., 1976; Levitt, 1976) or on Monte Carlo sampling (Tanaka and Scheraga, 1975, 1976). These methods introduced simplified models (coarse-grained) to represent protein chains, such as the use of two centres to represent an amino acid (Levitt and Warshel, 1975) or the use of a single sphere to represent side chains (Tanaka and Scheraga, 1975, 1976). These simplifications were based on an underlying assumption that the fine details of a protein structure are not critical for the formation of the correct fold of the polypeptide chain.

During the late seventies and early eighties, a lot of emphasis was placed on understanding local residue-residue interactions and the prediction of a protein's secondary structure (Argos et al., 1976; Chou and Fasman, 1977; Garnier et al., 1978; Kabsch and Sander, 1983b). Accurate secondary structure prediction was seen as a required step for inferring how each of these local structure elements interacted globally to form protein folds. Given their importance to protein structure prediction, the definitions of secondary structure were formalized and compiled into a dictionary (Kabsch and Sander, 1983a). Secondary structure

prediction continued to be a focus of the protein structure prediction community throughout the eighties; several works published in the late eighties describe the use of neural networks and other machine learning techniques to improve the precision of secondary structure prediction (Cohen et al., 1986; Gibrat et al., 1987; Holley and Karplus, 1989; Kneller et al., 1990; R., 1987). However, it was only in the late nineties when multiple sequence alignments were combined to these machine learning approaches, which culminated in a significant increase in the precision of predictions (Geourjon and Deleage, 1995; Jones, 1999; Rost, 2001).

The late seventies saw the creation of the first dedicated database of protein structures, the Protein Data Bank (Bernstein et al., 1978), and the early attempts of protein structure classification (Chothia and Levitt, 1976). The notorious work by Chothia and Levitt (1976) classified soluble proteins into four major groups (All α , All β , α/β , and $\alpha + \beta$). The work by Chothia and Levitt paved the way for the creation of more elaborate structure classifications schemes such as SCOP (Murzin et al., 1995). Along with these classification schemes came the realization that understanding the hierarchy of protein folds could prove to be beneficial for protein structure prediction (Deleage and Roux, 1987; Klein and DeLisi, 1986; Nakashima et al., 1986).

With the steady increase in the number of protein structures deposited in the PDB, homology modelling methods became gradually more applicable. Several template-based methods were conceived in the late eighties/early nineties (Blundell et al., 1987; Dudek and Scheraga, 1990; ŠAli and Overington, 1994) and were capable of producing good models with some success if a reliable template was identified correctly.

Template-free methods were also continuously developed throughout the late eighties/early nineties (Cohen et al., 1991; Cost and Salzberg, 1990; Hinds and Levitt, 1992, 1994; Samudrala et al., 1999). Lattice based models were introduced (Hinds and Levitt, 1992,

1994; Samudrala et al., 1999) and genetic algorithms became a popular choice for sampling the conformational space (Pedersen and Moult, 1996; Sun, 1993).

Given the wide range of protein structure prediction software available in the early nineties and the complexity of the tasks involved in the prediction process, an experiment was conceived to blindly assess the quality of the predictions of different groups/software. Every two years, an iteration of the Critical Assessment of protein Structure Prediction (CASP) experiment takes place (Moult et al., 2014, 2011a, 1995). CASP is a community endeavour in which blind protein structure prediction is carried out by a multitude of participating groups. The submitted models are later validated by a committee of assessors. The objective of CASP is to assess the ability of the scientific community (the participants of CASP, in particular) to predict protein structures from sequences, given little to no *a priori* knowledge of these structures. CASP results show that predicted protein structures are far from achieving experimental quality for a significant number of cases (Moult et al., 2014). The problem of accurate and consistent protein structure prediction is one of the biggest open problems in biology.

Protein structure prediction can be divided into two distinct categories (Moult et al., 2014): template-based and template-free modelling. These two approaches will be described in more detail in the next sections.

1.8.2 Template-Based Modelling

Template-based methods are based on the observation that proteins that share similar sequences present similar structures (Chothia and Lesk, 1986). These methods use the target protein's sequence to search for a suitable template, i.e. a protein of known structure with a detectable sequence homology to the target. As discussed in Section 1.3, methods based on profile HMMs are the most sensitive for detecting homology relationships and are the methods of choice when searching for suitable templates (Remmert et al., 2012).

A majority of the protein families do not present a suitable template (refer to Section 1.7.2). The remaining modelling cases can be sub-divided according to relative difficulty to identify a good template. Comparative modelling, also known as homology modelling, can be performed for the easier template-based modelling cases, in which a close sequence homologue (>40% sequence identity) with known structure is available. Homology modelling results can achieve similar quality to experimentally determined structures (Moult et al., 2014). It has been reported that homology *in silico* models can be used for virtual docking of small compounds, producing comparable results to experimentally determined structures (Ripphausen et al., 2010).

For the hard modelling cases, the ones where the best detectable template presents low sequence identity to the target, accurate topology can usually be determined, but the models are not as useful as experimentally determined structures. This is often referred to as protein threading or fold recognition. Protein threading operates under the assumption that protein structures are more conserved than sequences and, therefore, that protein structures can still be used as templates despite distant homology relationships being detected. The main distinction between homology modelling and protein threading is the information used to identify suitable templates. In Homology modelling cases, sequence information alone can be used to detect a suitable template, whereas protein threading refers to the cases where profile methods that incorporate secondary and, sometimes, tertiary structure in order to identify a template are used.

Template identification is the most important step for template-based modelling. Multiple templates can be used to generate models, especially in cases where different templates cover different parts of the sequence. Using more than one template to model the same protein region can also be done, but single template methods tend to generate better results (Larsson et al., 2008).

Once a template has been identified, the two sequences are aligned in order to establish a residue-to-residue equivalence between the two proteins. The alignment used to perform template identification can also be used to establish this equivalence. The atomic coordinates of the template are used to define the three-dimensional structure of the target based on the residue-residue mapping given by the alignment.

After an initial model for the target structure has been generated, this model can be assessed according to a score (refer to Section 1.8.8) and further refined (refer to Section 1.8.9). Some methods rely on generating multiple models and selecting the one that is thought to best represent the correct structure. Multiple models can be generated using different templates, different alignments, and by generating models in a stochastic manner.

One of the main categories of CASP is the assessment of template-based modelling. There are several successful template-based methods such as I-TASSER (Zhang, 2008), LEER (Joo et al., 2015) and MULTICOM (Wang et al., 2010). For the latest iteration of CASP, the best ranked group for the template-based modelling category, Zhang (Xu and Zhang, 2012a; Zhang, 2008), produced experimental quality models for seven cases out of 39, presenting a minimum C- α RMSD of 0.52Å and a mean C- α RMSD of 6.59Å.

1.8.3 Template-free Modelling (*de novo* Protein Structure Prediction)

Given that a majority of protein families do not present a member with known structure, template-free modelling has to be performed for most protein families. Template-free modelling, also known as *de novo* protein structure prediction, encompasses all methods in which protein structure prediction occurs without the assistance of a template. Traditionally, *de novo* structure prediction is approached as global optimization problem, in which a scoring function is defined to evaluate how close a given model is to the native structure. The underlying hypothesis is that the native structure will correspond to the global minimum of

this scoring function. Global optimization of the scoring function (finding the minimum of the function), therefore, corresponds to identifying a near-native structure.

CASP results have shown that template-free approaches do not achieve the same success as comparative modelling and, for most cases, an inaccurate topology is predicted (Moult et al., 2014). As mentioned in Section 1.5, the number of possible conformations for a given protein is extremely large. Methods that perform *de novo* protein structure prediction have to address this large number of possible conformations. The other main limitation of *de novo* structure prediction is the fact that a poorly formulated scoring function may lead to a large number of biologically incorrect local minima or to a local minimum conformation that does not correspond to the correct topology.

Many statistical techniques have been employed to ensure efficient sampling of the conformational space and to improve the accuracy of the scoring function. In particular, a divide and conquer paradigm is necessary to tackle the problem of *de novo* protein structure prediction. Predicting a protein structure in the absence of a template is usually performed by solving several smaller problems (divide) such as the prediction of secondary structure and torsion angles, identification of fragments of known structures that present local similarity to the target and prediction of distance constraints from multiple sequence alignments. The solution of each of these individual tasks is then combined (conquest) during model generation to improve protein structure prediction results.

1.8.4 Predicted Secondary Structure

Initial attempts to predict secondary structure from sequence relied on calculating the propensity of each of the amino acid types to adopt each of the secondary structure types (Rost, 2001). A significant increase in precision of predicted secondary structure was obtained by using Multiple Sequence Alignments in conjunction with the amino acid SS propensities (Heffernan et al., 2015; Jones, 1999; Rost, 2001).

Most of the state-of-the-art methods perform machine learning to classify each residue of the target sequence into three main secondary structure types (Helical, Strand, or Coiled) (Faraggi et al., 2012; Heffernan et al., 2015; Jones, 1999; Rost, 2001; Yaseen and Li, 2014). Usually a window is centred around the residue of interest and neighbouring residues and their SS propensities are used as features for the machine learning/classification (Heffernan et al., 2015).

State-of-the-art secondary structure prediction software achieve approximately 80% precision, on average, for a representative subset of known structures (Faraggi et al., 2012; Heffernan et al., 2015; Yaseen and Li, 2014). Given that these methods are based on a multiple sequence alignment, they predict a consensus secondary structure for the proteins in the alignment. The consensus will not fully agree with the particular secondary structure of any individual protein. For this reason, there is an estimate for the upper limit of the precision that these approaches can achieve, at approximately 90% (Dor and Zhou, 2007).

Top SS predictors present very similar precisions (Heffernan et al., 2015), yet recent results have shown that the software SPIDER2 presents slightly higher precision when compared to other approaches such as SCORPION (Yaseen and Li, 2014), PSIPRED (Jones, 1999) and SPINE-X (Faraggi et al., 2009, 2012). There are no results that ascertain whether this higher precision will lead to any improvement in the quality of models generated *de novo*. Our analyses were developed based on PSIPRED and SPINE-X, as SCORPION and SPIDER2 were published in the past year.

Secondary structure prediction software tend to err in the transition regions between secondary structure elements. Intrinsically disordered regions also represent a secondary structure prediction challenge and there are dedicated software packages designed to identify and predict intrinsically disordered regions. Most methods provide a confidence score for the secondary structure predicted for each target residue, which is useful to ascertain which secondary structure predictions are more likely to be correct.

1.8.5 Predicted Torsion Angles

In a similar fashion to secondary structure prediction, torsion angles can be predicted using a combination of propensities for each amino acid types to assume specific torsion angles in conjunction with sequence information from MSAs (Faraggi et al., 2009; Heffernan et al., 2015). The prediction of torsion angles is, however, not as easy as secondary structure prediction because there is a continuous interval of possibilities. Considering the relationship between secondary structure and torsion angles (refer to Section 1.4.3), the prediction of torsion angles is sometimes based on predicted secondary structure, where the latter is used to restrict the possible torsion angles that are considered for predictions (Heffernan et al., 2015).

Several secondary structure prediction software algorithms (Faraggi et al., 2012; Heffernan et al., 2015) also provide torsion angle predictions and a predicted solvent accessible surface for each residue. Interestingly, ϕ angle predictions are more accurate than ψ angle predictions. Torsion angle predictions are not precise enough to be used to reconstruct a protein's topology (Heffernan et al., 2015). Instead, predicted torsion angles can be used to direct sampling during *de novo* model generation.

1.8.6 Heuristics

As mentioned in Sections 1.5 and 1.8, it is not computationally tractable to sample every possible conformation for a given target. Therefore, the problem of *de novo* protein structure prediction relies on the use of heuristic techniques.

Heuristic techniques consist in finding a sub-optimal solution in a feasible time, as opposed to searching for the optimal solution, which would not be tractable for most cases. Most heuristic approaches approximate problems that are NP-Complete or NP-Hard by encountering a sub-optimal solution in polynomial time.

In protein structure prediction, heuristics are commonly employed to perform global optimization, i.e. to find a sub-optimal conformation that corresponds to a local minimum of the scoring function. Many different conformations (referred to as decoys) are usually stochastically generated to explore the combinatorial space. By generating a sufficient number of decoys, one can ensure that a large portion of the conformational space is sampled. Nonetheless, sampling of near-native conformations may be prevented due to a poorly-formulated scoring function or due to constraints imposed by the way each individual decoy is constructed.

1.8.7 Scoring Function

Currently, most *de novo* protein structure prediction software rely on a scoring function to generate models. A scoring function is usually defined as a weighted sum of different potentials, i.e. independent components that can be used to quantify how native-like a given conformation is. Different software employ different scoring functions, which can differ on how individual potentials are implemented, on the weights given to each potential and on which potentials are considered for the scoring function.

The potentials of a scoring function can be subdivided into two types:

1. **Statistical (knowledge-based) potentials:** are based on statistical descriptors derived from all known structures, which are used to describe individual amino acid propensities such as pairwise distances and solvent accessibility. Given that these potentials are derived from known native structures, they might accurately describe the intrinsic properties of these conformations, yet not be sufficient to discriminate between incorrect topologies and near-native conformations (Wang et al., 2004).
2. **Physics-based potentials:** are based on the physical forces that coordinate protein folding (Dill, 1990), such as electrostatics, Lennard-Jones potential, hydrogen bonding

and solvation. These forces are parametrized into a force field, and the combined gradient of these forces can be used to derive a (potential) energy function. Force fields are particularly important for Molecular Dynamics, in which the action of each force is integrated over time to simulate the multiple atoms of a protein as a dynamic physical system.

1.8.8 Model Quality Assessment

As mentioned in Section 1.8.6, many decoys are generated during *de novo* protein structure prediction. In order for protein structure prediction to be informative, it is necessary to assess which of these decoys correspond to accurate topologies and which correspond to misfolded conformations. This process is known as model Quality Assessment (QA).

A perfect scoring function would accurately convey how similar a given decoy is to the native state. Therefore, model quality assessment is only necessary because the scoring functions used in protein structure prediction are imprecise.

There are four distinct approaches to QA: single decoy, clustering (consensus), quasi-single, and meta methods (Kryshtafovych et al., 2014). In single decoy approaches, a scoring function is used to rank all decoys that have been generated. The single decoy with the best score is selected as output. The scoring function that is used for ranking is not necessarily the same as the one used during model generation and it is debatable whether the same or different scoring functions should be used for each step.

Clustering methods rely on clustering all generated decoys based on their pairwise similarity, and then selecting a representative decoy from the largest cluster(s). This representative decoy can be chosen based on the average similarity to all other decoys in the cluster or by means of a scoring function. Clustering methods tend to perform significantly better at CASP compared to single decoy methods (Kryshtafovych et al., 2014). In the way the CASP experiment is designed, up to five models can be submitted for each target. For that reason,

clustering methods tend to select a decoy from five different clusters (maximize divergence) in an attempt to increase their likelihood of selecting a decoy with the correct topology. Yet, this strategy has little to no application in a real case scenario.

A Quasi-single approach generates quality estimates for each individual model, but relies on comparisons made to an ensemble of decoys or to potentially related proteins of known structure to calculate this estimate. The performance of some quasi-single QA methods in CASP10 was superior to the performance of single decoy methods, yet this may have occurred due to the way the datasets in CASP were constructed (Kryshtafovych et al., 2014). The results of clustering and quasi-single QA methods depend on the composition of the decoy ensemble. The decoy ensembles in CASP are sufficiently diverse as models are produced by a wide range of groups and methods, which favours clustering and quasi-single QA over single decoy approaches.

A final QA approach is a meta approach, which relies on identifying the consensus between multiple QA software, merging results generated by different scoring functions, different clustering algorithms and outputting their consensus as a solution.

1.8.9 Model Refinement

Model refinement refers to approaches that aim improve the overall quality of initial protein models, refining these models towards the native structure. There is a multitude of techniques employed to perform model refinement (Nugent et al., 2014), most of which tend to lead initial decoys further away from the native state. Notoriously, the most effective approaches are based on use of force fields and short Molecular Dynamics simulations (Mirjalili et al., 2014). CASP10 was the first iteration of CASP in which a refinement approach, described in Mirjalili et al. (2014), was used to consistently improve the quality of a majority of generated models. Although a consistent improvement was observed, the improvements are fairly small

and refined models are usually closer to the initial model than to the native state (Nugent et al., 2014).

1.9 Translation and Protein Synthesis

Protein synthesis, also known as translation, is the process in which the information contained in a molecule of mRNA gets interpreted and used to construct a polypeptide chain (Figure 1.20). Translation is mediated by a macromolecular complex known as the ribosome, which catalyzes the incorporation of amino acids into a growing peptide chain. The sequence of this peptide chain is determined by the genetic information contained the mRNA. In order to better understand protein synthesis, it is necessary to consider the organization of the genetic code.

1.9.1 The Genetic Code: codons, mRNA, and tRNA

As mentioned in Section 1.1, the central dogma of biology defines a flow of information in which DNA is transcribed to mRNA and then translated into proteins. The information, in this case, is encoded in the composition of DNA and mRNA. In particular, mRNA molecules are polymers formed by multiple copies of four distinct nucleotides. Each nucleotide is comprised of a phosphate group, a sugar group (ribose in the case of RNA) and a nitrogenous base. The four nucleotide types of RNA are characterized by four different nucleic bases: Adenine, Cytosine, Guanine, and Uracil (each represented by the letters A,C,G,U).

The genetic code can be broken down into triplets of nucleotides known as codons. Given that there are four nucleotide types, there are $4^3 = 64$ possible codons. There is a correspondence between codons and the 20 canonical amino acids (Figure 1.21). Given that there are 64 codons and only 20 amino acids, the genetic code is degenerated, i.e. multiple codons may encode for the same amino acid. In a vast majority of species, the genetic code

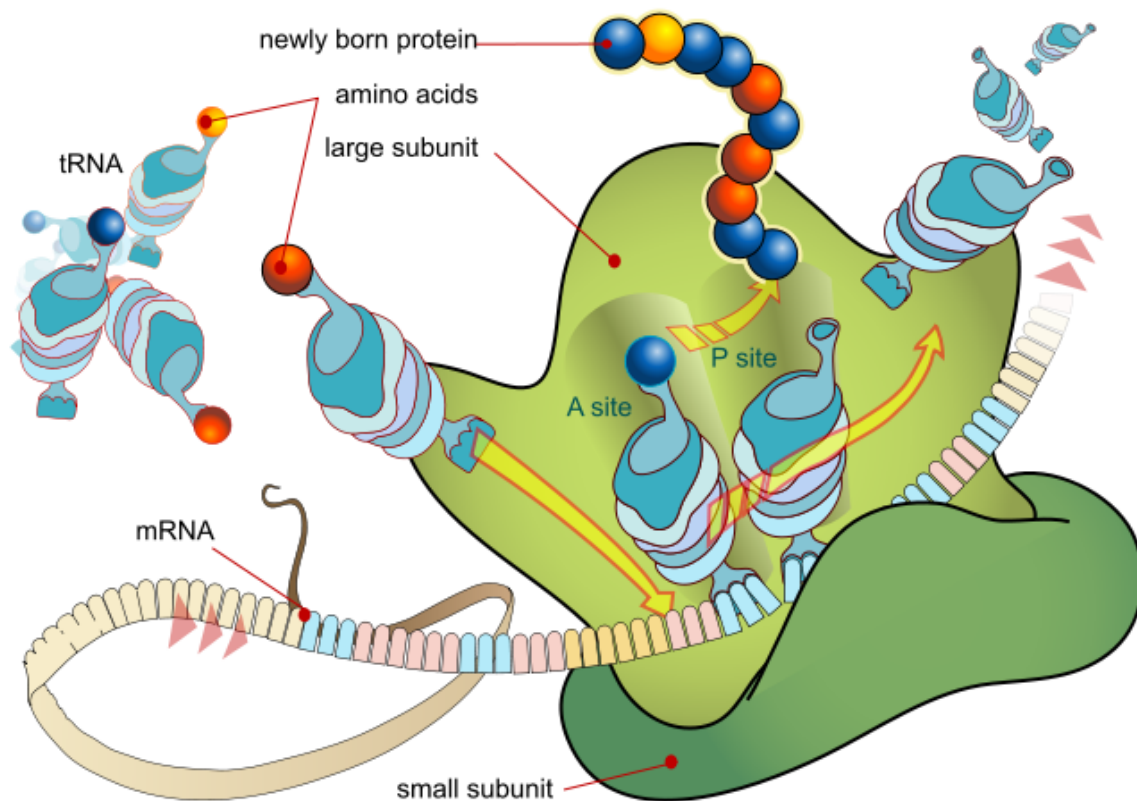


Fig. 1.20 Schematic representation of translation. The ribosome is represented in green, tRNAs in blue, individual amino acids are shown as spheres and the mRNA is shown as an elongated strand.

is also unambiguous; each codon encodes a single amino acid type. Three codons signalize the stop of translation.

Another class of molecules that is fundamental for translation is the Transporter RNA (tRNA). Each molecule of tRNA carries (transports) one amino acid, and the interaction between tRNAs and mRNAs is crucial for protein translation. Just as mRNA, tRNAs also present a triplet of nucleotides that encodes information. The nucleotide triplets of tRNA are known as anti-codons as they interact with the codons in mRNA. This interaction is mediated via hydrogen bonding, where U bonds to A (and vice-versa) and C bonds to G (and vice-versa).

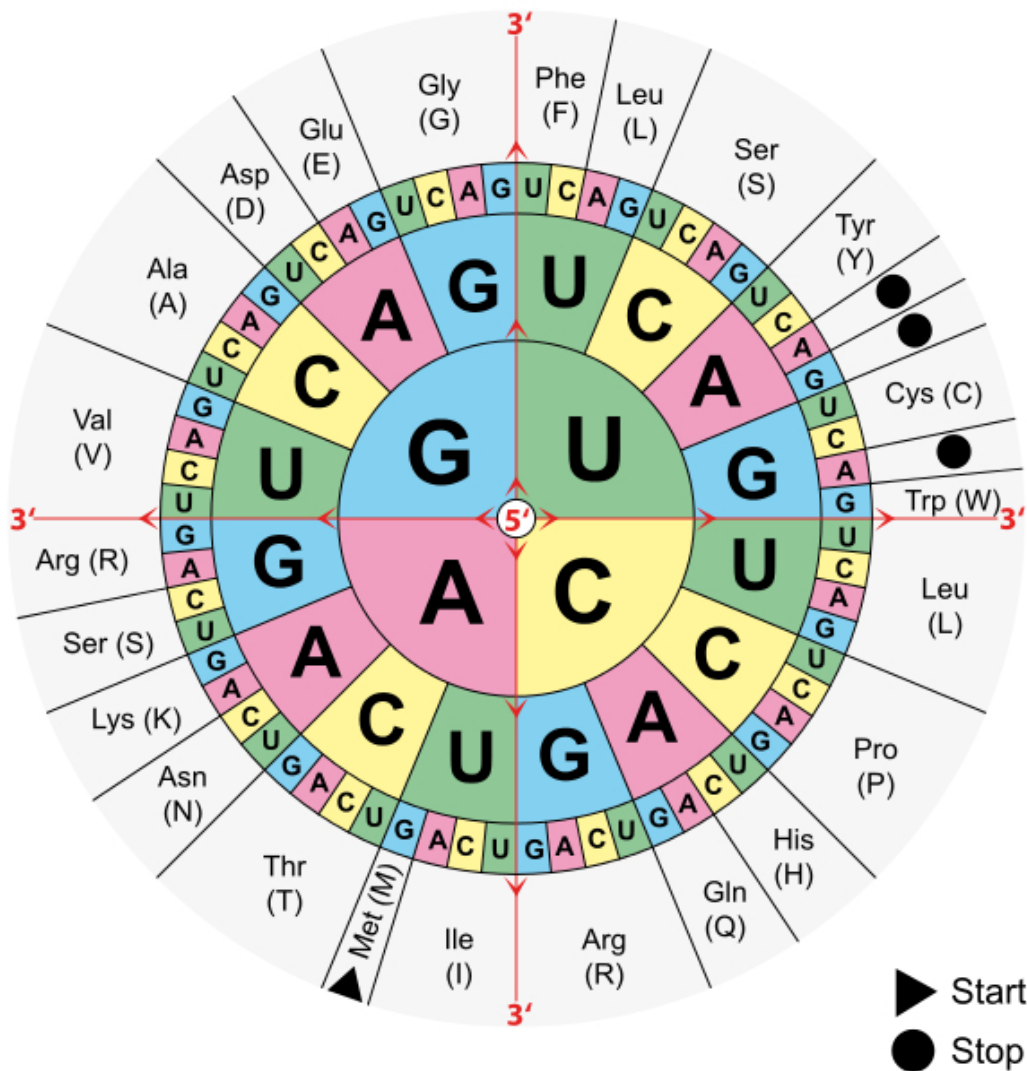


Fig. 1.21 Codons, defined by nucleotide triplets, and the amino acids they encode for. Three codons encode for the stop of translation (stop codons) and 61 codons encode for the 20 canonical amino acids.

1.9.2 Ribosome and the Machinery for Protein Translation

The ribosome is a multi-molecular complex formed by non-covalent interaction of several proteins and ribosomal RNAs (rRNAs) (Nissen et al., 2000). They are divided into two sub-units, a larger unit that presents an exit tunnel from where the growing peptide chain is extruded, and a smaller unit which facilitates the interaction of the complex with mRNA.

Translation occurs when the small subunit binds to a tRNA containing a methionine, which is encoded in the mRNA by the initiation codon, AUG. The small subunit bound to this tRNA interacts with the corresponding initiation codon in the mRNA, which in turn promotes the recruitment of the larger ribosomal unit and the beginning of translation. The larger subunit of the ribosome contains three tRNA binding sites, referred to as A (as it binds to an Aminoacyl-tRNA), P (as it binds to a Peptidyl-tRNA), and E (as it binds to a free tRNA which is about to Exit the translation machinery). The ribosome catalyses the formation of a peptide bond between the peptide attached to the peptidyl-tRNA (the tRNA bound to the growing peptide) and the aminoacyl-tRNA (the tRNA bound to a single amino acid). Once the peptide bond is formed, the previously peptidyl-tRNA becomes free and is displaced to the E site, whereas the previously aminoacyl-tRNA becomes attached to the growing peptide chain, thus changing into a peptidyl-tRNA which gets displaced to the P site. During the catalysis of peptide bond formation, the A binding site becomes free to be bound by the next aminoacyl-tRNA, transporting the amino acid corresponding to the next codon in the mRNA sequence.

Translation begins at the N-terminus and ends at the C-terminus. The growing peptide protrudes, one residue at a time, from an exit tunnel (also known as the ribosome tunnel). The ribosome tunnel is mostly comprised of RNA, but also contains some protein residues specially towards the exit of the tunnel (Nissen et al., 2000). At its narrowest point, the ribosome tunnel can still accommodate α -helices, which has been shown to play a role in secondary structure formation.

1.10 Biologically Inspired *de novo* Protein Structure Prediction

In the following chapters, we will describe our efforts to use biology as an inspiration to improve *de novo* protein structure prediction. In Chapter 2, we explore how to build better fragment libraries for fragment-based protein structure prediction. Our results show that better fragment libraries can be built using predicted secondary structure information and a combination of random and exhaustive searches. This information was used to implement Flib, a novel fragment library generation software. We compared Flib with two fragment library generators and show that Flib libraries present higher precision and coverage and that they can be used to produce better models when compared to the other approaches.

In Chapter 3, we compare several methods that use co-evolution to infer protein residues that are in contact. The contact predictors have been compared in terms of the precision and spread of predicted contacts. Our results show that contact predictors present higher precision for contacts between β -strand residues and lower precision for contacts between α -helical residues. We tested to see which contact predictor is best suited to be used in our protein structure prediction pipeline. We also show how these predicted contacts can be used to ascertain whether a correct model has been produced for a given target.

In Chapter 4, we describe the implementation of SAINT2, a cotranslational *de novo* protein structure prediction program. SAINT2 is able to perform predictions sequentially, starting from the N-terminus. We used SAINT2 to investigate whether the cotranslational protein folding hypothesis (the notion that proteins fold as they are being synthesised) can be used to improve *de novo* protein structure prediction (Chapter 5). Our results show that SAINT2 Cotranslational can produce better models and up to 2.5 times faster when compared to SAINT2 *In vitro*, a traditional non-sequential approach.

Chapter 2

Building a better fragment library for *de novo* protein structure prediction

Software Availability and Authorship: The fragment library generator Flib was fully implemented by Saulo H. P. de Oliveira and is freely available at FigShare - <http://dx.doi.org/10.6084/m9.figshare.1328655>

2.1 Introduction

Fragment-based *de novo* protein structure prediction is the current standard for template-free modelling of proteins. This approach, exemplified by ROSETTA (Bonneau et al., 2002, 2001; Bradley et al., 2005; Raman et al., 2009), relies on creating a library of fragments extracted from known protein structures. In this context, a fragment is a continuous subset of the residues of a protein (Figure 2.1). Usually, such structural fragments are less than 20 residues long. These fragments are used to restrict the conformational space to residue conformations observed in nature.

Each of the fragments in a library are chosen to represent a specific position in the sequence to be modelled (target), as shown in Figure 2.2. The most common technique used

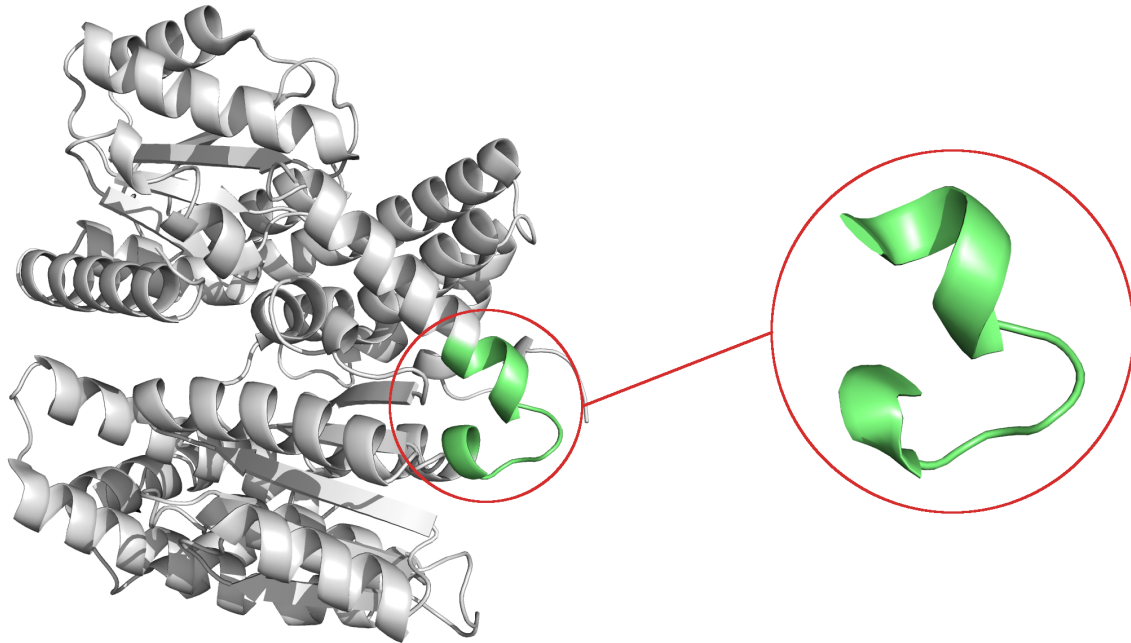


Fig. 2.1 A structural fragment (green) representing a loop conformation of a given protein structure (PDB ID: 1BMT).

to select fragments is the sequence similarity between the fragment and the region of the target sequence that the fragment represents. Fragments from the library are pieced together in order to generate complete models of the target structure. To explore the conformational space, many fragments in the library are chosen to represent each position in the target. For instance, ROSETTA's fragment libraries contain 200 fragments per target position (Raman et al., 2009). The number of possible fragment combinations increases exponentially with protein length. Considering that many proteins are hundreds of residues long, it is not possible to test every fragment combination. In order to explore this large combinatorial space, heuristics are needed (Bonneau et al., 2001). Commonly used heuristics rely on statistical and physical potentials to ensure that global structural features of proteins are sustained/respected in the models generated from the combinations of fragments.

A major problem for all fragment-based *de novo* approaches occurs when the fragment library for a given target does not contain good fragments for a particular region. In that case,

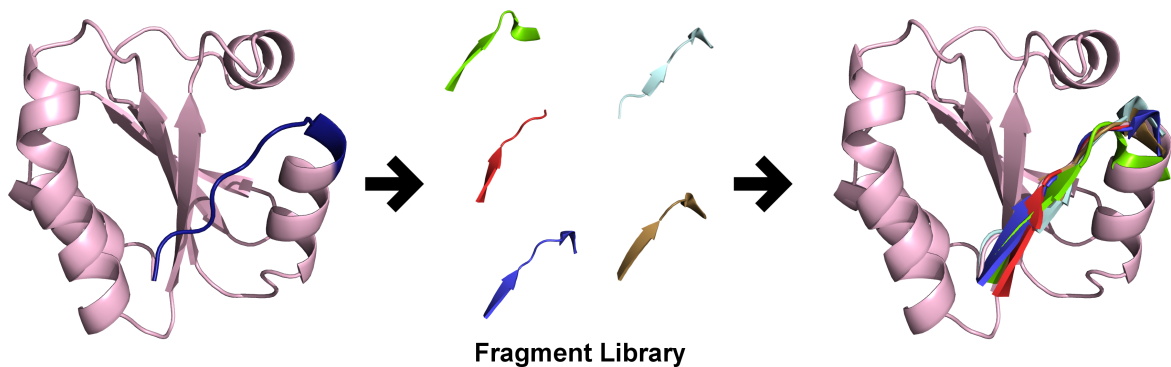


Fig. 2.2 Representation of a fragment library for a given protein structure (PDB ID: 1AIU). Multiple fragments in the library are used to represent a specific position in the target. The fragment library is the set of all fragments used to represent every target position.

low accuracy models will be generated regardless of the precision of the potentials being used and regardless of the amount of computation time invested in the modelling routine (Holmes and Tsai, 2004). For that reason, accurate fragment library generation is crucial to the success of template-free modelling (Holmes and Tsai, 2004).

2.1.1 Building Fragment Libraries

Fragment libraries are built using using features that can be extrapolated from the target's sequence, such as predicted secondary structure, predicted torsion angles and the target's sequence itself. Traditionally, sequence similarity is the main score component used to extract fragments. There is an assumption that fragments with similar sequences tend to share similar structures (Gront et al., 2011; Holmes and Tsai, 2004; Raman et al., 2009). This assumption is based on the same principles behind homology modelling (see Section 1.8.2). In the following sections, we will explore different approaches used to build fragment libraries.

NNMake

NNMake (Gront et al., 2011) is ROSETTA's method for fragment library generation. NNMake extracts fragments from a template database of non-identical high resolution structures.

Structures in this database present high experimental resolution ($<2.5 \text{ \AA}$) and share less than 50 % sequence identity with any other structure in the database. No R-factor cutoff is used. NNMake scores every length nine segment of the target sequence exhaustively against all length nine segments within its template database. NNMake's score is based on sequence similarity, predicted secondary structure and predicted torsion angle. A library for the target sequence consists of the 200 top-scoring fragments per target position. The latest version of NNMake was tested on a set of 62 small globular proteins (all shorter than 150 residues in length). In their test procedure, the authors removed all sequence homologues. In order to assess the quality of generated fragment libraries, ROSETTA was used to generate decoys for each target. The number of decoys generated varied according to target length (ranging from 4,065 to 19,183 decoys). The average coordinate root-mean square deviation (cRMSD) between the native structure of the target and the top 0.1 percentile models were computed (average of the 62 targets cRMSD = $3.75 \text{ \AA}$). One of the current limitations of NNMake relates to the use of fragments with a fixed length (nine residues long). This may not be ideal as it has been shown that accurate structures can be built using fragments as short as four residues (Kolodny et al., 2002). It has also been reported that fragments longer than nine residues can generate better results if the modelling routine is adjusted accordingly (Handl et al., 2012). Other fragment generation software extract either longer fragments or fragments with varying lengths (Hildebrand et al., 2009; Kalev and Habeck, 2011; Lee et al., 2005; Zhang et al., 2005).

Other fragment library generation software also attempt to increase the amount of local structural information used to generate libraries. For example, FRAGFOLD (Jones and McGuffin, 2003) uses super-secondary structure fragments, which express the relationship between consecutive secondary structure elements. FRazor (Li et al., 2008) builds on NNMake and incorporates solvent accessibility and contact capacity into its scoring scheme. The authors of FRazor claim that their scoring scheme improves the precision of NNMake

libraries by discarding low quality fragments suggesting that a sequence-based score can benefit from additional structural information.

HHFrag

HHFrag (Kalev and Habeck, 2011) selects fragments slightly differently from many other methods using profile hidden Markov models (HMM). HHFrag uses the HHpred toolchain (Hildebrand et al., 2009) to build a profile-HMM of the target sequence and a profile-HMM for each sequence in a pre-defined template database. The template database used by HHFrag is the April, 2010 build of PDBselect25 (Griep and Hobohm, 2009), a subset of 4,824 protein chains with less than 25% sequence identity extracted from the PDB (Berman et al., 2000). Sequence and predicted secondary structure information are used in the generation of the profile-HMM. The HMM of the target is divided into a set of overlapping HMM fragments of variable length (6-21 residues). Fragment extraction is performed by HMM-HMM alignment using the HHSearch algorithm Söding (2005). Each of the 6 to 21 residues-long target HMMs is aligned and scored against every HMM profile for the proteins in the template database. All fragments with a probability ≥ 0.2 are accepted. For positions where a minimum of ten fragments are not identified, fragments with lower probabilities are accepted, if possible, until the minimum threshold of ten fragments is fulfilled. HHFrag was tested on a set of 105 proteins (Moult et al., 2011b). The average length of fragments obtained was 10.3 ± 3.6 residues. In order to assess fragment library quality, two measures were defined: precision and coverage. Precision is defined as the proportion of good fragments (RMSD to native structure $< 1.5\text{\AA}$) in the library. Coverage is the percentage of target residues represented by at least one good fragment in the library. HHFrag reports a higher precision ($62 \pm 16\%$) compared to NNMake ($38 \pm 17\%$). However, sequence homologues were not excluded from the HHFrag results, which may inflate the method's precision. HHfrag also reports a coverage

of $71 \pm 13\%$, which is far lower than NNMake ($\approx 92\%$). For some target positions, HHFrag does not output any fragments, which can cause difficulties during the modelling step.

SAFrag

A recent fragment library generation programme SAFrag (Shen et al., 2013) also builds HMM profiles to detect fragment candidates, in an analogous fashion to HHFrag. SAFrag HMM profiles are extrapolated from a 27-state structural alphabet (see Section 1.8.3). The extracted fragments vary in length from 6 to 27 amino acids. Fragments are scored based on a profile-profile comparison, using the Jensen-Shannon divergence (Briët and Harremoës, 2009). Two different template databases can be used by SAFrag: pdb25 and pdb50. Pdb25 is the same database used by HHFrag, whereas pdb50 imposes a 50 % pairwise sequence identity cutoff. The method was validated on a set of 111 targets (Moult et al., 2011b). SAFrag reports a higher precision and coverage than HHFrag. However their cutoff for defining a good fragment is less strict (RMSD to native structure $< 2.0\text{\AA}$). They also allowed homologues and the target structure itself to be included in their template database. Further, the method outputs on average less than two fragments per target position, which suggests that homologue structures are dominating the fragment libraries generated by SAFrag. SAFrag is only available as a web-server.

2.1.2 Building a Better Fragment Library

As described above, methods for fragment library generation differ in the number of fragments used per position, in the length of the fragments used, in the selectivity of the template databases from which fragments are extracted, and in the way the extraction is performed. In our work, we investigated these aspects of fragment library generation and how they affect the precision of the library. We also evaluated the impact of including/excluding homologues among the set of known structures from which fragments can be extracted in order to assess

its impact on methods such as HHFrag and SAFrag. In a real *de novo* structure prediction method, homologues would not be available, so it is important to exclude those fragments during method training and validation.

Current methods score all types of fragments using the same methodology regardless of the predominant predicted secondary structure of the fragment. Here we analysed the relationship between the predominant predicted secondary structure of fragments and our ability to accurately predict fragments for that position. We observed that fragments with a predominant predicted secondary structure (e.g. α -helical fragments) can be predicted more accurately than other types of fragments.

Based on these analyses, we have implemented Flib, a fragment library generation software that exploits the predominant predicted secondary structure of fragments to increase the precision of generated fragment libraries. We have generated fragment libraries for two validation sets: a set of 41 structurally diverse proteins extracted from the PDB (PDB-representative set) and a set of 275 protein domains that were used in CASP9 (Moult et al., 2011a) and CASP10 (Moult et al., 2014) (CASP set). Fragment libraries generated by Flib were compared with libraries generated by NNMake (Gront et al., 2011) and HHFrag (Kalev and Habeck, 2011) and found to obtain the best balance between precision and coverage in both test sets.

Finally, we used the Flib libraries for protein structure prediction, using our custom implementation of a fragment-based protein structure prediction algorithm, SAINT2, which was based on Ellis et al. (2010a). SAINT2 generated accurate (TM-Score to native structure > 0.5) predictions for 12 of the 41 proteins in our test set. We compared our modelling results against running SAINT2 with NNMake fragment libraries. NNMake libraries generated accurate models for 8 cases. Of the 13 cases for which accurate models were generated using SAINT2, Flib libraries generated more accurate models in 10 cases.

These results indicate that Flib can be used to improve the accuracy of *de novo* protein structure prediction and demonstrate the importance of discriminating between different secondary structure fragments when performing fragment extraction.

2.2 Materials and Methods

2.2.1 Data Sets

We have used three different data sets for the analyses described in this chapter. Each of these data sets will be described in the following sections.

Training Set

Our method was trained using a set of 43 structurally diverse proteins extracted from the PDB (Berman et al., 2000). A full list of these proteins is given in Table 2.1. These proteins are all single chain, single domain proteins proportionally distributed into the four SCOP (Murzin et al., 1995) protein classes: all α , all β , α/β , and $\alpha + \beta$. They are also evenly spread in terms of length, ranging from 59 to 508 residues. Each of the proteins in our dataset belongs to a different Pfam family (Punta et al., 2011). Secondary structure for each protein was computed using the software DSSP (Kabsch and Sander, 1983a) and predicted secondary structure was computed using PSIPRED (Jones, 1999). Predicted torsion angles for each protein were computed using the software SPINE-X (Faraggi et al., 2009, 2012).

Table 2.1 Training Data Set

SCOP Class α/β			
PDB ID	Pfam Family ID	Protein Length	Resolution
1H75	PF00462	81 Residues	1.70 Å
1IU9	PF01177	111 Residues	2.04 Å

PDB ID	Pfam Family ID	Protein Length	Resolution
1E6K	PF08924	130 Residues	2.00 Å
1P90	PF02579	145 Residues	1.80 Å
1FTG	PF00258	168 Residues	2.00 Å
1QCY	PF00092	193 Residues	2.30 Å
2A14	PF00072	263 Residues	1.70 Å
1IZZ	PF01965	283 Residues	2.31 Å
1QUE	PF00175/PF00970	303 Residues	1.8 Å
1KRM	PF00962	356 Residues	2.5 Å
3BSG	PF07821	414 Residues	1.95 Å
1PGN	PF00393	482 Residues	2.3 Å
SCOP Class $\alpha + \beta$			
1VJW	PF00037	60 Residues	1.75 Å
1MWP	PF02177	96 Residues	1.8 Å
1GNU	PF02991	117 Residues	1.75 Å
1R9H	PF00254	135 Residues	1.80 Å
206L	PF00959	164 Residues	1.75 Å
2FS3	PF13302	195 Residues	1.52 Å
1DZF	PF01191	215 Residues	1.9 Å
1DXJ	PF00182	242 Residues	1.8 Å
1MAT	PF00557	264 Residues	2.4 Å
1JKS	PF01163	294 Residues	1.50 Å
1MC4	PF01118	370 Residues	2.77 Å
2FKF	PF02879	462 Residues	2.00 Å
SCOP Class All β			

PDB ID	Pfam Family ID	Protein Length	Resolution
1MHN	PF06003	59 Residues	1.8 Å
1TEN	PF00041	90 Residues	1.8 Å
2G1L	PF00498	104 Residues	2.6 Å
1IFR	PF00932	121 Residues	1.4 Å
1BFG	PF00167	146 Residues	1.6 Å
2FR2	PF08768	172 Residues	1.5 Å
1EE6	PF03211	197 Residues	2.30 Å
1UAI	PF08787	224 Residues	1.2 Å
2C9A	PF00047	259 Residues	2.7 Å
1O4Y	PF00722	288 Residues	1.48 Å
1HG8	PF00295	349 Residues	1.73 Å
1NKG	PF09284	508 Residues	1.50 Å
SCOP Class All α			
1AIL	PF00600	73 Residues	1.9 Å
1RRO	PF13202	108 Residues	1.3 Å
1U61	PF00636	138 Residues	2.15 Å
1SL8	PF13202	191 Residues	1.7 Å
1QUU	PF00435	250 Residues	2.5 Å
1T5J	PF03747	313 Residues	2.7 Å
1PO5	PF00067	476 Residues	1.6 Å

The 43 proteins comprising our test data set separated by SCOP classes. Proteins are single-domain, single chain, and belong to distinct PFam families.

PDB-Representative Validation Data Set

Our fragment library generation method was validated using a set of 41 structurally diverse proteins extracted from the PDB (Berman et al., 2000). A full list of these proteins is given in Table 2.2. These proteins are all single chain, single domain proteins proportionally distributed into the four SCOP (Murzin et al., 1995) protein classes: all α , all β , α/β , and $\alpha + \beta$. They are also evenly spread in terms of length, ranging from 54 to 504 residues.

Table 2.2 PDB-Representative Validation Data Set

SCOP Class α/β			
PDB ID	Pfam Family ID	Protein Length	Resolution
1AIU	PF00085	105 Residues	2.00 Å
1NAT	PF00072	124 Residues	2.45 Å
2RN2	PF00075	155 Residues	1.48 Å
1ILW	PF00857	180 Residues	2.05 Å
1VL1	PF01182	232 Residues	1.55 Å
1XWY	PF01026	264 Residues	2.00 Å
2HVM	PF00704	273 Residues	1.80 Å
1OBR	PF00246	326 Residues	2.3 Å
1VFF	PF00232	423 Residues	2.50 Å
1SMD	PF00128	496 Residues	1.6 Å
SCOP Class $\alpha + \beta$			
1WM3	PF00240	72 Residues	1.2 Å
1CEW	PF00031	107 Residues	2.0 Å
1EKG	PF01491	127 Residues	1.8 Å
1Z2U	PF00179	150 Residues	1.1 Å

PDB ID	Pfam Family ID	Protein Length	Resolution
1SQW	PF03657	188 Residues	1.90 Å
1XKR	PF04509	203 Residues	1.75 Å
1W66	PF03099	232 Residues	1.08 Å
1RL0	PF00161	255 Residues	1.4 Å
2YVT	PF12850	260 Residues	1.60 Å
1MSK	PF02965	331 Residues	1.8 Å
1AYE	PF02244	401 Residues	1.8 Å
1B4V	PF00732	504 Residues	1.5 Å
SCOP Class All β			
1CSP	PF00313	67 Residues	2.45 Å
1BMG	PF13895	98 Residues	2.5 Å
1XD6	PF01453	112 Residues	2.0 Å
1NEP	PF02221	130 Residues	1.7 Å
1CZT	PF00754	160 Residues	1.87 Å
1T9F	PF02815	187 Residues	2.00 Å
2AYH	PF00722	214 Residues	1.6 Å
1P6F	PF13895	241 Residues	2.2 Å
1SEF	PF07883	274 Residues	2.05 Å
1WL7	PF04616	312 Residues	1.9 Å
1OKQ	PF00054	394 Residues	2.80 Å
SCOP Class All α			
1ENH	PF00046	54 Residues	2.1 Å
2J9V	PF03997	99 Residues	2.0 Å
2MHR	PF01814	118 Residues	1.3 Å

PDB ID	Pfam Family ID	Protein Length	Resolution
1JWF	PF00790	147 Residues	2.10 Å
1SFE	PF01035	180 Residues	2.10 Å
1SDI	PF04356	213 Residues	1.65 Å
1VIN	PF00134	268 Residues	2.0 Å
1V5C	PF01270	386 Residues	2.0 Å

The 41 proteins comprising our PDB-Representative validation data set separated by SCOP classes. Proteins are single-domain, single chain, and belong to distinct Pfam families.

CASP Validation Data Set

We have also validated Flib on a set of 275 domains that were used in CASP9 (Moult et al., 2011a) and CASP10 (Moult et al., 2014). We have used all domains available from both experiments to compose this validation set.

2.2.2 Homologue Identification

Sequence homologue identification was performed using HHSearch (Söding, 2005). We have used HHSearch with default parameters:

- Database: PDB70_05Jun14.
- Number of iterations = 2.
- E-value cutoff for inclusion in resulting alignment = 0.001.

HHSearch hits with a probability of 99.5% or higher were considered to be homologues.

2.2.3 Template Databases

There are two main criteria used for culling protein structures from the PDB (Berman et al., 2000) when assembling template databases: pairwise sequence identity and resolution. NNMake accepts what it defines as non-identical sequences (50% identity cutoff) whereas HHFrag imposes a stricter cutoff of 25% pairwise sequence identity. NNmake only uses structures with a resolution better than 2.5Å, whereas HHFrag does not impose any resolution cutoff. We built three protein template databases by culling sequences from the PDB: Database Flib, Database NNMake and Database HHFrag. For Database Flib, we removed any protein that presented a resolution worse than 5Å or that presented more than 90% sequence identity to another protein already in the database. For Database NNMake, we used the same selection criteria defined by NNMake (resolution cutoff of 2.5Å and 50% identity cutoff). Database HHFrag used the same criteria as HHFrag, the PDBselect25 (Griep and Hobohm, 2009). We have also compiled four SCOP-specific template databases by splitting Database Flib according to four SCOP class (all α , all β , α/β , and $\alpha + \beta$). Each SCOP-specific database contains all the templates in Database Flib with a given SCOP class.

These databases were further processed: we precomputed the secondary structure for every entry in each of the template databases using DSSP (Kabsch and Sander, 1983a). We classified each residue in all protein sequences into seven distinct groups based on their backbone torsion angles. These seven groups are based on areas of the Ramachandran Plot as defined by Choi and Deane (2010). These areas define the environments for our environment-specific substitution matrices (see Section 2.2.4). Therefore, each entry in a database is represented by three strings: sequence, secondary structure and Ramachandran region identifiers.

2.2.4 Fragment Scores

Three main scores were built and tested for the analyses described in this chapter. All of the scores are defined using a pairwise comparison between fragment and target residues. The fragment scoring schemes will be described in the following sections.

Ramachandran-specific Sequence Score

The Ramachandran plot is a good way of visualizing the torsion angles of multiple residues in a given protein or sets of proteins (see Section 1.2.4). Secondary structure imposes constraints on the torsion angles that a given residue can assume. Therefore, secondary structure confines residues to specific regions of the Ramachandran plot. There are six regions on the Ramachandran plot that are associated with secondary structures and a seventh region can be defined as the remaining area of the plot (in other words, anywhere outside the six canonical regions).

In Choi and Deane (2010), the seven regions of the Ramachandran plot are treated as environments to build substitution matrices. Substitution matrices can be built by analysing the frequency of different aminoacids in a multiple sequence alignment (see Section 1.3.1). These matrices describe the propensity for a given residue to be substituted by another. Environment-specific substitution matrices are built by considering the frequency of aminoacids in a specific environment (i.e. the frequency of aminoacids in α -helical regions). In that way, the environment-specific substitution matrices describe the propensity for a given residue in that environment to be substituted by another. The substitution matrices described in Choi and Deane (2010) incorporate additional torsion angle information when compared to a standard sequence alignment score (i.e. using the BLOSUM62 scoring matrix).

The fragment Ramachandran-Specific Sequence Score is defined as the sum of the individual score of each pair of fragment/target residues. The individual score between a fragment residue and a target residue is given by the corresponding environment-specific sub-

stitution matrix. The environment to be used for the score is determined by the precomputed Ramachandran region identifier, as described in Section 2.2.3. Gaps between the structural fragment and the target sequence are not allowed for the purpose of this score.

Predicted Secondary Structure Score

The Predicted Secondary Structure Score is based on a pairwise comparison between the target fragments' predicted secondary structures as output by PSIPRED (Jones, 1999) to the database fragments' known secondary structure as output by DSSP (Kabsch and Sander, 1983a). Analogously to the Ramachandran-Specific Sequence Score, the Predicted Secondary Structure score is defined as a pairwise sum of the individual score of each pair of fragment/target residues. Instead of using a substitution table to perform the individual scoring of residue pairs, The following scoring scheme was defined to compare the secondary structure alignment:

- Match = 1
- Mismatch = -1
- No gaps are allowed between a structural fragment and the template sequence.

Predicted Torsion Angle Score

In order to evaluate the Predicted Torsion Angle Score, the torsion angles (ϕ , ψ) of every database fragment are computed and compared to the predicted torsion angles for the target fragment as output by SPINE-X (Faraggi et al., 2009, 2012). The Predicted Torsion Angle Score is defined as:

$$Score = \sum_{i=1}^n \sqrt{(\phi_{predicted} - \phi_{fragment})^2} + \sqrt{(\psi_{predicted} - \psi_{fragment})^2} \quad (2.1)$$

Where n corresponds to the length of the fragment being scored. Periodic boundary conditions were enforced by selecting the appropriate angle representation that minimized the differences $(\phi_{predicted} - \phi_{fragment})^2$ and $(\psi_{predicted} - \psi_{fragment})^2$. For instance, if a particular $\phi_{predicted}$ is 170 degrees, it can also be represent as -190 degrees. The most suitable of these representations is selected and used according to the value of $\phi_{fragment}$.

2.2.5 Classifying Fragments According to Secondary Structure

We classified each target position according to its predicted predominant secondary structure into four Secondary Structure (SS) classes: *majority α -helix*, *majority β -strand*, *majority loop* and *other*. For example, if more than half of the residues of a fragment are part of an α -helix, then the fragment is classified as a *majority α -helix* fragment. If a fragment does not have a predominant SS type, we place it in the *other* category.

2.2.6 Fragment Extraction with Flib

Fragments were generated for each of the proteins in our training data set using two extraction methods: random extraction and exhaustive extraction. In all cases, all fragments from homologues to the target were removed (see Section 2.2.2).

Random Extraction

The random extraction method consisted of scoring 5,000 randomly selected fragments per target sequence position from a template database. The length of each fragment was randomized to be between six to 20 residues. Each fragment was scored according to the Ramachandran-Specific Sequence Score and to the Predicted Secondary Structure Score. Every fragment is accepted depending on whether its scores satisfy an acceptance cutoff. We have selected the cutoffs that achieve the best precision whilst maximizing the coverage (Figure 2.3). For a detailed description of precision and coverage, refer to Section 2.2.7.

Different cutoffs were determined and used within each of the fragment SS classes described in Section 2.2.5. The score cutoffs are detailed in Table 2.3.

Table 2.3 Flib Score Cutoffs for Random Extraction

	Ramachandran-Specific Sequence Score	Predicted Secondary Structure Score
<i>Majority α-helix</i>	0	2
<i>Majority β-strand</i>	0	2
<i>Majority loop</i>	-6	0
<i>Other</i>	-6	0

The library resulting from random extraction presents, on average, 2,000 fragments per target position.

Exhaustive Extraction

For the exhaustive extraction, all possible fragments from a template database were scored against every position in the target. Analogous to the random extraction, fragments were scored based on the Ramachandran-Specific Sequence Score and the Predicted Secondary Structure Score. The top 1,000 scoring fragments are selected for each target position as we found that the precision was not increased by the inclusion of more fragments.

The top 1,000 fragments per target position obtained by exhaustive extraction are merged with the 2,000 fragments per target position obtained by random extraction. The resulting fragment library presents approximately 3,000 fragments per target position (LIB3000).

Fragment Library Enrichment

For each target position, we rank the fragments in LIB3000 according to the Predicted Torsion Angle Score and select the top 20 highest scoring fragments per target position (LIB20). We further enrich LIB20 by including any fragment from LIB500 (analogous to LIB20) that presents less than 0.5Å RMSD to the highest scoring fragment per target position.

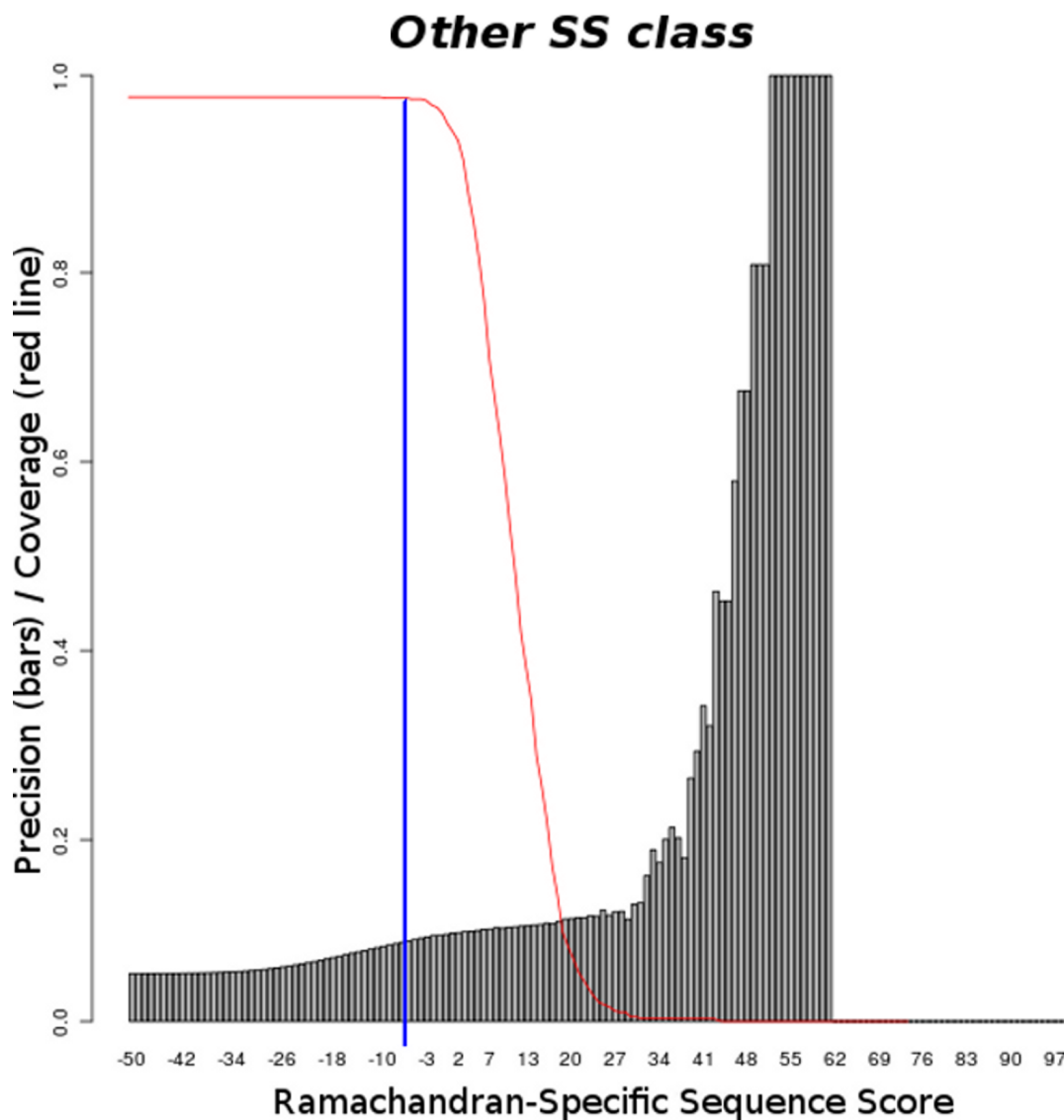


Fig. 2.3 We have varied the Ramachandran-Specific Sequence Score cutoff (x-axis) for accepting fragments and assessed the effect of the cutoff on the precision (y-axis - bars) and the coverage (y-axis - red line) within each SS fragment class. In this figure, the precision and coverage are shown for all *Other* SS class fragments and are averaged over the 43 proteins in our training set. We select the cutoff (blue line) that maximizes precision while maintaining coverage as close as possible to 100% .

Protein Threading Hits Library

In the final step of our routine, we perform protein threading using the target sequence as input to HHSearch (Söding, 2005). Default parameters for HHSearch were used to perform protein

threading. Protein threading hits that originated from homologues, as described earlier, are removed from HHSearch's output. We extract every possible nine-residue fragment from the remaining threading hits (Protein Threading Library). The fragments in the Protein Threading Library are ranked according to the hit score output by HHSearch. We select a maximum of 20 fragments per target position. Fragments belonging to *Majority α -helical* positions are removed from the Protein Threading Library in a post-processing step. All fragments in the Protein Threading Library are added to LIB20 to generate the final output of Flib. This final library presents, on average, ≈ 33 fragments per target position.

2.2.7 Validation

We assessed fragment library quality using two commonly employed metrics: global precision and coverage. Precision is defined as the number of good fragments divided by the total number of fragments in a library (the proportion of good fragments in the libraries). Coverage is defined as the number of residues represented by at least one good fragment divided by the number of residues of the target (the proportion of protein residues represented by a good fragment).

The quality of fragments was assessed by superimposing the fragment on to the target's known structure. Different methods employ different cutoffs to distinguish between good fragment conformations and bad fragment conformations. Instead of selecting a single cutoff, we have varied the good fragment cutoff between 0.1 to 2.0 Å computing the precision and coverage across the range. Fragments with an RMSD to the native structure below this varying cutoff are considered to be good fragments (Figure 2.4).

HHFrag

In order to generate fragment libraries using HHFrag, we have used HHFrag v2.1 with default parameters.

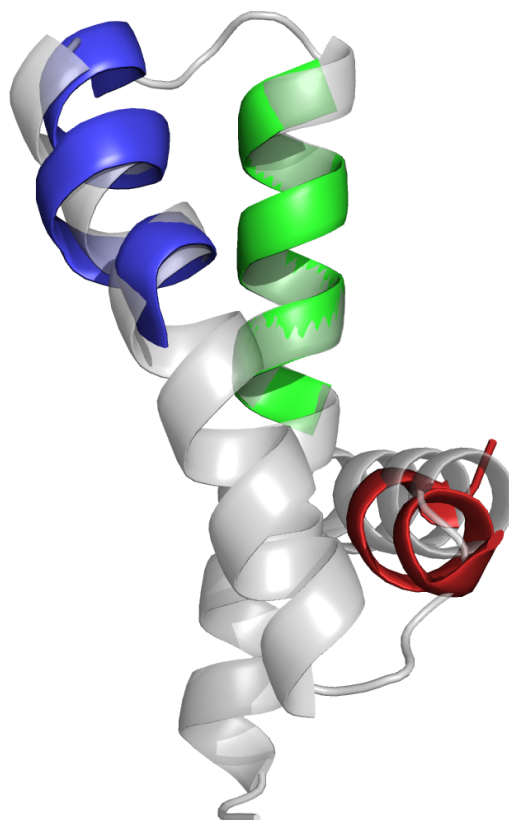


Fig. 2.4 Representation of different fragments superimposed to the structure of protein 1AIL. Good fragments (green) present low RMSD to the native structure. Certain fragments (blue) may be considered good fragments by more lenient methods such as SAFrag, but not by more stringent methods such as NNMake. Bad fragments (red) fail to represent the native structure correctly.

NNMake

We have used NNMake from the Build 3.5 of MiniROSETTA. In order to generate the fragment libraries, default parameters for NNMake were used.

2.2.8 Model Generation

We have generated 1,000 decoys for every protein in our PDB-Representative Validation set using two different approaches: Flib's fragment libraries with SAINT2 and NNMake's fragment libraries with SAINT2. SAINT2 is our custom implementation of a cotranslational *de novo* protein structure prediction software (see Chapter 3).

2.3 Results

2.3.1 The Rationale Behind Flib

Flib extracts fragments from a database of known structures using a target sequence. A framework describing Flib's pipeline is shown in Figure 2.5. The first step in our routine is to exclude homologues of the target sequence from the template database. The target sequence is then used to predict the target's secondary structure and torsion angles. Along with the target's sequence, these three features are used to score fragments on the template database. Fragments are extracted using a combination of random and exhaustive methods. The resulting fragment ensemble is enriched using our Predicted Torsion Angle score. The final step in Flib is to further enrich the final library with non-homologue fragments from protein threading hits, as output by HHSearch (Söding, 2005).

2.3.2 Building Better Template Databases

The template database, the initial set of structures from which fragments for libraries will be extracted, can be built in many ways. Different methods differ in the sequence identity and resolution cutoffs used to cull sequences from the PDB (Gront et al., 2011; Kalev and Habeck, 2011; Shen et al., 2013). Regardless of how the template database is built, for testing

purposes it is important to remove homologues of the target, which is not often performed by existing approaches (Kalev and Habeck, 2011; Shen et al., 2013).

To understand how to construct better fragment libraries, we have compared three different template databases and how they affect the precision and coverage of generated libraries (Figure 2.6). Database NNMake and Database HHFrag are based on the databases in Gront et al. (2011) and Kalev and Habeck (2011) respectively (see Section 2.2.3 for more details). For Database Flib, we impose a 90% sequence identity cutoff and a 5.0Å resolution cutoff. No significant difference was observed in the precision and coverage of fragment libraries generated by different databases and we chose Database Flib as our database of choice.

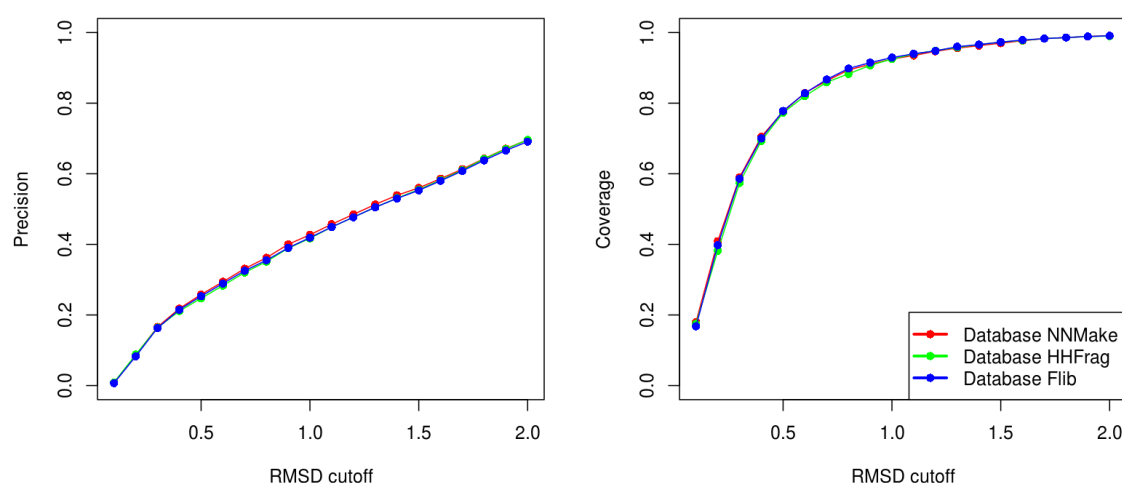


Fig. 2.6 To assess the effect of using different template databases to build fragment libraries, we analysed the precision of LIB20 built using three different template databases: Database Flib, Database NNMake and Database HHFrag. We varied the fragment RMSD to native structure cutoff that defines a good fragment from 0.1 to 2.0Å (x-axis). The precision indicates the proportion of good fragments in the generated libraries (y-axis) and it was averaged on the 43 proteins of our Training Data Set.

SCOP-specific Template Databases

We have also investigated the impact of using SCOP-specific template databases for fragment library generation. For this analysis, we divided Database Flib into four SCOP-specific databases. Each SCOP-specific template database contains structures belonging exclusively to one of four SCOP classes ($\alpha + \beta$, α/β , All β , and All α). We compared the precision and coverage of LIB20 built using Database Flib against the same library built using one of the SCOP-specific template database according to the target's SCOP class (Figure 2.7). Results are shown for the 43 proteins in our Training Data Set. In general, using Database Flib produces marginally better results.

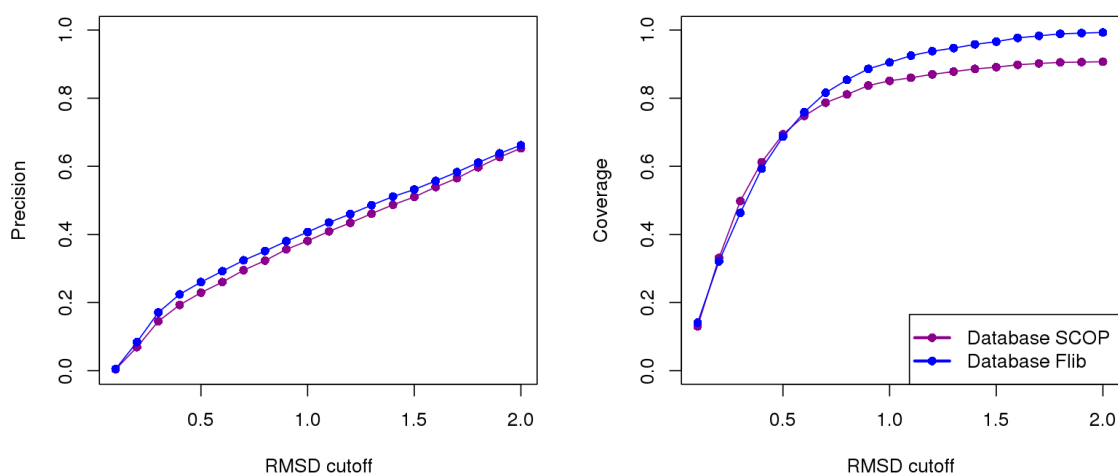


Fig. 2.7 To assess the effect of SCOP-specific template databases to build fragment libraries, we analysed the precision and coverage of LIB20 built using Database Flib (blue) against the same library built using SCOP-specific template databases, according to the target's SCOP class (purple). We varied the fragment RMSD to native structure cutoff that defines a good fragment from 0.1 to 2.0Å (x-axis). The precision and coverage were averaged on the 43 proteins of our Training Data Set.

2.3.3 Building a Better Scoring Scheme for Fragments

In order to perform *de novo* protein structure prediction where the target structure is not known, sequence-related scores are used to predict good quality fragments.

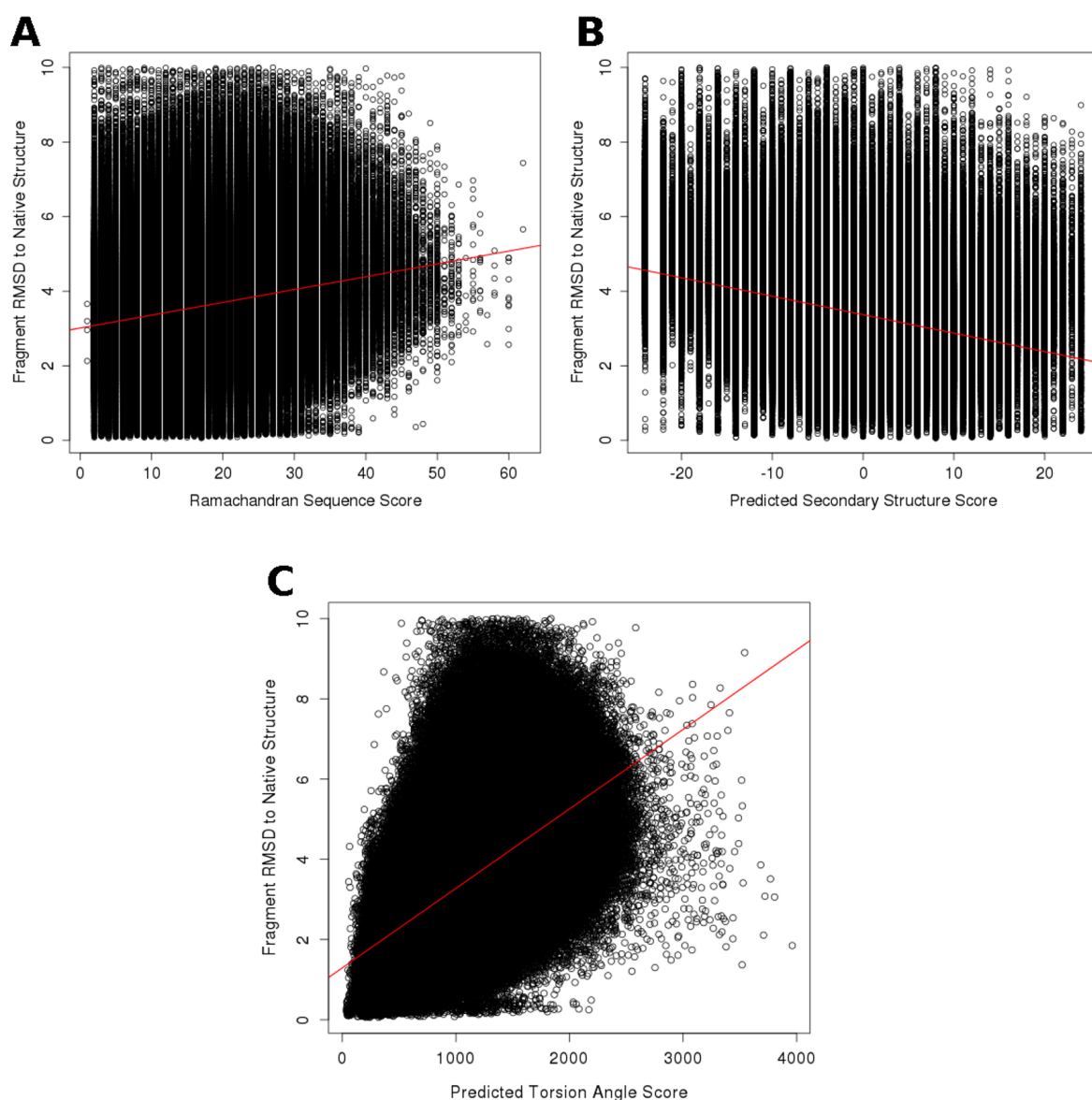


Fig. 2.8 Fragment RMSD to native structure against three different scores: Ramachandran-Specific Sequence Score (A), Predicted Secondary Structure Score (B), and Predicted Torsion Angle Score (C). Results are shown for 1,000 fragments extracted at random for each of the 43 proteins in our Training Data Set.

We developed three different scoring schemes to rank fragments using target sequence-related features: Ramachandran-Specific Sequence Score, Predicted Secondary Structure Score and Predicted Torsion Angle Score. We assessed the potential of each of these scores to predict fragment quality in our Training Data Set (Figure 2.8). We observed little to no correlation between the scores and the fragment RMSD to native structure for an ensemble of fragments extracted for each of the 43 proteins in our Training Data Set. In general, fragments with poor scores tend not to represent the native structure correctly. We also observed a higher correlation between the Predicted Torsion Angle Score and fragment RMSD to native structure as compared to the other two scoring schemes, because of this we opted to use Predicted Torsion Angle Score to rank the fragments in our fragment libraries.

2.3.4 Improving Fragment Extraction

After building an appropriate template database and defining a scoring scheme, the next step in a fragment library generation pipeline is fragment extraction. Most fragment library generation methods use an exhaustive search approach (Gront et al., 2011; Jones and McGuffin, 2003; Kalev and Habeck, 2011; Li et al., 2008; Shen et al., 2013). Given the poor (or lack of) correlation observed between the fragment scoring schemes and fragment quality, we compared the use of an exhaustive approach against a random approach to extract fragments. In our exhaustive approach, every fragment ranging from 6 to 20 residues in the template database is scored against all of the positions of the target. The exhaustive library is composed of the top 1,000 scoring fragments per target position.

In our random approach, 5,000 randomly selected fragments are scored for each target position. Fragments that satisfy a predetermined score cutoff are accepted and fragments that do not are discarded (see Section 2.2.6 for more details). On average, the random library contains 2,000 fragments per position. Surprisingly, we observed that fragment libraries extracted at random present slightly higher precision and coverage than the ones generated

exhaustively (Figure 2.9). Our results suggest that random extraction produces better results because above a certain score threshold, we observe no correlation between the scores and fragment quality (Figure 2.8).

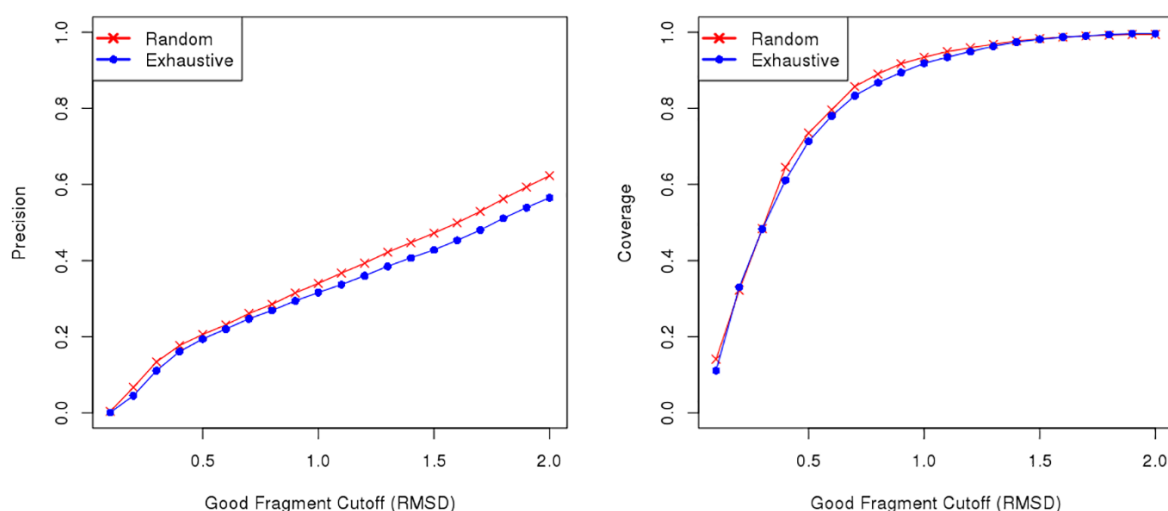


Fig. 2.9 Comparison between Flib's random extraction and exhaustive extraction methods. Analysis of the precision and coverage of fragment libraries generated by Flib using two different approaches for fragment extraction: random extraction (red), and exhaustive extraction (blue). We varied the RMSD to native structure cutoff that defines a good fragment from 0.1 to 2.0 Å (x-axis). The average precision on the 43 proteins in the Training Data set (left) and the average coverage (right) are shown for fragment libraries containing the top 1000 scoring fragments extracted exhaustively or at random.

Within Flib's pipeline, we combine the exhaustive and random libraries. This combined library has, on average, 3,000 fragments per target position (LIB3000). We rank all the fragments in this library and output the 20 top-scoring fragments per target position according to the torsion angle score (LIB20). On average, within our test data set, 69% of the fragments in LIB20 are extracted by the random method and 31% by the exhaustive protocol.

2.3.5 Enriching Fragment Libraries

We observed that a fragment ensemble containing only the highest scoring fragment per target position according to the Predicted Torsion Angle score presented a very high precision, albeit at a loss of coverage. We decided to exploit this finding by implementing an enrichment

step within Flib. We include fragments from LIB500 (analogous to LIB20, but considering the 500 top-scoring fragments) that present less than 0.5Å RMSD to the highest scoring fragment for each target position according to the Predited Torsion Angle Score. On average, 6.5 fragments per position are added to LIB20 during the enrichment step.

2.3.6 Protein Threading Hits Library

The final step in our fragment library generation routine is to add to LIB20 non-homologous fragments extracted from protein threading hits, as output by HHSearch. Protein threading identifies protein segments that present structural similarity to a given target (refer to Section 1.8.2). As we remove homologues, these protein threading hits are too unreliable to be used as templates for template-based modelling, but may still provide locally similar fragments. We have found that adding such fragments, when they are available, to LIB20 increases the precision of our method (Figure 2.10). On average, less than five Protein Threading Hits fragments per position are added to LIB20.

We have also assessed the impact of adding Protein Threading Hits fragments to the precision within each of our Secondary Structure fragment classes (see Section 2.2.5) in our Training Data set. Results are shown for each of the four Secondary Structure classes on Figure 2.11. Fragments from Protein Threading Hits present significantly higher precision in the *Majority β -strand* and *Other* Secondary Structure classes, whereas LIB20 presents significantly higher precision for the *Majority α -helical* Secondary Structure class. For that reason, we do not include fragments from Protein Threading Hits that fall into the *Majority α -helical* Class in Flib's fragment libraries.

2.3.7 Secondary Structure Classes Determine Fragment Quality

Secondary structures (α -helices and β -strands) have restrictions by definition in the torsion angles of their residues, whilst loop regions are not so constrained and can assume a wider

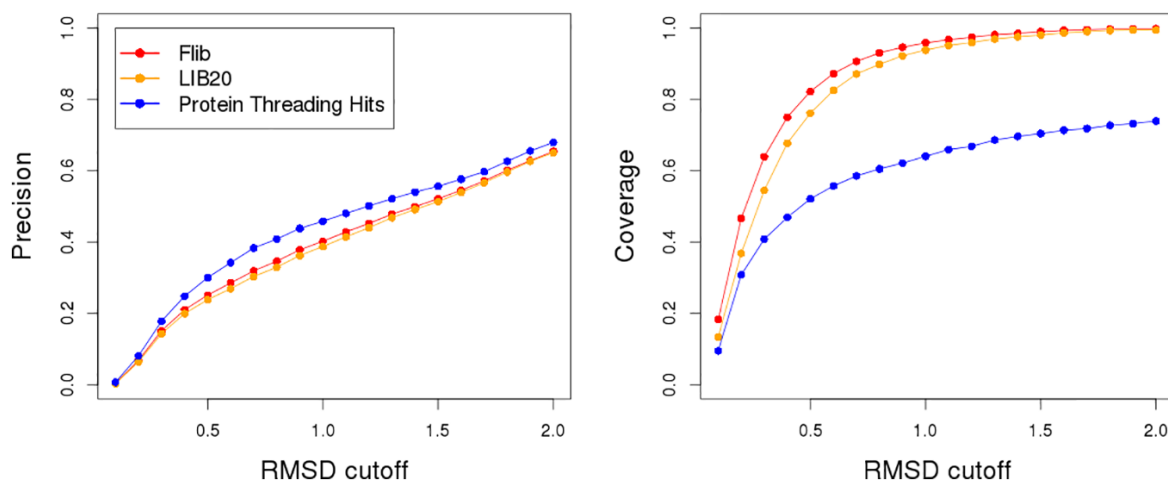


Fig. 2.10 Effect of protein threading hits on fragment library quality. Analysis of the impact of fragments extracted from protein threading hits. Precision and coverage are shown for the fragment libraries generated by LIB20, Protein Threading Hits and Flib (a combination of the other two approaches). We varied the RMSD to native structure cutoff that defines a good fragment from 0.1 to 2.0 Å (x-axis). The average precision and coverage on the 43 proteins in our Training Data set are shown for each approach.

range of conformations (Choi et al., 2013). We hypothesized that fragments with a larger number of loop would be harder to predict. In order to test our hypothesis, we have investigated the relationship between the fragment RMSD to native structure and the predominant predicted secondary structure of fragments.

Here a fragment is described as representing a target position N when it represents all the target residues between N and $N+L$ (where L is the length of the fragment). We classify a target position as belonging to one of four distinct classes based on the predominant predicted secondary structure of its residues. The four classes of secondary structure (SS classes) are: *Majority α -helical*, *Majority β -strand*, *majority loop* and *other* (no predominant predicted secondary structure). For more details, refer to Section 2.2.5.

We analysed the RMSD to the native structure of fragments extracted at random. The fragments were grouped according to our predicted SS classes. The spread of fragment RMSD for the top 200 scoring fragments is shown for every position in the target 1E6K (Figure 2.12). The RMSD spread of the top-200 fragments replicate the results obtained with

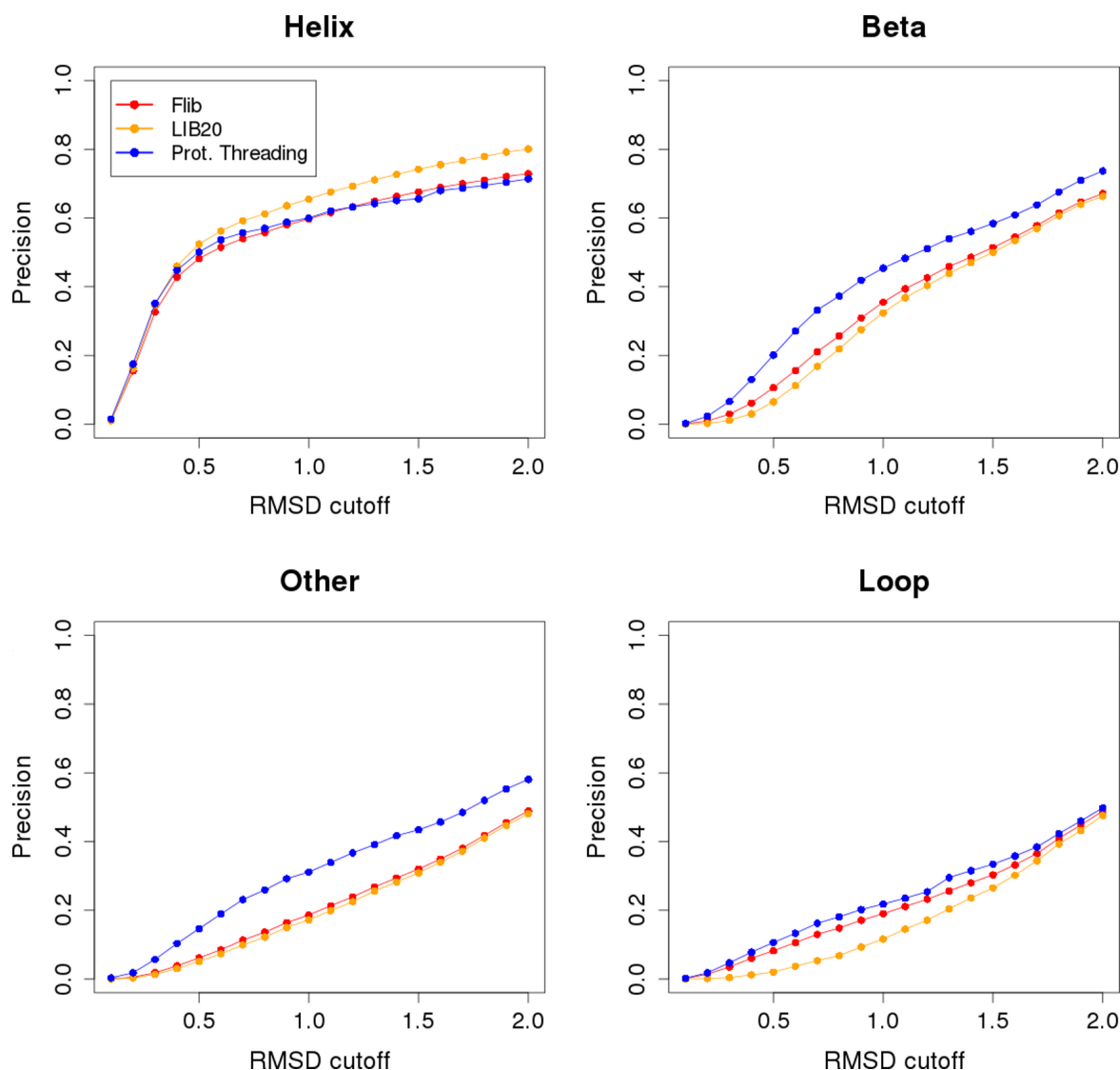


Fig. 2.11 Analysis of the impact of fragments extracted from protein threading hits on the precision within each Secondary Structure fragment class. Precision is shown for the fragment libraries generated by LIB20, Protein Threading Hits and Flib (a combination of the two previous approaches). We varied the RMSD to native structure cutoff that defines a good fragment from 0.1 to 2.0 Å (x-axis). The average precision within each SS Class on the 43 proteins in the Training Data set are shown.

LIB3000. This figure typifies our general observation that there is a strong relationship between the fragment RMSD to native structure and our four SS classes. The RMSDs for *Majority α -helical* fragments are significantly lower than the RMSDs for other SS classes. *Majority Loop* fragments and *Other* fragments show a wider variability and are much harder

to predict accurately. The difference between SS classes is important in two ways. Firstly, as current methods only offer coverage and precision across all SS classes, very poor *Majority Loop* and *Other* precision may be hidden by high *Majority α -helical* precision. Secondly, these results suggest that during fragment library generation, it may improve results if we treat fragments differently according to their predominant predicted secondary structure. For that reason, Flib uses different cutoffs for accepting fragments based on SS class. Less stringent cutoffs for Majority Loop and other fragments are used as their variability is far higher. We have observed that adopting different cutoffs for each SS class improves the precision of the libraries generated by Flib.

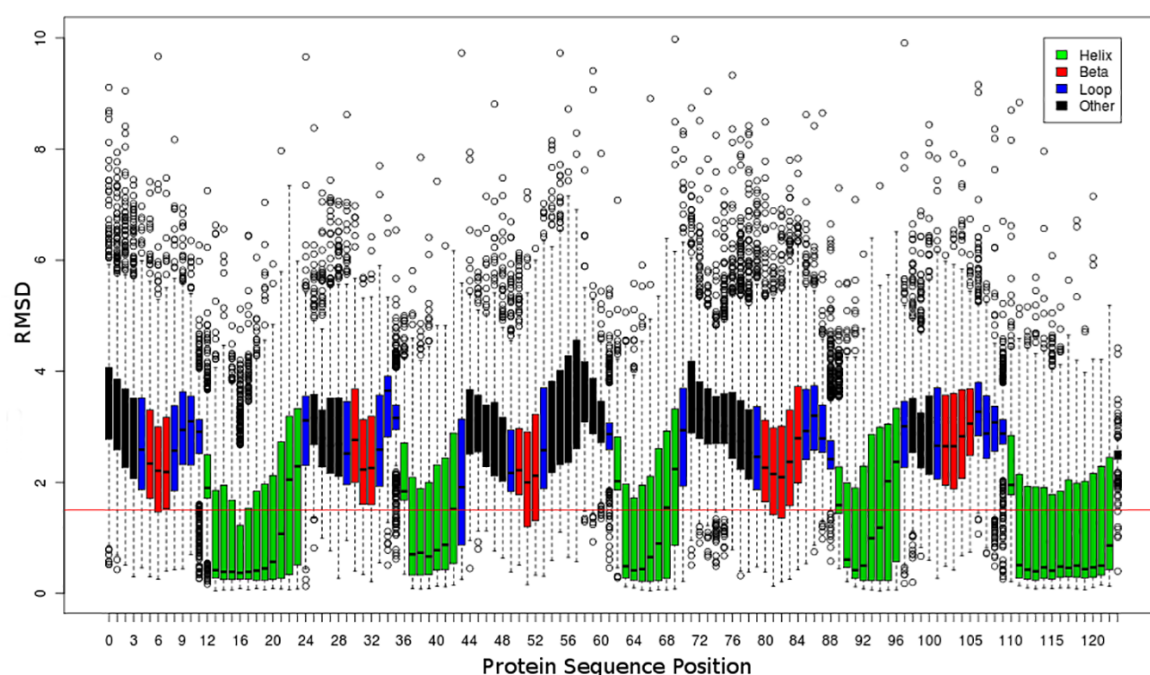


Fig. 2.12 Relationship between secondary structure class (SS-Class) and fragment quality. Boxplot of the RMSD to native structure (y-axis) of 200 fragments per target position (x-axis) for the protein 1E6K. The top-200 scoring fragments from LIB3000 were selected and are displayed. This subset of LIB3000 was chosen to increase performance of data visualization. Four Different SS Classes are defined: *Majority α -helical* (green), *Majority β -strand* (red), *Majority Loop* (blue) and *Other* (black). Positions for which fragments are *Majority α -helical* or *Majority β -strand* present significantly lower RMSDs to the native structure and a smaller spread compared to *Majority Loop* and *Other* positions.

2.3.8 Cross-comparison Between Flib and Other Software

We have compared Flib against NNMake and HHFrag (Figures 2.13, 2.14, 2.15, 2.16). A large scale comparison including SAfrag could not be performed because the software is only available as a web-server. In order to perform a rigorous comparison, we have used two distinct validation sets: a set comprised of 275 protein domains from CASP9 and CASP10 (CASP Validation Data set) and a set of 41 structurally diverse proteins (PDB-Representative Validation Data set). The second set was built to be representative of the PDB, both in terms of protein lengths and distribution of proteins amongst different SCOP classes (refer to Section 2.2.1 for more details).

In all analyses, fragments extracted from homologues were discarded (the impact of filtering out fragments from homologues is described in the next section). If we compare the overall precision and coverage, Flib presents higher precision compared to NNMake and higher coverage compared to HHFrag (Figures 2.13, 2.15). This difference appears to be due to the increase in performance for the SS Classes *Majority α -helical* and *Majority β -strand* (Figures 2.14, 2.16). HHFrag coverage is significantly lower than that of the other two methods. At a 1.0 Å RMSD cutoff, HHFrag's fragment libraries describe slightly more than half of the target residues correctly ($\approx 55\%$ coverage on PDB-representative set, $\approx 65\%$ coverage on the CASP set). HHFrag failed to produce any fragments for $\approx 13\%$ of the positions. This can become a problem during structure prediction considering that modelling routines generally require at least one fragment representing every target position.

PDB-Representative Validation Set

When comparing the three programmes using the PDB-Representative Validation Set, Flib achieves the best balance between coverage and precision. Considering a good fragment cutoff of 1.0 Å, HHFrag presents the highest average overall precision, $\approx 43\%$, compared to Flib, $\approx 35\%$, and NNMake, $\approx 29.1\%$. But HHFrag's precision is increased due to a reduced

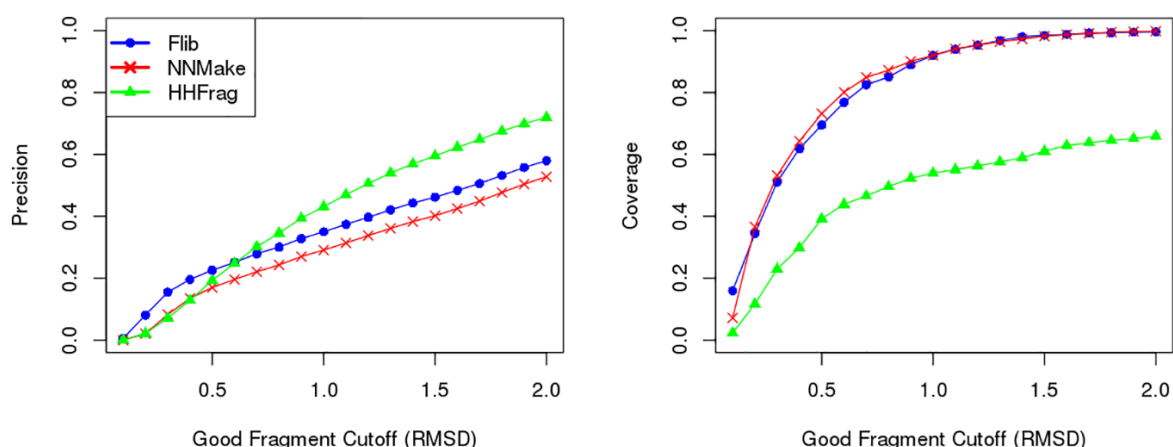


Fig. 2.13 Comparison between HHFrag, NNMake and Flib. Precision (left) and coverage (right) of fragment libraries generated using NNMake (red), HHFrag (green) and Flib (blue). The precision and coverage of the fragment libraries are averaged on a set of 41 structurally diverse proteins (PDB-Representative). We varied the RMSD cutoff that defines a good fragment (x axis) and evaluated the precision (proportion of good fragments in the libraries) and coverage (proportion of protein residues represented by a good fragment) for each method.

number of fragments output per position (see below). HHFrag also boosts its precision by not outputting any fragments for regions that are harder to predict (as stated above, on average, $\approx 13\%$ of the residues are not represented by any fragment in an HHFrag generated library). Not outputting fragments for low confidence regions will improve precision, but will also cause difficulties during protein structure prediction.

On the PDB-Representative Validation set, Flib outputs, on average, 26 fragments per position, with an average length of ≈ 7.4 residues. HHFrag outputs on average 10 fragments per position, with an average length of 9.1 residues. Generating a smaller number of fragments can improve precision, but can represent a problem during modelling since less conformations will be sampled. NNMake always outputs 200 fragments per position with a constant length of nine residues.

The three methods perform well at predicting fragments for *Majority α -helical* fragments, however, at $1.0\text{\AA}$ RMSD cutoff, Flib's precision is 74.6%, which is higher than NNMake's 59% and HHFrag's 64.7%. Flib's precision for *Majority β -strand* fragments is also higher.

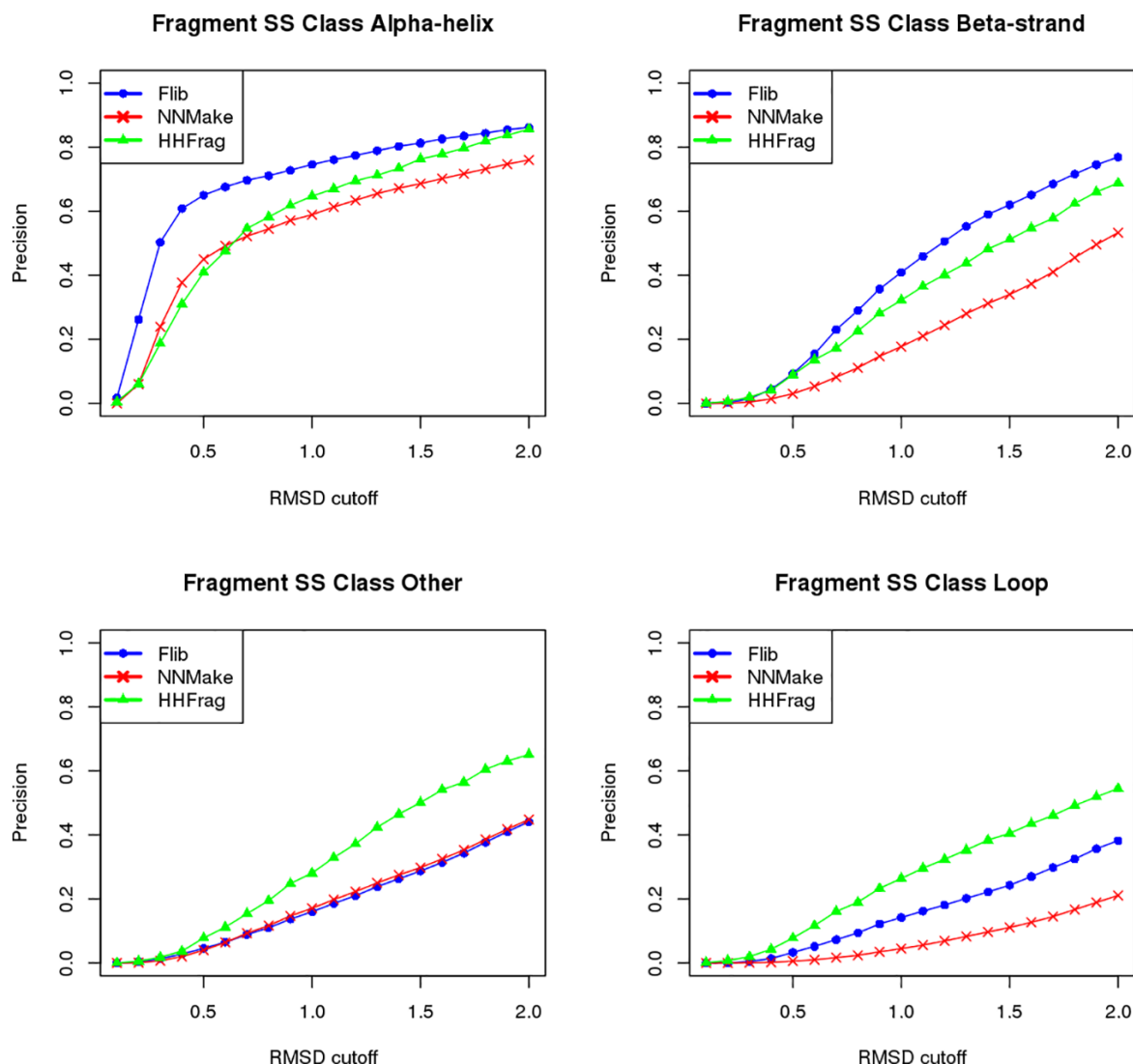


Fig. 2.14 Precision of fragment libraries generated using NNMake (red), HHFrag (green), and Flib (blue) separated by SS Class. The precision of the fragment libraries were averaged on a set of 41 structurally diverse proteins. We varied the cutoff that defines a good fragment (x axis) and evaluated the precision (proportion of good fragments in the libraries) for each method within four different SS classes: *Majority α -helical* (top left), *Majority β -strand* (top right), *Majority Loop* (bottom right) and *Other* (bottom left).

At 1.0Å RMSD, Flib presents 41% precision against NNMake's 17.7% and HHFrag's 32.2%. The precision of Flib for the other two SS classes are comparable to NNMake's precision (less than 5% difference in precision), but lower than HHFrag's. The coverage of Flib libraries slightly exceeds the coverage of NNMake for all SS classes.

CASP Validation Set

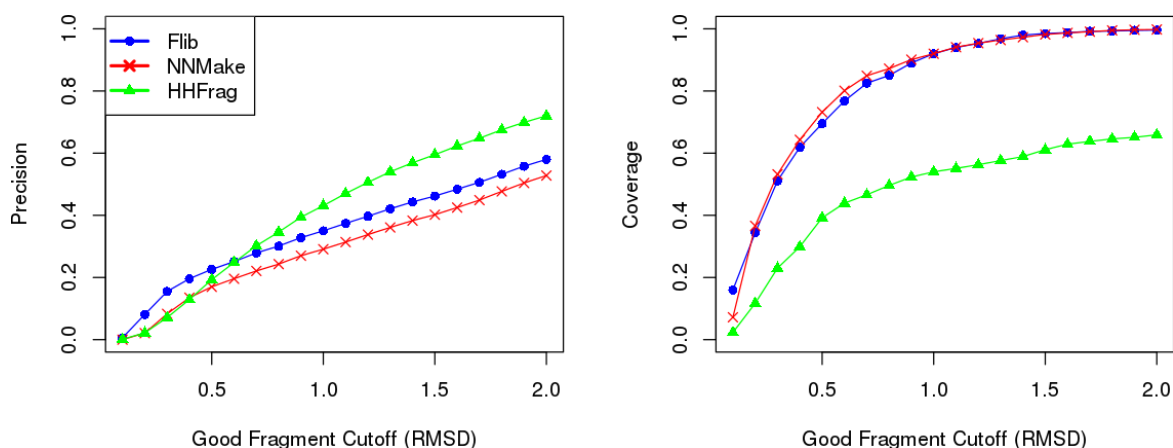


Fig. 2.15 Precision (left) and coverage (right) of fragment libraries generated using NNMake (red), HHFrag (green) and Flib (blue). The precision and coverage of the fragment libraries are averaged on a set of 275 protein domains that were used in CASP9 and CASP10. We varied the RMSD cutoff that defines a good fragment (x axis) and evaluated the precision (proportion of good Fragments in the libraries) and coverage (proportion of protein residues represented by a good fragment) for each method.

Similar results were observed for the CASP Validation Set. Flib also achieves the best balance between coverage and precision. Considering a good fragment cutoff of 1.0 Å, Flib presents the highest average overall precision, $\approx 39.4\%$, compared to HHFrag, $\approx 36.9\%$, and NNMake, $\approx 31.1\%$. On this set, Flib outputs, on average, 30.9 fragments per position, with an average length of ≈ 7.7 residues. HHFrag outputs on average 4.4 fragments per position, with an average length of 9.7 residues. NNMake always outputs 200 fragments per position with a constant length of nine residues.

Within each SS class, results on the CASP Validation Set are also analogous to results on the PDB-Representative Validation set. The three methods perform well at predicting fragments for *Majority α -helical* fragments, however, at 1.0 Å RMSD cutoff, Flib's precision is 60.6%, which is higher than NNMake's 54% and HHFrag's 58.1%. Flib's precision for *Majority β -strand* fragments is also higher. At 1.0 Å RMSD, Flib presents 32.5% precision against NNMake's 16.2% and HHFrag's 17.7%. The precision of Flib for the other two

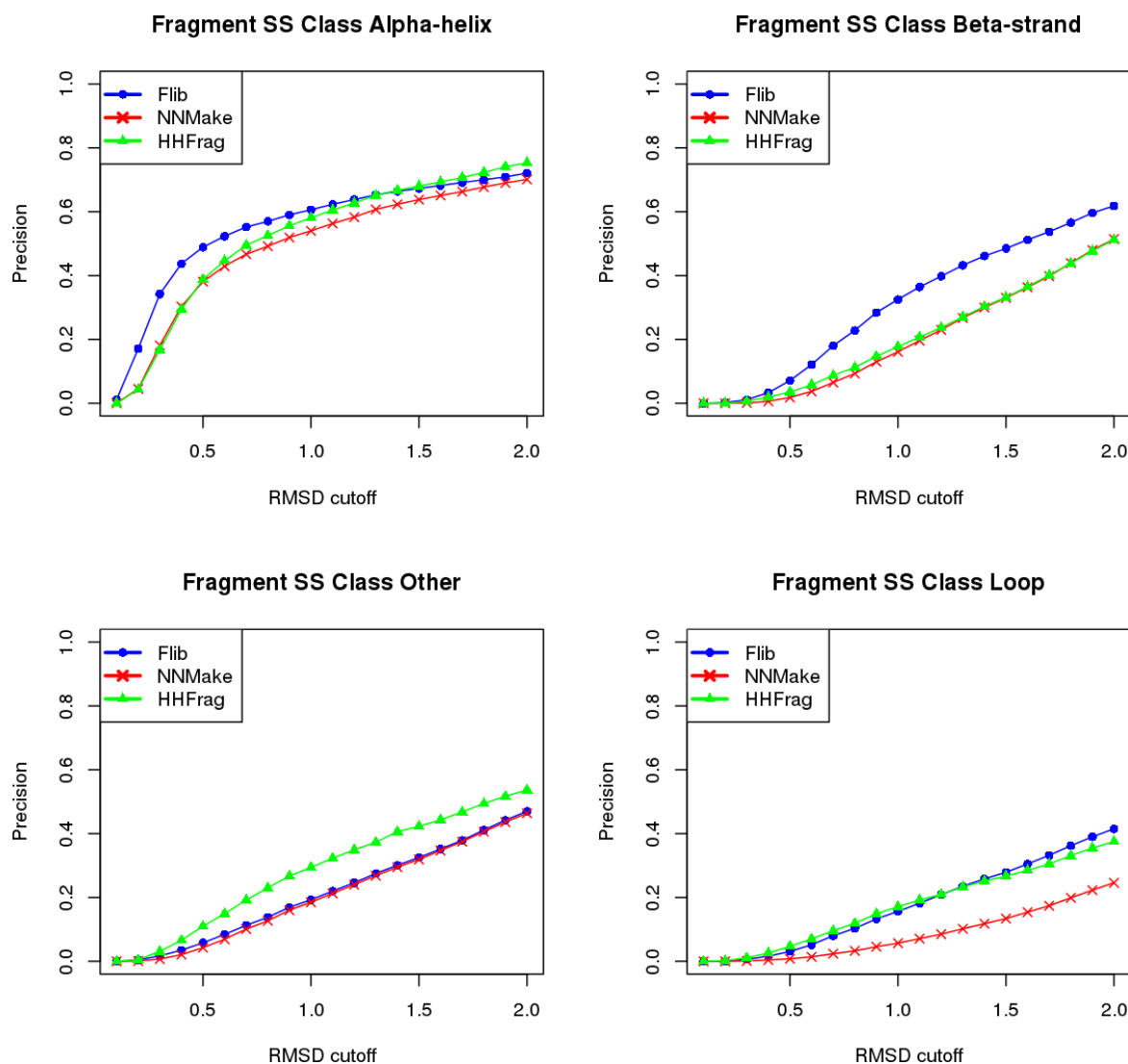


Fig. 2.16 Precision of fragment libraries generated using NNMake (red), HHFrag (green), and Flib (blue) separated by SS Class. The precision of the fragment libraries were averaged on a set of 275 protein domains that were used in CASP9 and CASP10. We varied the cutoff that defines a good fragment (x axis) and evaluated the precision (proportion of good fragments in the libraries) for each method within four different SS classes: *Majority α -helical* (top left), *Majority β -strand* (top right), *Majority Loop* (bottom right) and *Other* (bottom left).

SS classes are comparable to NNMake's precision (less than 5% difference in precision), but lower than HHFrag's. The coverage of Flib libraries slightly exceeds the coverage of NNMake for all SS classes.

Statistical Assessment of Cross-Comparison Results

To assess the statistical significance of our results, we have compared the distribution of RMSDs to the native structure of all fragments output by Flib and NNMake, for all targets in our PDB-Representative Validation Data set (Figure 2.17). We performed a Kolmogorov-Smirnov test (alternative hypothesis that the cumulative distribution function of the RMSDs of Flib fragments is greater than the cumulative distribution function of the RMSDs of NNMake fragments) and we obtained a p-value of 2.2×10^{-16} . This indicates that Flib generates fragments with statistically significant lower RMSDs compared to NNMake.

Comparing Fragment Libraries with Different Fragment Lengths

We have used RMSD to assess the quality of different fragment libraries. As mentioned in Section 1.5.3, RMSD is a length dependent measure and longer fragments will present higher RMSD as a consequence of this. Given that Flib's fragments are on average shorter than NNMake's, we investigated whether Flib's higher precision could be explained due to the length dependence of the RMSD. We built a new fragment library considering only the first eight residues of each of the fragments output by NNMake. We have chosen to use eight residues as this is approximately the average fragment length of Flib's fragment libraries. We noticed a slight improvement in the precision of NNMake's libraries, but Flib libraries still presented a higher precision and coverage (data not shown) when compared to the modified NNMake libraries.

2.3.9 Effect of Homologues on Fragment Library Quality

We carried out an analysis to assess the impact of extracting fragments from sequence homologues of the target protein on fragment library quality. It has been shown that when a suitable template (a homologue) can be found for a specific target, template-based modelling is the most accurate way to model the structure of that target (Moult et al., 2014). Hence,

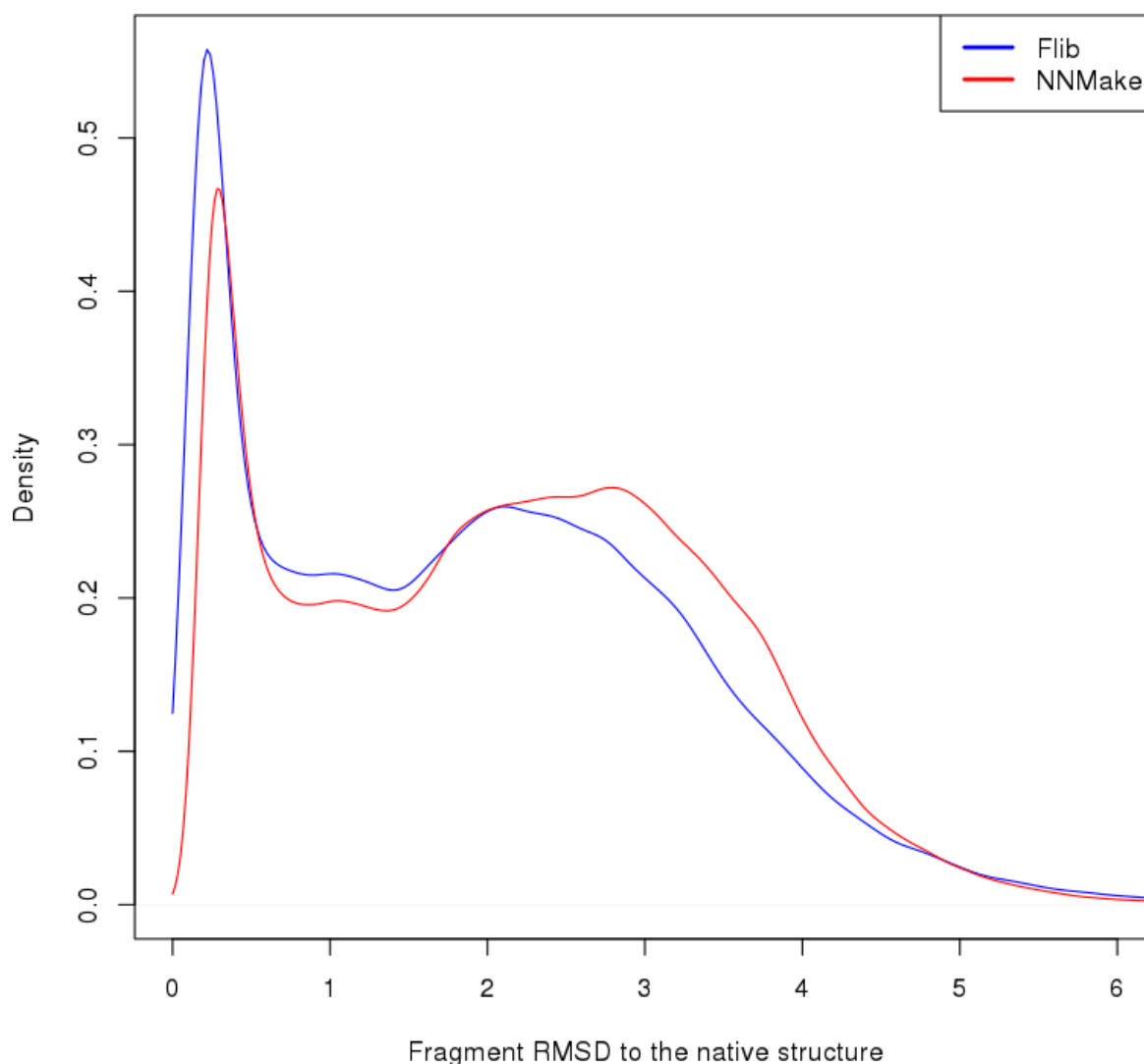


Fig. 2.17 Distribution of RMSDs to the native structure of all fragments as generated by NNMake (red) and Flib (blue). Fragments were generated for the 41 proteins in the PDB-Representative Validation Data set. Flib generates fragments with statistically significant lower RMSDs compared to NNMake ($p\text{-value} = 2.2 \times 10^{-16}$). Density plot was generated using a gaussian kernel density estimator in R.

de novo protein structure prediction tends to be used only in cases where no homologues can be found. For that reason, a fragment library that is representative of a real *de novo* protein structure prediction case should not contain fragments extracted from homologues. Nonetheless, not all current methods for fragment library generation exclude such fragments

from their outputs or from their published tests (Gront et al., 2011; Kalev and Habeck, 2011; Li et al., 2008; Shen et al., 2013).

We have analysed the impact of including/excluding fragments extracted from sequence homologues from fragment libraries generated by two different methods: NNMake (Gront et al., 2011) and HHFrag (Kalev and Habeck, 2011) (Figure 2.18). Homologues were extracted from the significant hits output by HHSearch (see Section 2.2.2 for more details). Results were generated using the PDB-Representative Validation Dataset of 41 structurally diverse proteins. HHFrag does not provide an option to exclude homologues from their template databases, and fragments resulting from homologues had to be removed from their final outputs in a post-processing step.

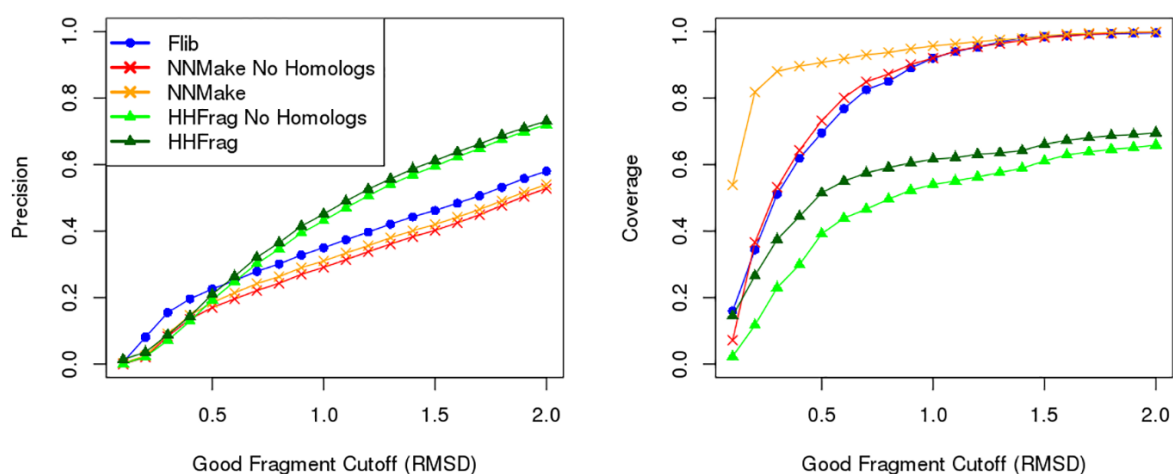


Fig. 2.18 Effect of Homologues on fragment library quality. Precision (left) and coverage (right) of fragment libraries generated using three different methods: NNMake (crosses), Flib (circles), and HHFrag (triangles). We varied the cutoff that defines a good fragment (x axis) and evaluated the precision (proportion of good fragments in the libraries) and coverage (proportion of protein residues represented by a good fragment) for each of the methods when: homologues are included (red and orange) and when homologues are excluded (light and dark green). Homologues are always excluded from Flib (blue).

We have compared the precision and coverage of fragment libraries generated before and after homologue removal. If we consider that the cutoff of 1.0Å defines a good fragment, homologue exclusion leads to a loss of precision from $\approx 35\%$ to $\approx 29.1\%$ for NNMake and from $\approx 45\%$ to $\approx 43.1\%$ for HHFrag libraries. Homologue exclusion leads to a loss

of coverage from $\approx 61.6\%$ to $\approx 54\%$ for HHFrag libraries and from $\approx 96\%$ to $\approx 92\%$ for NNMake libraries. Our results show that fragments extracted from homologues increase the precision and coverage of fragment libraries.

2.3.10 Model Generation/Protein Structure Prediction

Our results indicate that Flib libraries present higher precision and coverage when compared to NNMake's. However, in order to determine how well those results translate to *de novo* protein structure prediction, it is necessary to test the applicability of our libraries within a protein modelling framework. Therefore, we have generated models using Flib libraries firstly to assess if accurate models could be generated using those libraries and secondly to compare to the models generated using NNMake's libraries. We used our custom implementation of the fragment-based *de novo* structure prediction software, SAINT2, to combine the fragments and to sample the conformational space (see Chapter 3 for further details).

We generated 1,000 decoys for each of the proteins in our PDB-Representative Validation Data set using SAINT2 and Flib libraries. We compared our results to the results obtained by generating 1,000 decoys with NNMake libraries using SAINT2 (Figure 2.19). The Flib libraries generated accurate models (TM-Score > 0.5) for 12 of the 41 cases in the PDB-Representative Validation Data set. The NNMake libraries generated accurate models for 8 of the 41 cases. Of the 13 cases for which accurate models were generated by either method, Flib libraries performed better in 10. Flib failed to generate a correct model in only one case where NNMake libraries produced an accurate result, whereas NNMake libraries failed to generate a correct model in 5 cases where Flib libraries produced good models.

Addressing the Low Number of Decoys

The number of decoys that were generated during the analysis described in the previous section is comparable to the number of decoys that were generated in previous works (Kalev

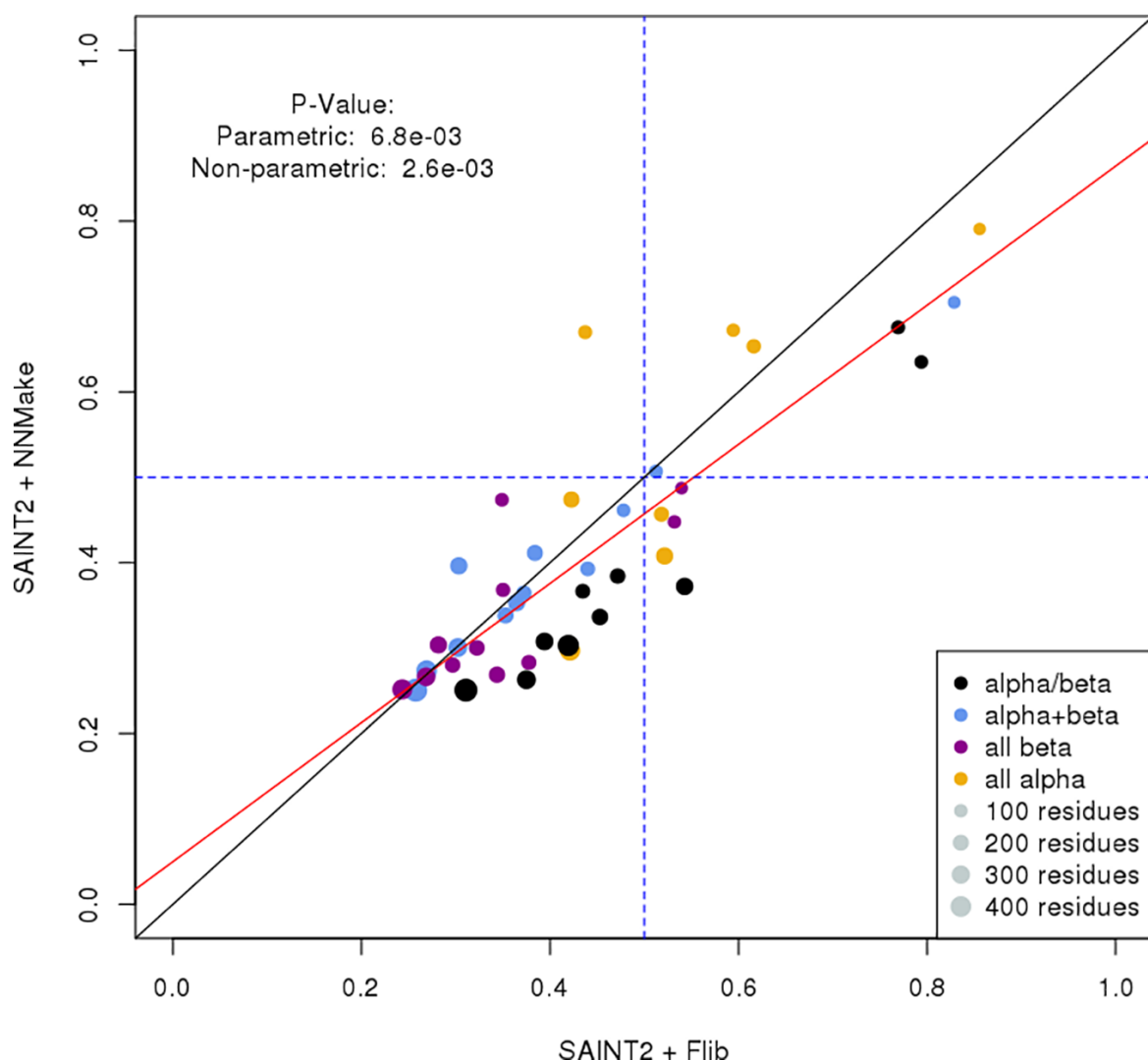


Fig. 2.19 TM-Score of the best decoy as generated by Flib+SAINT2 and by NNMake+SAINT2. For each approach, 1,000 decoys were generated and the best decoy (highest TM-Score when superimposed to native structure) was chosen. Results are shown for the 41 proteins in our PDB-Representative Validation Data set. We compared the TM-Score of best decoy generated by Flib + SAINT2 (x-axis) against NNMake + SAINT2 (y-axis). Each point represents a target. Point color represents the target's SCOP class and the point size is proportional to the protein length. The dotted lines indicate the cutoff for defining an accurate model (TM-Score > 0.5). Flib libraries generated accurate models for 12 of the 41 cases in our PDB-representative set. NNMake libraries generated an accurate model for 8 of the 41 cases. On the 13 cases for which accurate models were generated, Flib libraries performed better in 10 cases. Flib outperforms NNMake in 31 of the 41 cases.

and Habeck, 2011; Shen et al., 2013). However, this number is relatively low and it is hard to assess the statistical significance of our results. We tried to address this issue by comparing

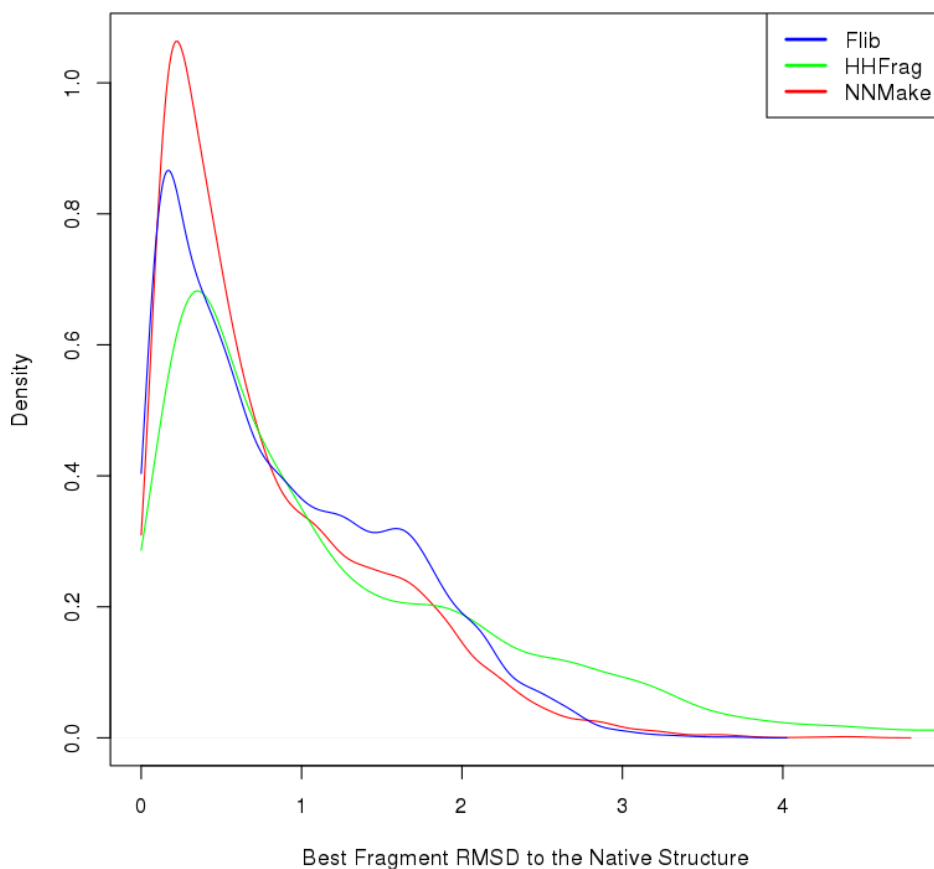


Fig. 2.20 Distribution of RMSDs to the native structure of the best fragment per target position as generated by each of NNMake (red), HHFrag (green), and Flib (blue). Fragment libraries were generated for the 41 proteins in our PDB-Representative Validation Data set. Best fragments for each target position were selected using the RMSD to the native structure.

the RMSD to the native structure of the best fragment for each target position obtained by each of Flib, NNMake and HHFrag (Figure 2.20). In principle, if the fragment assembly is exhaustive or has reached convergence, it is the best fragment within each window that ultimately determines the outcome of *de novo* structure prediction. Therefore, this comparison describes how well a fragment library can be used to model a target independent of the number of decoys generated. The RMSDs of the best fragments for each target position between Flib and NNMake are comparable, yet Flib libraries are nearly 10 times smaller than NNMake's. For that reason, Flib's best fragments are nearly 10 times more likely to be

chosen during the modelling routine. HHFrag's best fragments are significantly worse when compared to the other two methods.

2.4 Discussion

In this chapter, we have established that removal of homologues from any fragment library generation pipeline is essential to ensure that the precision and coverage obtained are representative of a realistic *de novo* structure prediction scenario, otherwise overly promising results will be shown.

We tested different template databases (subsets of the PDB) in order to understand how database size and selectiveness can affect the quality of generated fragment libraries. Our analysis revealed that larger template databases give marginally better results. This implies that errors introduced by low quality structures are compensated for by the diversity introduced by using more proteins. Our analyses also show that extracting fragments exclusively from templates that share the same SCOP-class as the target does not improve results.

The correlation between sequence and secondary structure scores and fragment RMSD to the native structure was also investigated. We observed that, once homologues are excluded from template databases, sampling at random from fragments that satisfy a score cutoff produces better results than extracting fragments exhaustively. We opted to employ a combination of both methods (random sampling and exhaustive sampling) in Flib. Exhaustive extraction is useful for finding high scoring fragments that are likely to be good, whereas random methods increase the diversity of the final ensemble.

We have observed that ranking fragments according to our Predicted Torsion Angle score improved results. Previous results suggest that predicted torsion angles perform better than predicted secondary structure in assisting protein structure prediction (Faraggi et al., 2009). Fragments extracted from protein threading hits were also added to our fragment libraries.

These fragments improved the accuracy of generated libraries and the fragment ensembles become more consistent because a large number of fragments are extracted from the same template structure. It has been reported that fragment consistency might be more important than target RMSD to generate good models (Abbass and Nebel, 2015). Fragment consistency is defined as how well a set of fragments representing different target positions can be pieced together.

Our analyses have also revealed a strong relationship between fragment RMSD from the native structure and the predominant predicted secondary structure of the fragments. We have separated fragments into four distinct classes (SS Classes) based on their predominant predicted secondary structure and have shown that more lenient cutoffs lead to higher precision in *Majority loop* and *Other* fragments, but stricter cutoffs lead to higher precision in the *Majority α -helical* and *Majority β -strand fragments*. These results also suggest that it is harder to predict good fragments for positions that are represented by *Majority loop* or *Other* fragments. During model generation, it may be beneficial to concentrate sampling efforts into these harder to predict positions.

Flib presents a better balance between coverage and precision when compared to HHFrag and NNMake. Compared to NNMake, Flib can generate fragments with varying lengths. This has been previously shown to improve protein structure prediction. Flib fragments are, on average, 1 residue shorter than NNMake fragments. However, we have shown that Flib libraries still presented higher precision and coverage compared to an adapted NNMake fragment library with shorter fragments. When compared to HHFrag, Flib presents a higher coverage. Flib also outputs fragments for every target position, which is necessary for structure prediction.

We have compared the improvement obtained by using Flib libraries against NNMake libraries in a protein structure prediction framework. Flib libraries generated accurate models in 12 out of the 41 test cases. Further, our libraries outperform NNMake in 10 of the 13

cases where an accurate model was generated. The number of decoys we have generated during our structure prediction analysis is relatively low. For that reason, we compared the RMSD to the native structure of the best fragment for each target position obtained by each of Flib, NNMake and HHFrag. In principle, if the fragment assembly is exhaustive or has reached convergence, it is the best fragment within each window that ultimately determines the outcome of *de novo* structure prediction. Therefore, this comparison describes how well a fragment library can be used to model a target independent of the number of decoys generated. The RMSDs of the best fragments for each target position between Flib and NNMake are comparable and as Flib libraries are nearly 10 times smaller they are better suited for structure prediction. Furthermore, the modelling step in our analysis is computationally intensive. For that reason, we chose to work with a reduced number of targets (41 proteins). We believe that our data set is large enough to assess the impact of using better libraries in a structure prediction context, despite probably not being large enough to be representative of the complete protein fold space.

Chapter 3

De novo Protein Structure Prediction Assisted by Co-evolution

3.1 Introduction

De novo protein structure prediction, also known as template-free modelling, is the method of choice for structure prediction when structural homologs to a target sequence cannot be effectively identified (Moult et al., 2014). Traditionally, free modelling methods perform protein structure prediction by means of a global optimization of a pre-defined function, in which the native structure is assumed to represent the global minimum (Anfinsen, 1972). If the energy function is imprecise, the heuristic search of the conformational space is likely to fail (Kinch et al., 2011).

One of the many challenges of *de novo* protein structure prediction is the implementation of an accurate energy function (Kinch et al., 2011). In order to address this problem, it is necessary to improve existing potentials. This has been observed during previous iterations of the Critical Assessment of protein Structure Prediction (CASP), in which the creation of new potentials or improvement of existing potentials led to better CASP results (Moult et al., 2014).

In particular, the last two iterations of CASP have shown that the use of correlated mutations can significantly improve *de novo* protein structure prediction. The principle of correlated mutations is that sequence positions in a multiple sequence alignment that mutate in a correlated fashion are likely to represent residues that share a spatial proximity (Figure 3.1). These co-evolving residues are thought to be protein contacts because of the assumption that a mutation in a given position occurs to compensate for another, thus preserving protein stability and function.

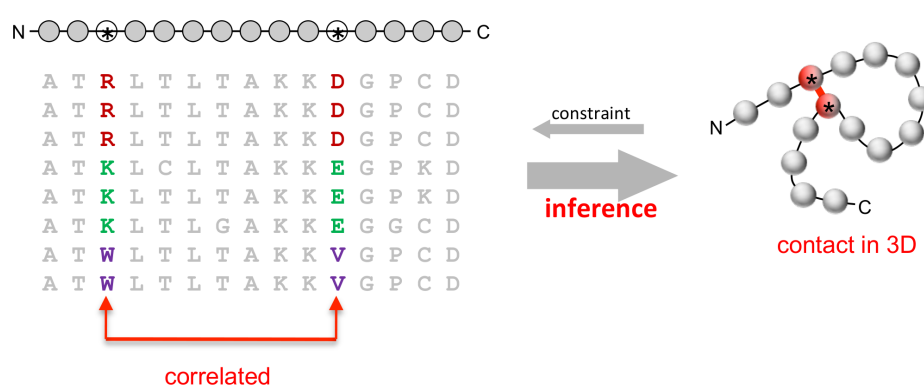


Fig. 3.1 Multiple sequence alignments can be used to detect pairs of protein residues that are mutating in a correlated fashion (coevolving). Coevolving pairs of residues are inferred to be in spatial proximity.¹

The notion of correlated mutations was first introduced by Gobel et al. (1994). This work was one of first studies that described how correlated mutations observed in a multiple sequence alignment could be used to infer spatial constraints in a protein structure. Gobel and colleagues showed that structural information (spatial constraints) could be inferred from sequence data even using a simplistic formulation of correlated mutations. More importantly, to our knowledge this was the first study that related how this information could be used to improve protein structure prediction.

Several works succeeded the original publication by Gobel U. et al (Fariselli et al., 2001a,b; Olmea and Valencia, 1997; Pazos et al., 1997; Shindyalov et al., 1994) . These

¹Image reproduced from Marks et al. (2011) under Creative Commons (CC BY 4.0 - <http://creativecommons.org/licenses/by/4.0>).

studies were based on similar assumptions and aimed to increase the precision of contacts inferred from correlated mutations. It became clearer that if these contacts could be inferred accurately enough, this would have a huge impact in the way models were generated.

In the early 2000s, a technique based on information theory and the concept of Mutual Information (MI) gained popularity (Cline et al., 2002; Liu and Bahar, 2012; Shackelford and Karplus, 2007). MI consists of ranking every pair of residues in a target sequence according to the mutual information between their corresponding pair of columns in a multiple sequence alignment of homologues to the target. The highest ranking pairs are considered to be in contact in the protein structure. Predictions computed using MI are not precise enough to be useful for *de novo* protein structure prediction (Horner et al., 2008). The failure of MI is thought to be caused by three sources of error (Marks et al., 2012): phylogenetic bias, entropic bias, and indirect coupling. Phylogenetic bias is observed because the sequences in the MSA present a phylogenetic relationship and columns are expected to be intrinsically correlated. The entropic bias is present when low sequence information is available, either because the number of sequences in the MSA is too small or there isn't enough sequence variation amongst the available sequences. Indirect coupling, also known as transitivity, relates to a correlation falsely observed between residues A and C because residues A and B and residues B and C present correlated mutations.

The precision of contacts inferred from correlated mutations has been improved with the exponential growth in the number of protein sequences available in the public domain and the emergence of new statistical techniques that address some of MI's limitations (Marks et al., 2012). Mean Field Direct Coupling Analysis (mfDCA) was one of the first methods to show a significant improvement in the precision of predicted contacts as compared to MI (Morcos et al., 2011). Unlike Mutual Information, mfDCA uses the information of all columns in the alignment when ascertaining the correlation between two individual columns. This was suggested to be important to address transitivity (Marks et al., 2011). Marks and

colleagues reported how mfDCA could be used to improve *de novo* structure prediction both for soluble (Marks et al., 2011, 2012) and transmembrane proteins (Hopf et al., 2012). Their method, EV-Fold Couplings, was initially tested on 15 soluble proteins and shown to obtain a predicted contact precision greater than 50% for 12 cases. These contact predictions were subsequently used as distance constraints within the *de novo* protein structure prediction software EV-Fold and shown to produce accurate structural models (Marks et al., 2011). An adapted version of EV-Fold called EV-Fold membrane was also tested for 25 transmembrane proteins and shown to produce accurate models for a majority of cases (Hopf et al., 2012).

Another approach that has been effectively used to predict protein contacts from multiple sequence alignments uses an estimate of the inverse covariance matrix to assign a score to residue pairs. The inverse covariance matrix approach has been implemented in PSICOV (Jones et al., 2012) and used to predict protein contacts for 150 targets. The top- $L/5$ long-range contact predictions (sequence separation >23) for a protein of length L were shown to have a precision greater than 0.5. These contact predictions were then used in the fragment-assembly software FRAGFOLD (Jones and McGuffin, 2003), generating accurate models for 100 out of the 150 targets.

A third approach has been developed based on learning the direct couplings as parameters of a Probabilistic Graphical Model (Markov random field) by maximizing its pseudo-likelihood. This approach has several implementations, such as plmDCA (Ekeberg et al., 2013), GREMLIN (Kamisetty et al., 2013) and CCMPred (Seemayer et al., 2014). GREMLIN has been tested for 329 protein targets selected from the Continuous, Automated Model Evaluation (CAMEO) (Haas et al., 2013) and shown to outperform PSICOV, plmDCA, and mfDCA. CCMPred was shown to obtain higher precision when compared to GREMLIN for the set of 150 targets previously described in Jones et al. (2012).

The three contact prediction approaches described above are partially orthogonal and output a significant number of non-overlapping predictions (Jones et al., 2014). Meta-

predictors have been developed to combine these non-overlapping predictions in an attempt to produce a consensus. PconsC (Skwark et al., 2013) and PconsC2 (Skwark et al., 2014) were based on using sixteen sets of predictions output by PSICOV and plmDCA, each obtained using a different multiple sequence alignment. A machine learning approach was used to generate the consensus set of predictions. PconsC2 was shown to obtain higher precision when compared to PSICOV, plmDCA and PconsC in three different sets: the 150 proteins used in the test set of PSICOV (Jones et al., 2012), a set comprised of 383 proteins selected from the PDB, and a set comprised of all the proteins used in CASP10 (Moult et al., 2014). Another consensus method, metaPSICOV, uses predictions output by Freecontact (a mfDCA implementation) (Kaján et al., 2014), PSICOV, and CCMPred. Resulting contact predictions are used as features in a two-layer neural network. MetaPSICOV was tested on the original PSICOV benchmark set of 150 protein, and it achieves a mean precision of 0.54 for top- L predicted long range contacts, reported to be 60% higher than PSICOV's and 40% higher than CCMPred's precision.

Apart from metaPSICOV and PconsC, which include predicted secondary structure as one of the features in their neural network, most methods do not consider secondary structure information directly when generating contact predictions. The software BBcontacts (Andreani and Söding, 2015) has been developed to infer beta-strand pairing based on predicted contact maps output by CCMPred. BBcontacts has been tested on a set of 916 proteins achieving 50% precision for β - β residue pairs at 50% recall using predicted secondary structure and 64% precision at 64% recall using true secondary structure. Another tool, EPC-map (Schneider and Brock, 2014), combines contact predictions output by GREMLIN with physicochemical information, reporting an average improvement in precision of 4.4% compared to GREMLIN for a set of 104 proteins of CASP10.

The precision of predicted contacts has been shown to be dependent on the number of sequences used in the multiple sequence alignment (Jones et al., 2012; Kaján et al., 2014;

Marks et al., 2011, 2012; Morcos et al., 2011). The usefulness of predicted protein contacts to *de novo* protein structure prediction is, therefore, restricted to protein families for which a sufficient number of sequences is available. It has been suggested that the number of sequences required to produce reliable contact predictions should be of the order of $5 \times L$, where L is the length of the protein. A recent study (Kamisetty et al., 2013) estimated that the number of protein families in Pfam for which no homologue structure is known (5146) and for which there are more than $5 \times L$ sequences in the multiple sequence alignment is only 291. According to the authors, only about 15% of *de novo* modelling cases can benefit from predicted contact information.

Several studies have suggested that the scores obtained from correlated mutation analysis represent an evolutionary signal, i.e. pairs of residues with higher scores should be more important for protein stability and folding than pairs of residues with lower scores (Kaján et al., 2014; Kamisetty et al., 2013; Marks et al., 2011; Morcos et al., 2011; Seemayer et al., 2014). It is unclear whether the most accurate set of predictions is also the one that best captures this evolutionary signal; whether the most accurate set of predictions will lead to the best modelling results. Different contact prediction software are tested with different structure prediction programs and in different test sets. It is, therefore, hard to draw an unbiased comparison between different contact predictors in order to identify the one that is most suitable to *de novo* protein structure prediction. Many of the works described above focus on improving the precision of contact predictions. However, it is also necessary to ascertain how we can use such contact predictions most efficiently in order to improve contact assisted *de novo* protein structure prediction.

We have compared nine contact predictors using our PDB-Representative Validation set of 41 proteins. Out of the methods tested, MetaPSICOV Stage 2 produced, on average, the most precise set of predictions. We have also assessed the precision of predicted contacts between different secondary structure types and shown that predictions of contacts between

α -helical residues tend to be less precise compared to predicted contacts formed between other secondary structure types. We observe that no one method is the most precise for all types. Our results also show that all methods present lower predicted contact precision for proteins belonging to the All α class when compared to other SCOP classes.

CASP results indicate that one correct prediction every twelve residues is sufficient for accurate *de novo* topology modelling (Kim et al., 2014). We compared the spread of correct predicted contacts output by each method and showed that metaPSICOV Stage 1 and CCMPred correctly predicted contacts are more evenly spread out than predictions output by other methods. It has also been suggested that long range contacts (established between residues with a sequence separation greater than 23) are more important than short range contacts for efficient model generation (Jones et al., 2012, 2014). We show that, out of the methods tested, metaPSICOV Stage 1 presents the highest average precision for predicted long range contacts.

We have performed an unbiased comparison of the usefulness of each method to *de novo* protein structure prediction by testing different contact predictors on the same data set and using the same structure prediction software, our cotranslational program SAINT2 (see Chapters 4 and 5). We found that metaPSICOV stage 1 produced the best modelling results in a consistent fashion, producing a model of similar quality to best possible model for 39 of 41 cases. We also observed a strong correlation between the number of satisfied correct predicted contacts and model quality. This relationship is lessened, but still observed if we consider the proportion of predictions, correct or otherwise, that are satisfied in generated models. These results suggest that improving the precision of predicted contacts further may not be as important for structure prediction as improving how these predictions are used for model generation.

3.2 Methods

3.2.1 Contact Definitions

Two protein residues are defined to be in contact if their C- β s (C- α s for Glycine) are less than 8Å apart. Trivial contacts occur due to residues being less than five residues apart and are not considered in the scope of our analyses. A short-range contact between residues i and j is defined when $5 \leq |i - j| \leq 23$. A long-range contact is defined when $|i - j| > 23$. We also define six classes of contacts between different secondary structure types: loop-loop as formed between two loop residues, loop-helix, loop-strand, helix-helix, helix-strand, and strand-strand.

3.2.2 Mean Field Direct Coupling Analysis (mfDCA)

The Mean Field Direct Coupling Analysis (mfDCA) is a class of contact predictors based on a maximum entropy model. In this approach, the global probability to observe a particular sequence in the Multiple Sequence Alignment ($P(A_1, \dots, A_L)$) can be expressed in terms of residue couplings $e_{ij}(A_i, A_j)$, single residue terms $h_i(A_i)$, and a normalization constant Z according to the formula:

$$P(A_1, \dots, A_L) = \frac{1}{Z} \exp \left\{ \sum_{1 \leq i \leq j \leq L} e_{ij}(A_i, A_j) + \sum_{1 \leq i \leq L} h_i(A_i) \right\} \quad (3.1)$$

The number of parameters e_{ij} and h_i is large enough that solving Equation 3.1 becomes computationally intractable. Using a mean field approximation, the couplings e_{ij} can be calculated from the inverse of the empirical covariance matrix as follows:

$$e_{ij}(A_i, A_j) = -(C^{-1})_{ij}(A_i, A_j) \quad (3.2)$$

And the empirical covariance matrix can be computed from the individual and pair frequencies of amino acids:

$$C_{ij}(A_i, A_j) = f_{ij}(A_i, A_j) - f_i(A_i)f_j(A_j) \quad (3.3)$$

Where $f_i(A_i)$ and $f_{ij}(A_i, A_j)$ are the individual and pair frequencies of amino acids at positions i and i, j , respectively.

Once the e_{ij} s have been estimated, the probability of observing a given pair of residues A_i, A_j at positions i and j in sequence A of the MSA can be obtained using the formula:

$$P_{ij}^{Dir}(A_i, A_j) = \frac{1}{Z} \exp \left\{ e_{ij}(A_i, A_j) + \tilde{h}_i(A_i) + \tilde{h}_j(A_j) \right\} \quad (3.4)$$

Where Z is a normalization constant and $\tilde{h}_i$ and $\tilde{h}_j$ can be calculated imposing a single residue marginal frequency count compatibility condition. Finally, the direct information (DI) between two columns i and j in the MSA is defined as:

$$DI_{ij} = \sum_{A_i, A_j=1}^q P_{ij}^{Dir}(A_i, A_j) \ln \left(\frac{P_{ij}^{Dir}(A_i, A_j)}{f_i(A_i)f_j(A_j)} \right) \quad (3.5)$$

The DI of every column pair is used to rank the pairs of residues most likely to be in contact. In a final filtering step, columns pairs with high DI and that are physically not feasible (i.e. beginning and end of the same α -helix) or that are less than 5 residues apart are not considered for output. MfDCA-based approaches output the top L ranked predictions, where L is the length of the target sequence.

For the analyses described in this chapter, we used Freecontact version 1.0.21, with standard parameters (Kaján et al., 2014).

3.2.3 Protein Sparse Inverse COVariance Analysis Program (PSICOV)

PSICOV is also based on calculating an inverse covariance matrix to predict protein contacts. The covariance matrix used by PSICOV is defined in terms of the observed marginal frequencies of amino acids ($f(A_i)$ and $f(B_j)$) and amino acid pairs ($f(A_iB_j)$) in an analogous fashion to that described in Equation 3.3.

The covariance matrices as defined by PSICOV are singular and sparse because even in MSAs with a large number of sequences, not every amino acid will be observed at every column. Therefore, the routine makes use of an algorithm called Graphical Lasso (Friedman et al., 2008) to perform the sparse inverse covariance estimation. The Graphical Lasso has performance issues in some cases and the matrix needs to be conditioned via shrinking before its inverse can be estimated. For more details, refer to (Jones et al., 2012).

A contact prediction score S_{ij} for two columns i and j in the MSA is computed using the absolute value of a submatrix of the inverse covariance matrix Θ . This submatrix corresponds to the 20×20 amino acid types ab observed in the two alignment columns, where contributions from gaps are ignored. The score is computed as follows:

$$S_{ij}^{contact} = \sum_{ab} |\Theta_{ij}^{ab}| \quad (3.6)$$

The score S_{ij} is then corrected for background effects using the Average Product Correction (APC) (Dunn et al., 2008). The corrected score is then converted into an estimated positive predictive value by fitting a logistic function to the observed distribution of scores.

For the analyses described in this chapter, we used PSICOV version 2.1 with a target precision matrix sparsity of 0.03 and default parameters (Jones et al., 2012).

3.2.4 Pseudo-likelihood Maximization Direct Coupling Analysis (plmDCA)

Both mfDCA and PSICOV are significantly different in terms of their derivation and estimation procedures. However, these methods rely on estimating the inverse of the covariance matrix, which follows from their maximum entropy model. The third category of contact predictors avoids the approximation of the inverse covariance matrix altogether by maximizing a pseudo-likelihood.

In this approach, a generalised Potts model is used to maximize the probability of observing all the amino acid sequences in the MSA with respect to the parameters $e_{ij}(A_i, A_j)$ and $h_i(A_i)$. In order to infer those parameters, the method suggests that the ordinary statistical approach would be to maximize the likelihood, which is equivalent to minimizing the (rescaled) negative log-likelihood function l :

$$l(h, e) = \log Z - \sum_{i=1}^L \sum_{A_i=1}^{q=21} f_i(A_i) h_i(A_i) - \sum_{1 \leq i < j \leq L} \left(\sum_{A_i, A_j=1}^{q=21} f_{ij}(A_i, A_j) e_{ij}(A_i, A_j) \right) \quad (3.7)$$

Where $f_i(A_i)$ and $f_{ij}(A_i, A_j)$ are the individual and pair frequencies of amino acids at positions i and i, j , respectively.

The function l is differentiable and, therefore, can be minimized by encountering the points for which the partial derivatives equal zero. However, minimization is computationally intractable and an approximation needs to be performed. A pseudo-likelihood can be used as an approximation if one substitutes the probability in Equation 3.1 with the conditional proba-

bility of observing A_i , given all the other observations ($A_{\setminus i} = A_1, A_2, \dots, A_{i-1}, A_{i+1}, \dots, A_{L-1}, A_L$).

The conditional probability is defined as:

$$P_{\{h_i, e_i\}}(A_i | A_{\setminus i}) = \frac{\exp(h_i(A_i)) + \sum_{j=1, j \neq i}^L e_{ij}(A_i, A_j)}{\sum_{A_k=1}^{q=21} \exp(h_i(A_k)) + \sum_{j=1, j \neq i}^L e_{ij}(A_k, A_j)} \quad (3.8)$$

And the resulting negative log-(pseudo)likelihood function can be derived:

$$l_{pseudo}(h, e) = -\frac{1}{M} \sum_{i=1}^L \sum_{j=1}^M \log \left[P_{\{h_i, e_i\}}(A_i^j | A_{\setminus i}^j) \right] \quad (3.9)$$

Where M is the number of sequences in the MSA. The minimization of the negative log-(pseudo)likelihood function is now computationally tractable and the couplings can be learned as parameters of this minimization. Given the large number of parameters, a final regularization step ensures that no overfitting occurs during machine learning. To perform this regularization, a penalty term based on the l_2 norm is added to the objective function.

Once the couplings e_{ij} have been computed, a score is obtained from the couplings using the Frobenius norm of the zero-sum gauge e'_{ij} :

$$\mathcal{S}_{ij}^{FN} = ||e'_{ij}||_2 = \sqrt{\sum_{A_i, A_j=1}^{q=21} e'_{ij}(A_i, A_j)^2} \quad (3.10)$$

Where:

$$e'_{ij} = e_{ij}(A_i, A_j) - e_{ij}(\cdot, A_j) - e_{ij}(A_i, \cdot) + e_{ij}(\cdot, \cdot) \quad (3.11)$$

In this formula, a center dot denotes the average over the concerned position.

The final scores computed by plmDCA are obtained by applying the average-product correction to $\mathcal{S}_{ij}^{FN}$, in an analogous fashion to the correction performed by PSICOV to its coupling-derived scores.

For the analyses described in this chapter, we have used two implementations of plmDCA: CCMPred v0.1.0 with standard parameters (Seemayer et al., 2014) and GREMLIN v2.01 with standard parameters (Kamisetty et al., 2013). GREMLIN requires the proprietary software Matlab to generate predictions.

3.2.5 Consensus Methods

PConsC and PConsC2

PConsC uses different sets of contact predictions output by PSICOV and plmDCA (standard parameters are used for both methods). In total, eight different MSAs are used as input for PSICOV and plmDCA, resulting in sixteen sets of predicted contacts. These MSAs are generated using HHblits (Remmert et al., 2012) against its bundled nr20 database or Jackhmmer (Johnson et al., 2010) against UniRef100, for four different E-value cut-offs (10^{-40} , 10^{-10} , 10^{-4} , 1).

PConsC uses a random forest method to perform the classification of the sets of predictions. The input to PConsC's classifier is the 16 sets of raw predicted scores obtained by PSICOV and plmDCA. To avoid overfitting, a training set of 48 proteins was used to perform a 5-fold cross validation, which led to the determination of the number of trees and a leaf size. The method trained a 100-tree random forest with an early stopping condition, which required a minimum of 500 samples in newly created leaves.

PConsC2 enriches the predictions obtained by PConsC by means of a deep-learning method, which is based on the notion that protein contacts are likely to be in proximity to other protein contacts in the contact map. Likewise, non-contacts are likely to be close to

other non-contacts. The deep-learning method used by PConsC2 is a feed-forward stack of random forest learners, which aims to recognize visual patterns in the contact maps.

We have used PConsC2 with standard parameters (Skwark et al., 2014) for the analyses described in this chapter. PConsC2 requires the proprietary software Matlab to generate predictions.

metaPSICOV

MetaPSICOV is a meta-predictor based on contact predictions output by Freecontact (v1.0.21), PSICOV (v2.1), and CCMPred (v0.1.0). Unlike PConsC/PConsC2, in metaPSICOV the same alignment is used for all three methods. MetaPSICOV is based on a two stage classifier using classic feed-forward neural networks, where the prediction scores of each of the three contact predictors are input as features. Additional features extrapolated from the multiple sequence alignment and from predicted secondary structure and predicted solvent accessibility are also used during classification.

MetaPSICOV Stage 1, the first stage classifier, uses 672 features to predict the likelihood of two residues being in contact. MetaPSICOV Stage 2 uses some of these 672 features and an additional 121 features which describe a two-dimensional window of the contact map predicted by the metaPSICOV Stage 1, centred around the position i, j . The use of these features in the second stage classifier shares the same principles as those used PConsC2 that neighbouring contacts are likely to be in proximity to other contacts in the contact map. In total, metaPSICOV Stage 2 uses 731 features.

MetaPSICOV is also capable of outputting predictions for pairs of residues forming a backbone hydrogen bond, which is referred to as metaPSICOV-HB. MetaPSICOV-HB is similar to metaPSICOV Stage 2, but it has some adjustments to infer a pair of residues where the first residue is predicted to be a hydrogen donor. In this case, a backbone hydrogen bond

contact is defined as a pair of residues i, j , where the amino nitrogen of i is less than $3.5\text{\AA}$ from the carbonyl oxygen of j .

In the analyses described in this chapter, we have used metaPSICOV v1.01 with standard parameters (Jones et al., 2014).

3.2.6 Secondary Structure-Specific Methods

Bbcontacts

Bbcontacts uses predicted contact maps as output by CCMPred to identify contacts between β -strands. Two Hidden Markov Models (HMMs), one for parallel and one for anti-parallel strands, are used to infer the pairing of the strands based on the predicted contact map and based on predicted secondary structure. We have used CCMPred v0.1.0 (Seemayer et al., 2014) to generate predicted contact maps and used bbcontacts with standard parameters to infer beta-strand pairings (Andreani and Söding, 2015).

3.2.7 Number of Predictions Considered

For the sake of establishing a fair comparison between all methods, we have considered only up to L predicted contacts in our analyses. The software bbcontacts usually outputs less than $L/2$ predicted contacts so its precision and spread were computed based on the number of predictions available.

3.2.8 Data Set

For the analyses presented in this chapter, we have used the PDB-Representative Validation Data set of 41 proteins described in Section 2.2.1.

3.2.9 Validation Metrics

The **precision** of predicted contacts (also referred to as PPV) is the percentage of true positives in the predictions output for a given target.

The **spread of correct predicted contacts** is defined as the largest protein segment for which no contacts were predicted. The spread is shown as a proportion of the protein length.

3.2.10 Model Generation

The contact predictions output by PSICOV, Freecontact, CCMPred, BBContact, metaPSICOV Stage 1, metaPSICOV Stage 2, and metaPSICOV HB were used as input for the cotranslational mode of SAINT2 (refer to Chapter 4). We have opted not to test the contact predictions output GREMLIN and PConsC2 as they rely on proprietary software. We generated 10,000 decoys for each set of predicted contacts and for all targets in our PDB-Representative Validation set. Standard parameters and scoring weights were used during decoy generation (for more information, refer to Section 4.2.18).

3.2.11 SAINT2: Contact Potential

SAINT2, our cotranslational protein structure prediction software, contains a predicted contact potential in its scoring function (see Chapter 4 for full details). The input of SAINT2's Contact Potential is a list of predicted contact pairs and their coupling score. There is no limit as to how many predicted contacts can be used as input to SAINT2. During SAINT2's scoring (refer to Section 4.2.9 for more details), a score penalty is assigned to each predicted contact that is not satisfied in a given peptide conformation. The penalty is linearly proportional to the distance between the pair of residues that were predicted to be in contact. The score of a given pair of residues i, j predicted to be in contact is given by:

$$S_{ij}^{contact} = \begin{cases} 0, & \text{if } \|\mathbf{C}_\beta(i) - \mathbf{C}_\beta(j)\| = \sqrt{\sum_{\kappa=x,y,z} (C_\beta^\kappa(i) - C_\beta^\kappa(j))^2} < 8.0 \text{ \AA} \\ \|\mathbf{C}_\beta(i) - \mathbf{C}_\beta(j)\| - 8.0 \text{ \AA}, & \text{otherwise.} \end{cases}$$

Where $\mathbf{C}_\beta(i)$ and $\mathbf{C}_\beta(j)$ represent the coordinates of the C- β s (C- α s in the case of glycine) of residues i and j . The predicted contact score of a given conformation is the sum of the scores of every pair of residues predicted to be in contact. Several formulations of predicted contact potentials were tested and this was chosen as it produced the best results with SAINT2. The time complexity of the predicted contact score is $O(L_{cont})$, where L_{cont} is the number of predicted contacts used as input (usually, $L_{cont} = L$).

3.3 Results

3.3.1 Comparing state-of-the-art contact predictors

We tested nine state-of-the-art contact prediction methods for the 41 proteins in our PDB-Representative Validation data set. We compared the average precision between the methods, as obtained for the top $L/10$, $L/5$, $L/2$ and L predicted contacts output by each approach (Figure 3.2), where L is the length of the protein. Our results show that contact predictions output by metaPSICOV Stage 2 are, on average, the most precise. This result was shown to be independent of the number of predicted contacts considered (up to L predictions). We observed similar results when considering the overall precision of each method (Figure 3.3).

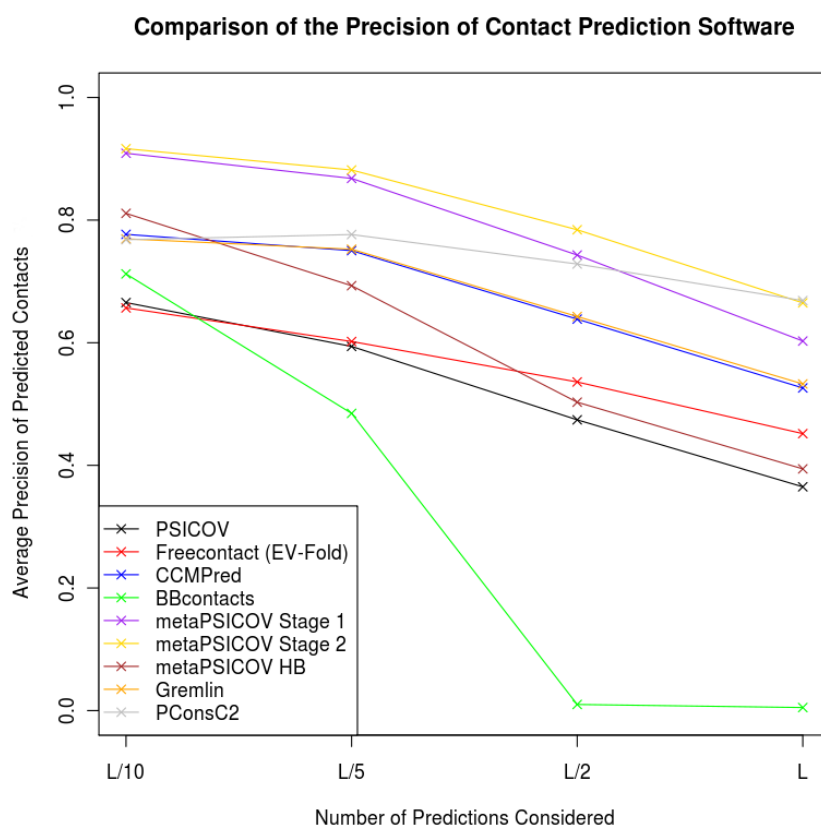


Fig. 3.2 The average precision of nine contact predictors for the 41 proteins in our PDB-Representative Validation data set. Precision is shown for the top $L/10$, $L/5$, $L/2$ and L predicted contacts.

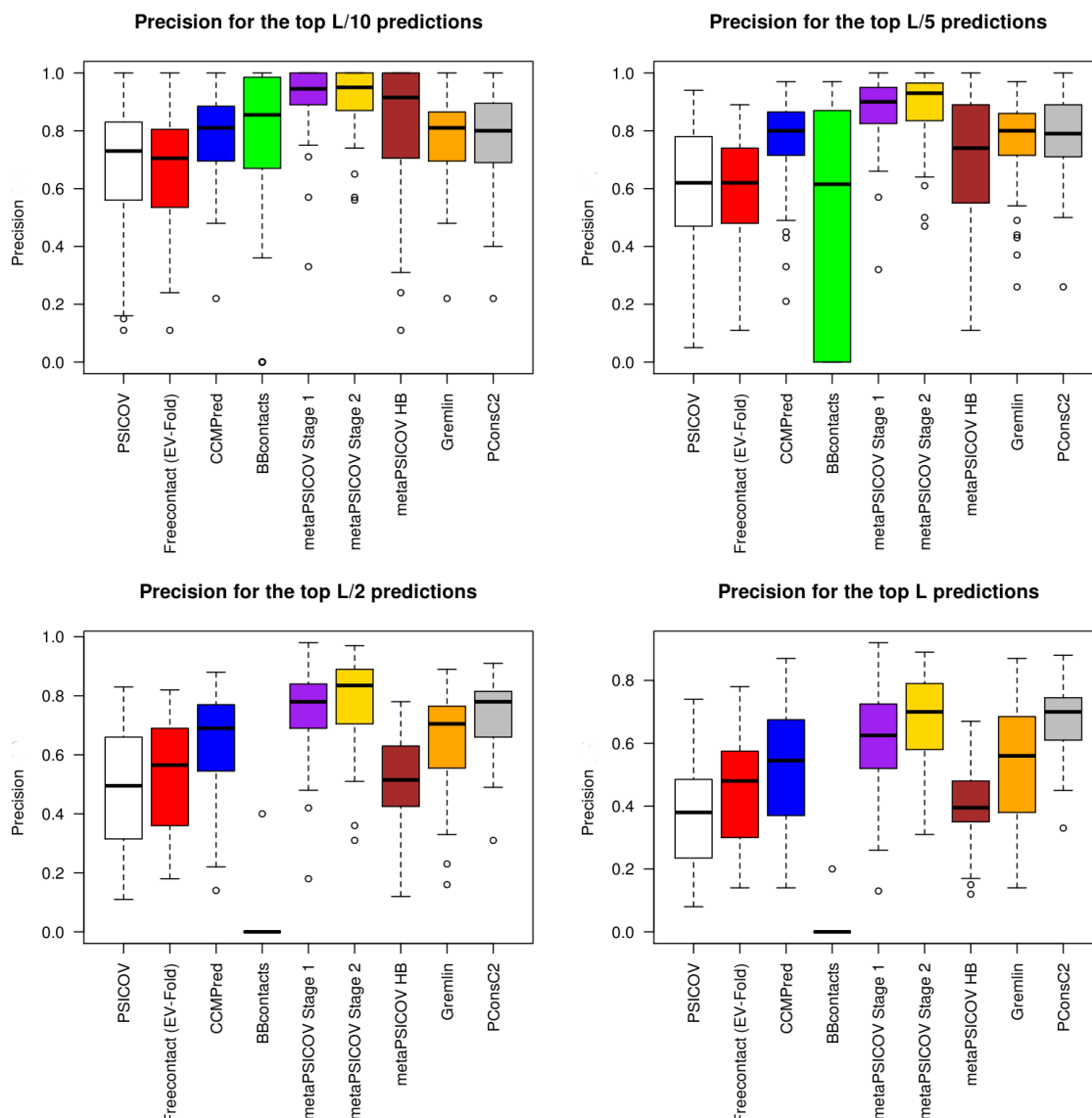


Fig. 3.3 Precision of nine contact predictors for the 41 proteins in our PDB-Representative Validation data set. Precision is shown for the top $L/10$, $L/5$, $L/2$ and L predicted contacts.

Precision of Short-Range and Long-Range Predicted Contacts

Long-range contacts have been suggested to be more useful than short-range contacts for *de novo* protein structure prediction (Jones et al., 2012). A long-range contact is formed between two residues that are more than 23 residues apart. To better understand the quality of the predictions output by different methods, we have quantified the precision and the

proportion of short and long-range predicted contacts generated in our PDB-Representative Validation data set (Figure 3.4).

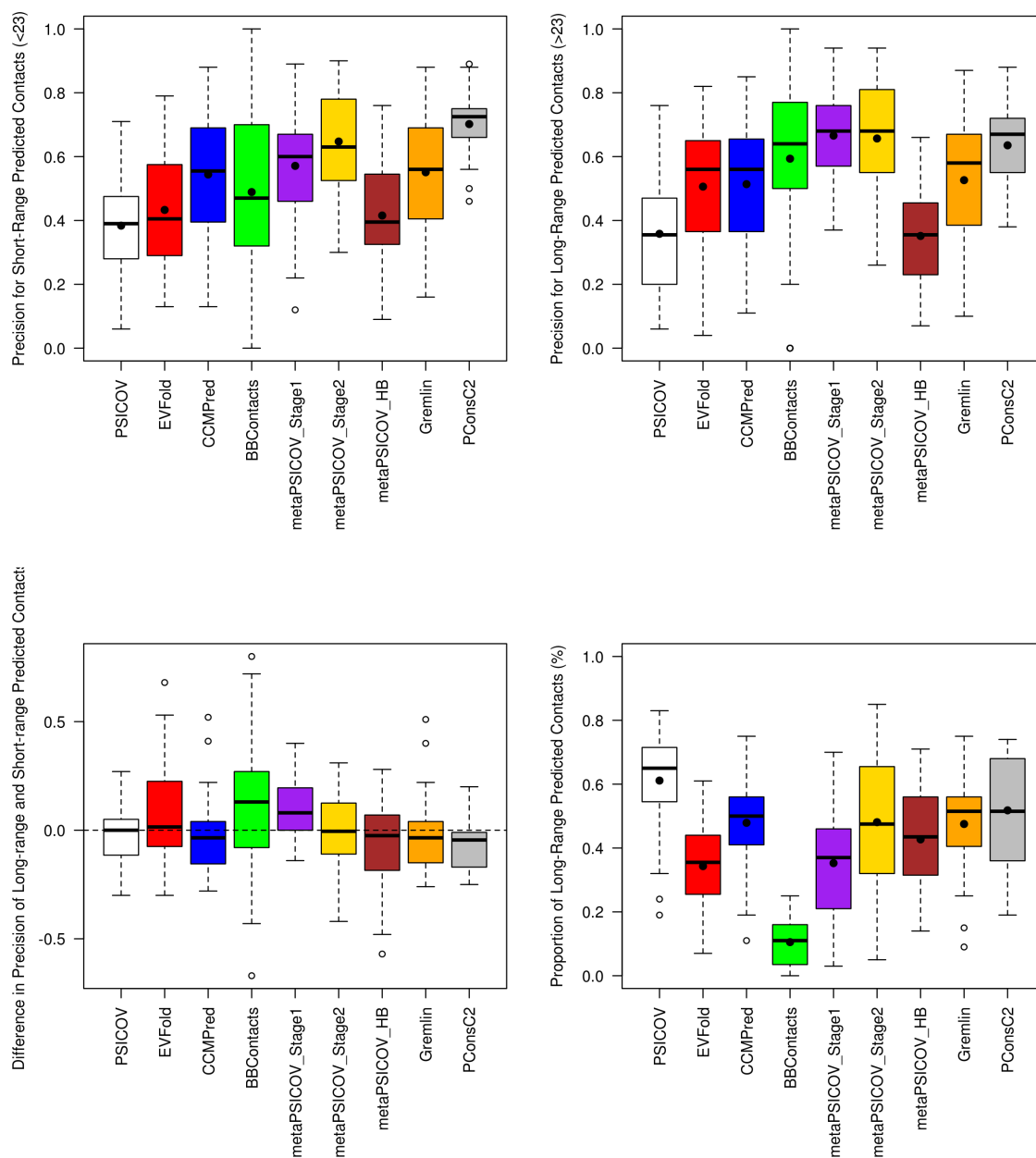


Fig. 3.4 Analysis of the precision and the proportion of short-range and long-range predicted contacts output by different contact predictors for 41 proteins in our PDB-Representative Validation data set. The average precisions and proportion are represented by a black dot.

PConsC2 and metaPSICOV Stage 1 presented the highest average precision for predicted short-range and long-range contacts, respectively. We assessed the proportion of short and long-range predicted contacts output by different methods for each case. This ratio varies significantly for different methods and for different proteins in our data set. EV-Fold, BBcontacts and metaPSICOV stage 1 output a higher number of predicted short-range contacts, whereas PSICOV outputs a higher number of predicted long-range contacts. The remaining methods are balanced in terms of the number of long and short-range predictions that they generate.

Precision of Predicted Contacts According to the Number of Sequences in the MSA

We assessed the effect of the number of sequences in the MSA on the precision of predicted contacts. We used the contact predictions generated by each method to build a confidence interval for the number of sequences necessary so that contact predictions are precise enough to be used in a *de novo* protein structure prediction (Figure 3.5). We used a precision threshold of 50%, as previously suggested in Jones et al. (2012). Our results show that metaPSICOV Stage 2 and PConsC2 are the methods that require the lowest number of sequences in the MSA in order to achieve 50% precision for the top L predicted contacts. These methods seem to require between ≈ 500 and $\approx 1,000$ sequences in the MSA.

Unlike previously suggested in Kamisetty et al. (2013), we observed that the correlation between the number of sequences in the MSA and the precision of contact predictions is independent of the protein length. We have used predicted contacts output by metaPSICOV stage 2 to demonstrate this result (Figure 3.6), however similar results were observed for all methods. The precision of predicted contacts is correlated to the log of the number of sequences in the MSA, but not to protein length. No additional information is gained by computing the correlation between the precision of predicted contacts and the log of the number sequences in the MSA divided by protein length.

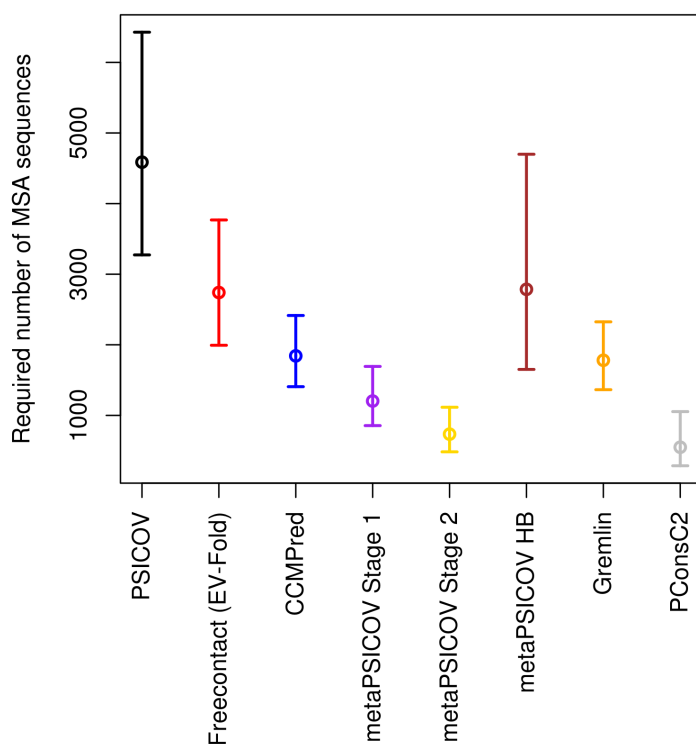


Fig. 3.5 Confidence Intervals (95%) for the estimates of the number of sequences required to produce contact predictions with a precision greater than 50% for different contact predictors.

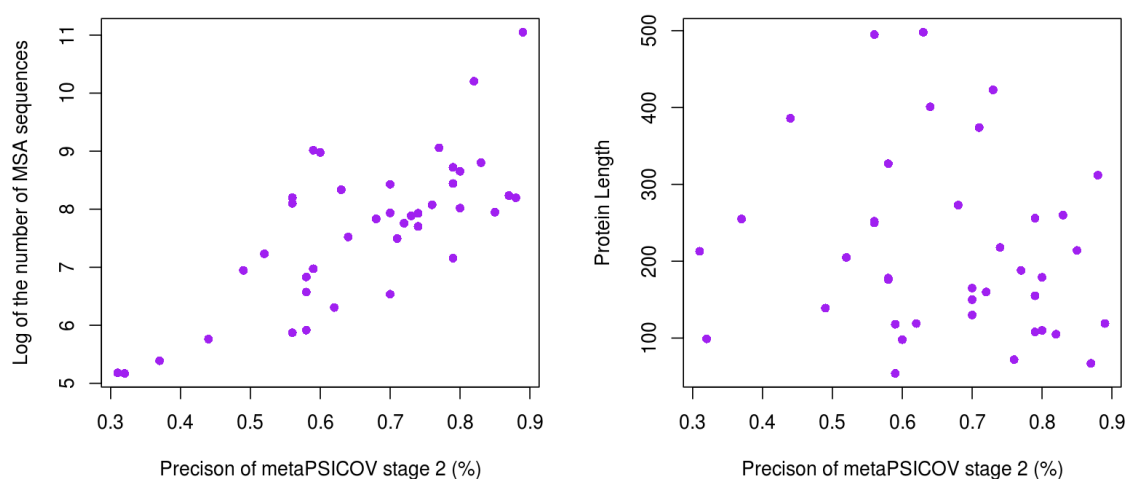


Fig. 3.6 Correlation between the precision of predicted contacts output by metaPSICOV stage 2 and the log of the number of MSA sequences (left) or protein length (right).

Precision of Predicted Contacts According to Secondary Structure

It has been reported that *bbcontacts* presents higher precision compared to other software when predicting contacts between β -strands (Andreani and Söding, 2015). Therefore, we assessed the precision of contacts predicted between different secondary structure types (Figure 3.7). Six categories of predicted contacts were compared. Here, a contact formed between two loop residues is described as loop-loop. Analogously, the remaining categories are loop-helix, loop-strand, helix-helix, helix-strand, and strand-strand. When considering contacts formed between different secondary structure categories, *metaPSICOV* Stage 2 presented the highest average precision for three categories (loop-loop, loop-strand, and helix-strand), *CCMPred* presented the highest average precision for helix-helix and strand-strand and *PConsC2* presented the highest average precision for loop-helix. The results in Figure 3.7 are shown for the top L predictions, but similar results were observed when we considered the top $L/10$ predicted contacts.

We have also assessed the proportion of predicted contacts for each secondary structure category (Figure 3.8). Our results show that both *bbcontacts* and *metaPSICOV-HB* are enriched for predicted contacts formed by at least one β -strand residue. These methods also output, on average, a lower number of predicted contacts between α -helices when compared to other predictors. Most predictors generate a far lower number of loop-strand predicted contacts than other types. *CCMPred* and *GREMLIN* achieved the proportions that were the most similar to the true contact distribution for all secondary structure categories.

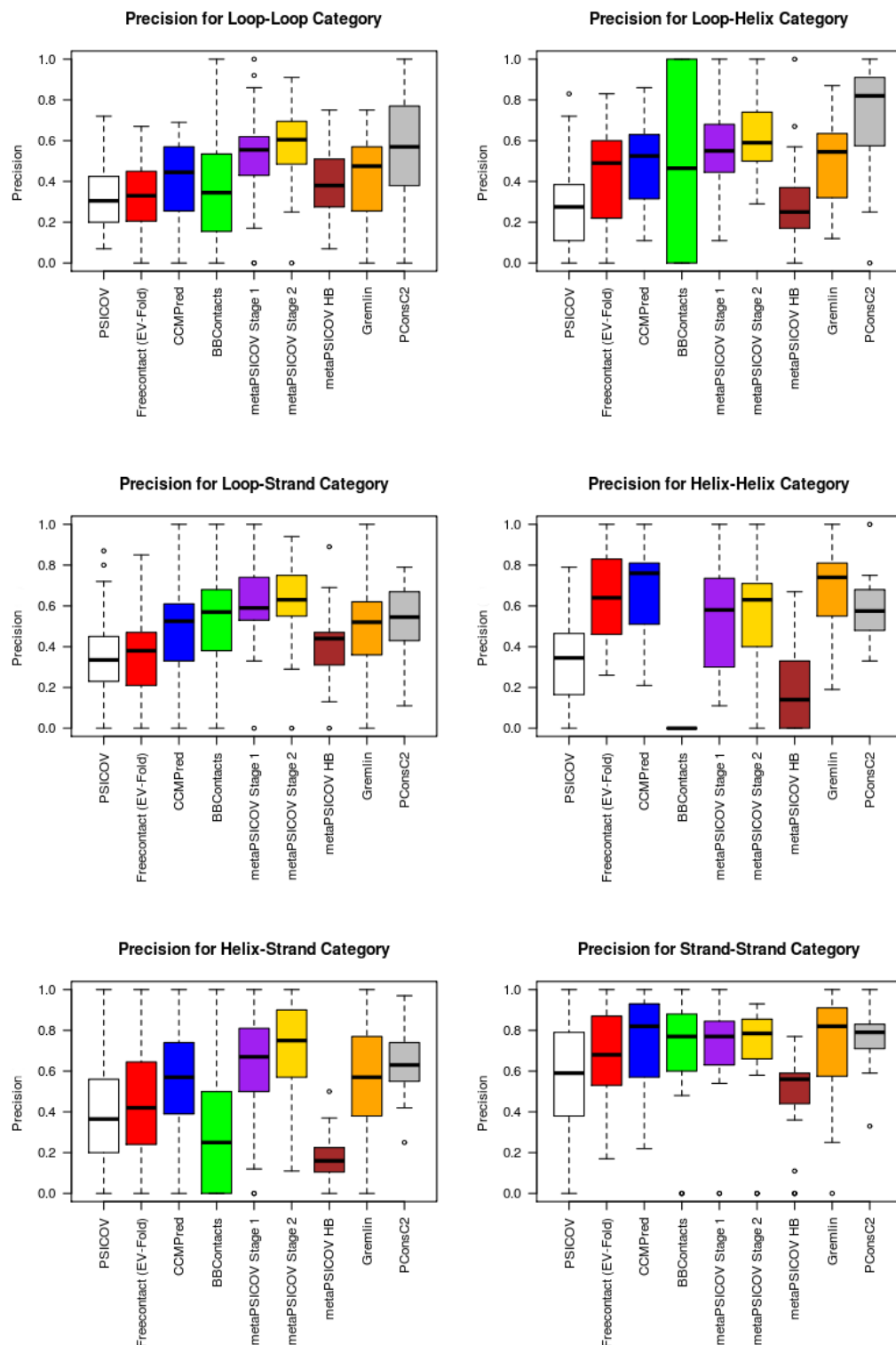


Fig. 3.7 Precision by secondary structure type of the top L predicted contacts as output by nine contact predictor for 41 proteins in our PDB-Representative Validation data set.

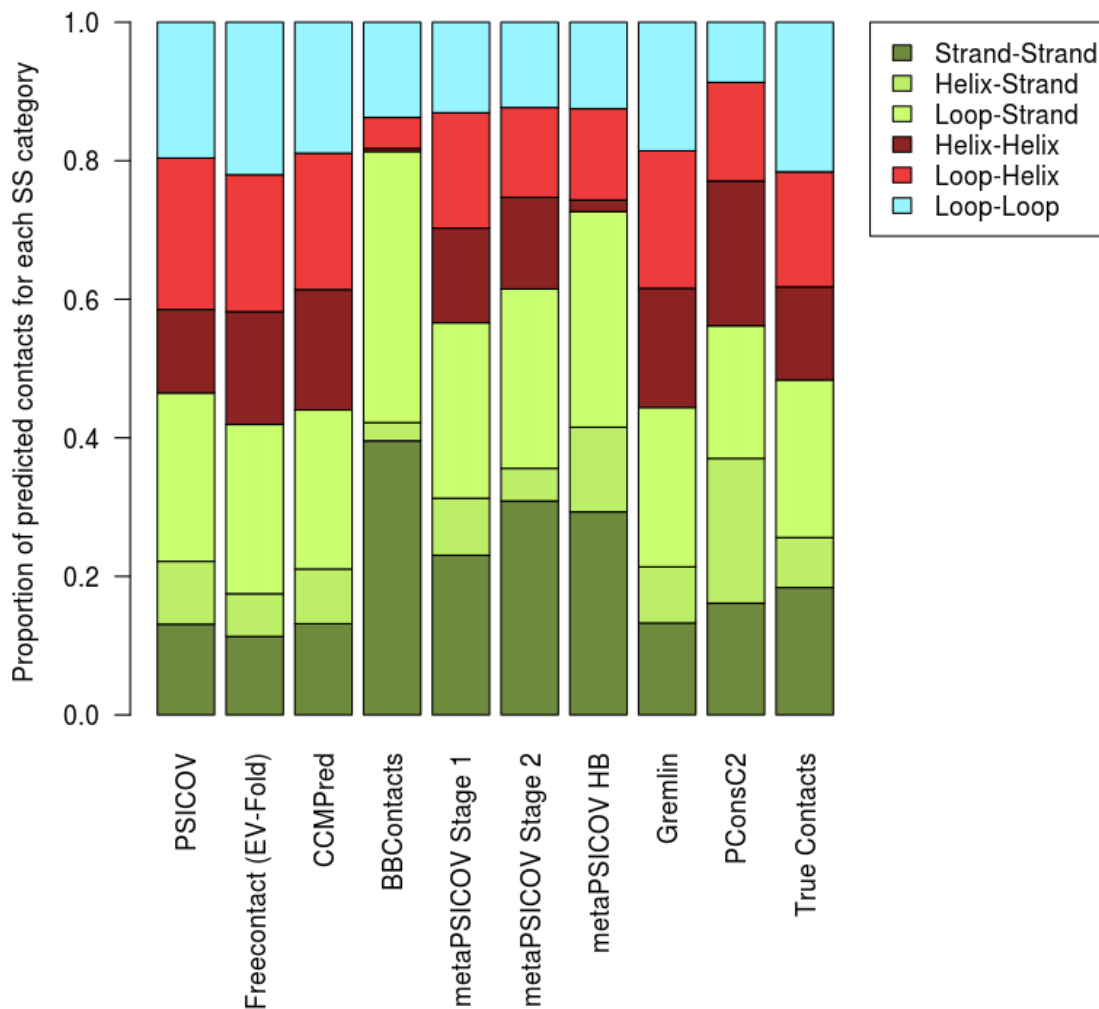


Fig. 3.8 Proportions for each SS category of top L predicted contacts as output by nine contact predictors for 41 proteins in our PDB-Representative Validation data set.

Precision of Predicted Contacts According to SCOP Class of the Target

We also assessed the precision of the predicted contacts within each of the four main SCOP classes (Figure 3.9). MetaPSICOV Stage 2 had the highest average precision for 3 out of 4 SCOP classes, PConsC2 having the higher average precision for All- α proteins. Interestingly,

the precision of all methods was significantly lower for proteins belonging to the All- α SCOP class than for other SCOP classes.

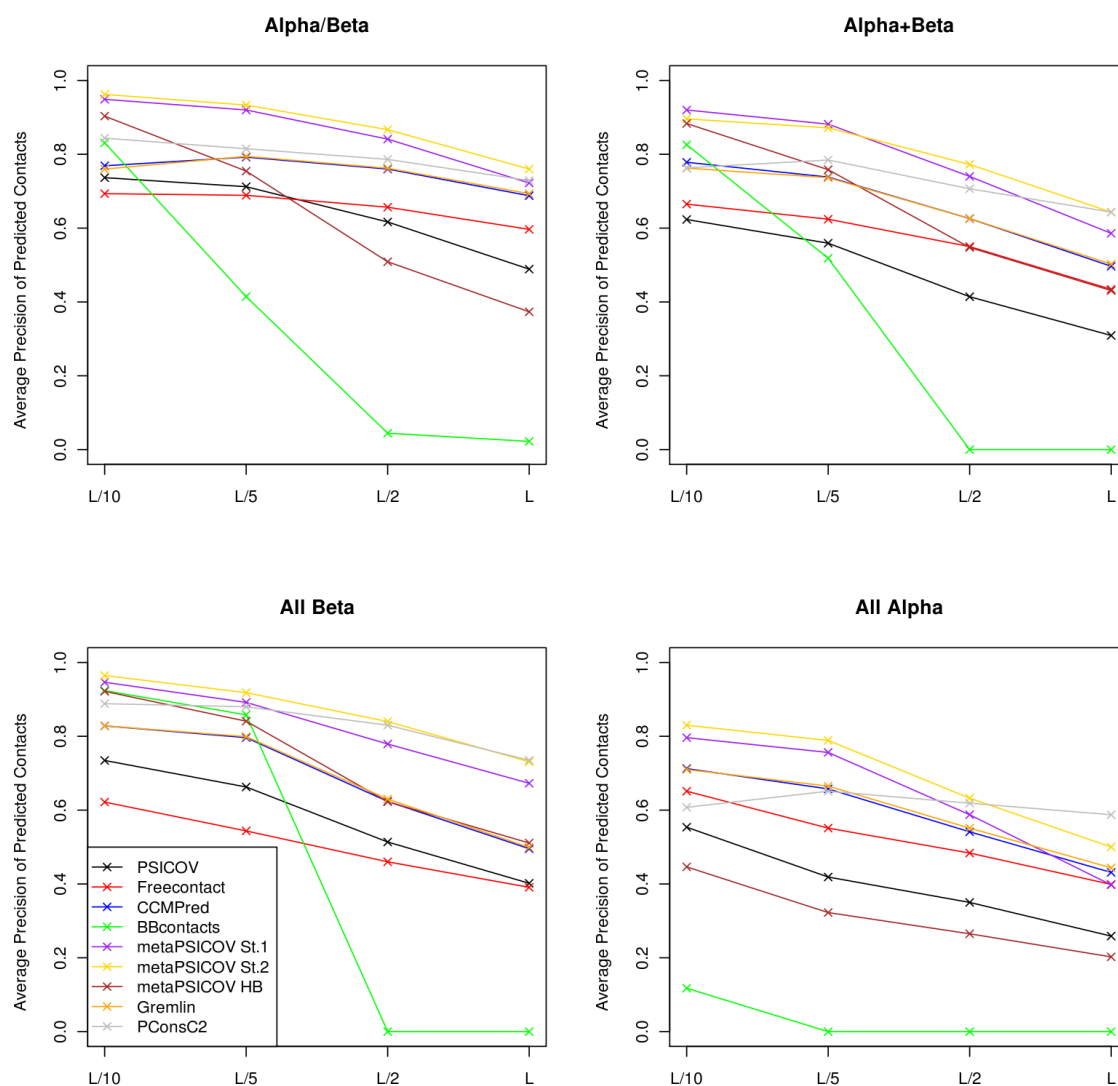


Fig. 3.9 Precision of top L predicted contacts according to target's SCOP class as output by nine contact predictors for 41 proteins in our PDB-Representative Validation data set.

Precision of Backbone Hydrogen Bond contacts

So far, we have compared the precision of different predictors to identify protein contacts. MetaPSICOV-HB, however, was designed to predict hydrogen bonds formed between back-

bone atoms. For this reason, we have compared the ability of different contact predictors to detect backbone hydrogen bonds (Figure 3.10). We have borrowed the definition of backbone hydrogen bond contacts from Jones et al. (2014), where they are defined as a backbone amino nitrogen and a backbone carboxyl oxygen that are less than $3.5\text{\AA}$ apart. As previously suggest by Jones et al. (2014), the number of protein contacts is significantly higher than the number of long-range hydrogen bonds, so we have considered only the top $L/10$ predictions for this analysis. We do not assign residues to be donors and/or acceptors, i.e. if a backbone hydrogen bond contact was predicted between i and j , either residue could be acting as a hydrogen bond donor/acceptor. Our results reveal that metaPSICOV-HB is the only method that predicts backbone hydrogen bond contacts with high precision.

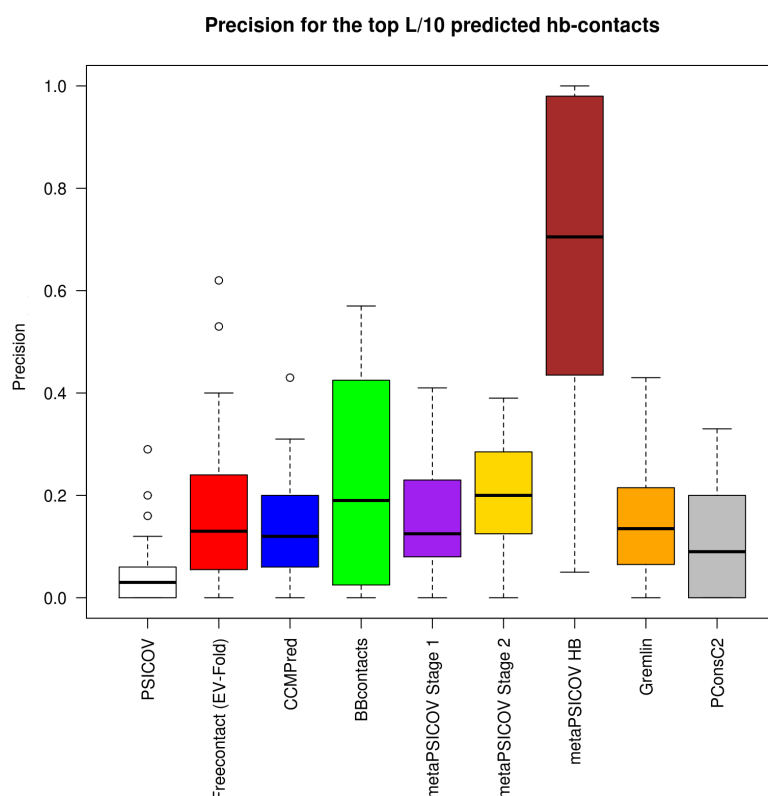


Fig. 3.10 Precision of top $L/10$ predicted backbone hydrogen bond contacts as output by nine contact predictors for 41 proteins in our PDB-Representative Validation data set.

Spread of Correct Predicted Contacts

It has been reported that the spread of correct predicted contacts is crucial for accurate topology modelling (Kim et al., 2014). To evaluate prediction spread, we use the length of the longest continuous segment of a target for which no correct predicted contacts were output. We compared the contact predictors in terms of their spread (Figure 3.11) and shown that metaPSICOV Stage 1 and CCMPred have the highest spread amongst the tested methods.

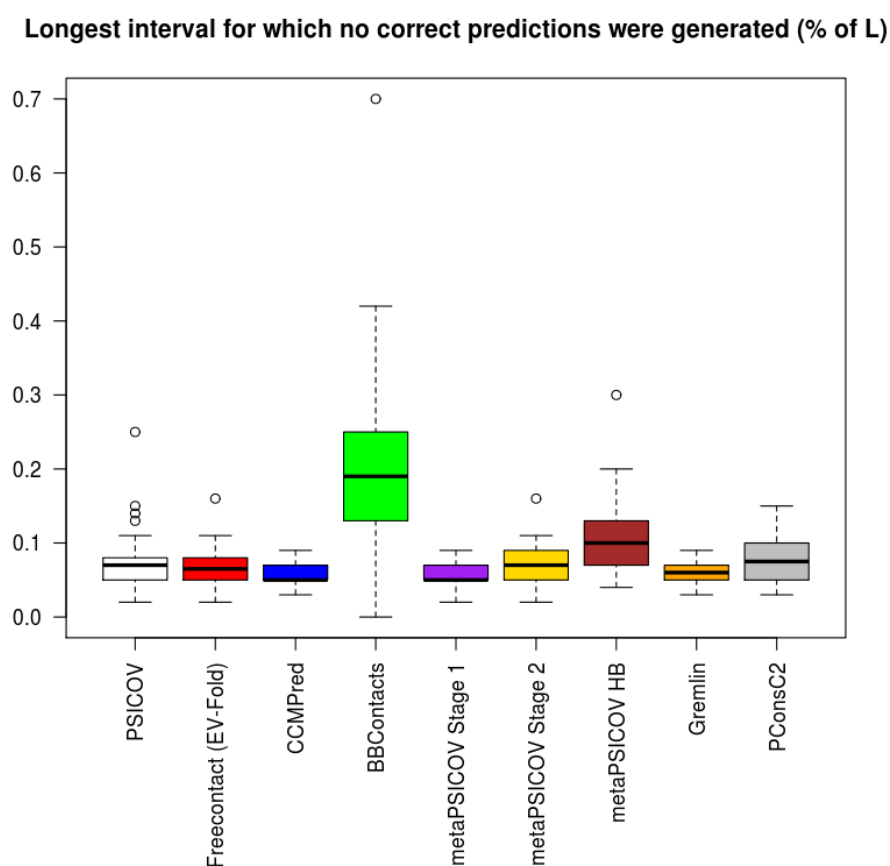


Fig. 3.11 Spread of the top L correct predicted contacts as output by nine contact predictors for 41 proteins in our PDB-Representative Validation data set.

3.3.2 Determining the Best Contact Predictor to Assist in *de novo* Protein Structure Prediction

We have used the top L predicted contacts as output by seven of the nine contact predictors to assist in *de novo* protein structure prediction using the Cotranslational Mode of SAINT2 (refer to Chapter 4 for more details). Predictions generated by GREMLIN were shown to be equivalent to the ones output by CCMPred and PConsC2 failed to produce a valid output for ten of our test cases. Therefore, GREMLIN and PConsC2 were not used in this analysis. For each contact predictor, we have used their predicted contacts with SAINT2 to generate 10,000 decoys for 41 proteins in our PDB-Representative Data Set. In total, $7 \times 41 \times 10,000 = 287,000$ decoys were generated. The TM-Score of the best decoy obtained by each method and for each protein in our data set are shown in Figures 3.12, 3.14.

An initial assessment of modelling results suggested that the best models produced by SAINT2 seemed to present similar qualities, regardless of the contact predictor used in model generation. Correct models were generated for 18 out of 41 cases and only two methods (CCMPred and metaPSICOV Stage 1) produced correct models (TM-Score > 0.5) for all 18 cases. We then assessed the difference in TM-Score Best of the best possible model against the best model produced for each of the contact predictors (Figure 3.13). Our results show that metaPSICOV Stage 1 generated a model within 0.05 TM-Score units of the best possible model for 39 of the 41 cases.

When considering the SCOP Class of the targets, the best models produced with SAINT2 + metaPSICOV Stage 1 and SAINT2 + metaPSICOV Stage 2 were better than the ones generated by SAINT2 + other contact predictors for α/β and $\alpha + \beta$ proteins (Figure 3.14). Contrary to the precision of predicted contacts, SAINT2 produced better models for All- α proteins and worse models for All- β proteins compared to other SCOP classes. For All- α proteins, a small number of correct predicted contacts seems to be sufficient for accurate model generation. This may be due to All β proteins being inherently harder to model and

All α proteins being inherently easier to model, as suggested by the quality of α -helical and β -strand fragments.

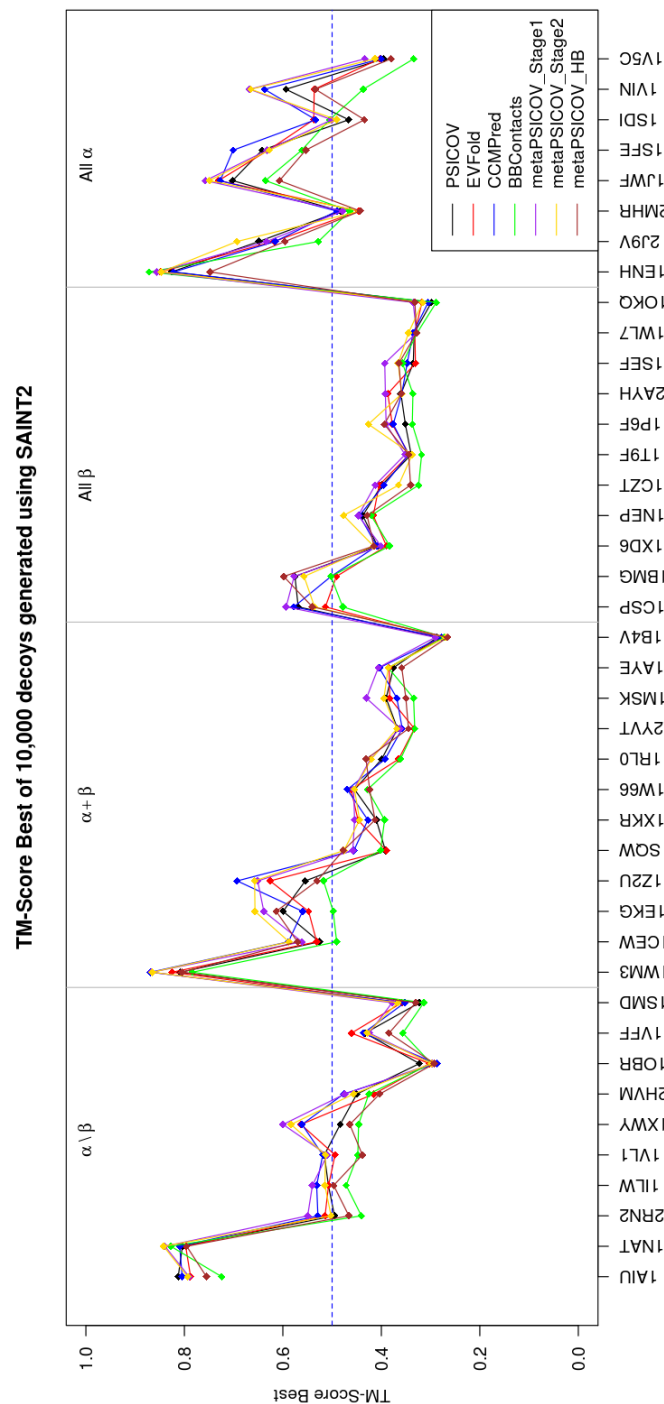


Fig. 3.12 TM-Score Best obtained using contact predictions of seven different predictors as distance constraints in SAINT2. 10,000 decoys were generated using SAINT2 Cotranslational per contact prediction method for each target in our PDB-Representative Validation Set.

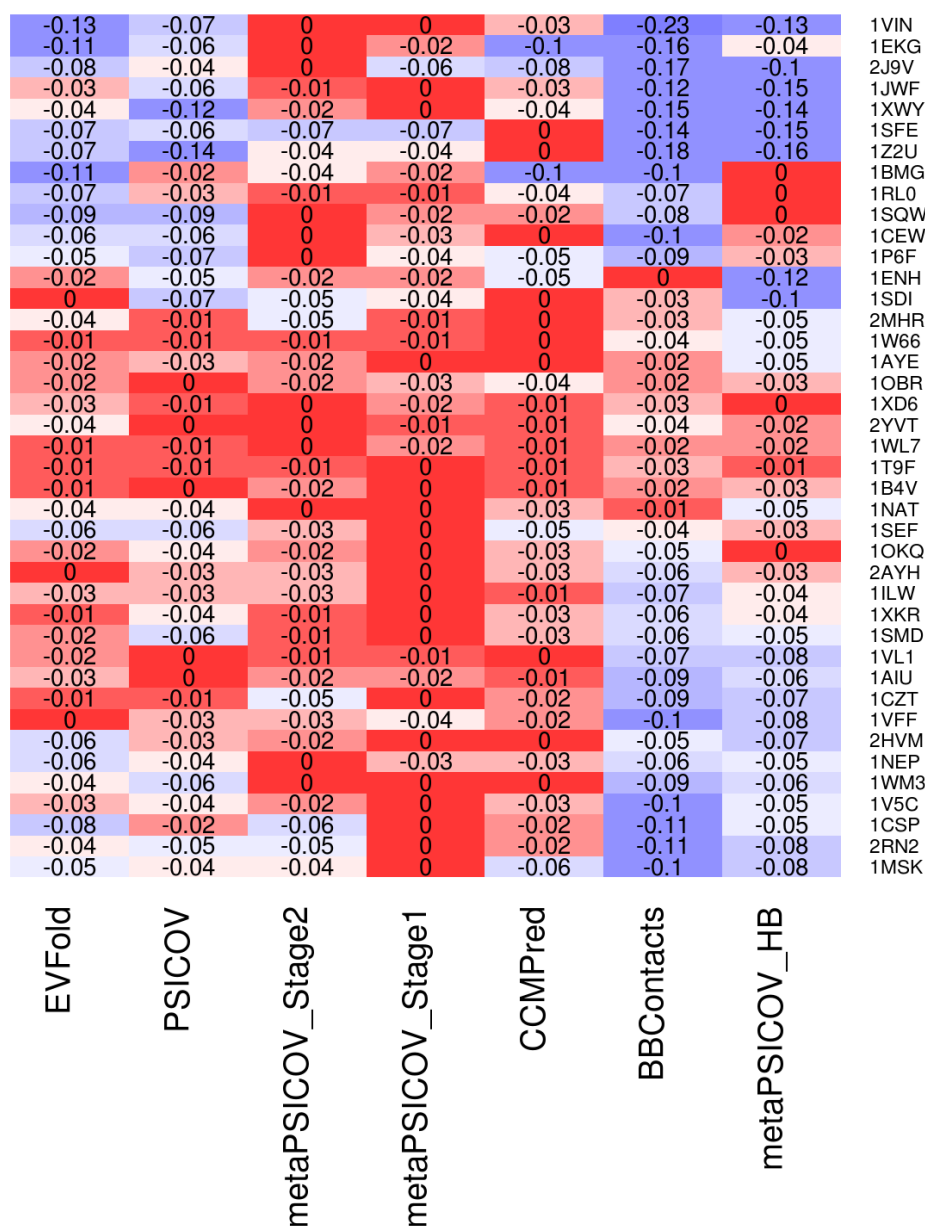


Fig. 3.13 Difference in TM-Score Best of the best possible model and the best model produced using each of seven contact predictors for the 41 proteins in our PDB-Representative Validation data set. We used SAINT2 with the predicted contacts output by each contact predictor to generate 10,000 decoys per target. The TM-Score of the best model generated amongst all contact predictors is represented by entries with a value of 0. Entries in red represent cases for which the best model produced using a given contact predictor is less than 0.05 TM-Score units away from the best model amongst all methods. Entries in blue represent the remaining cases.

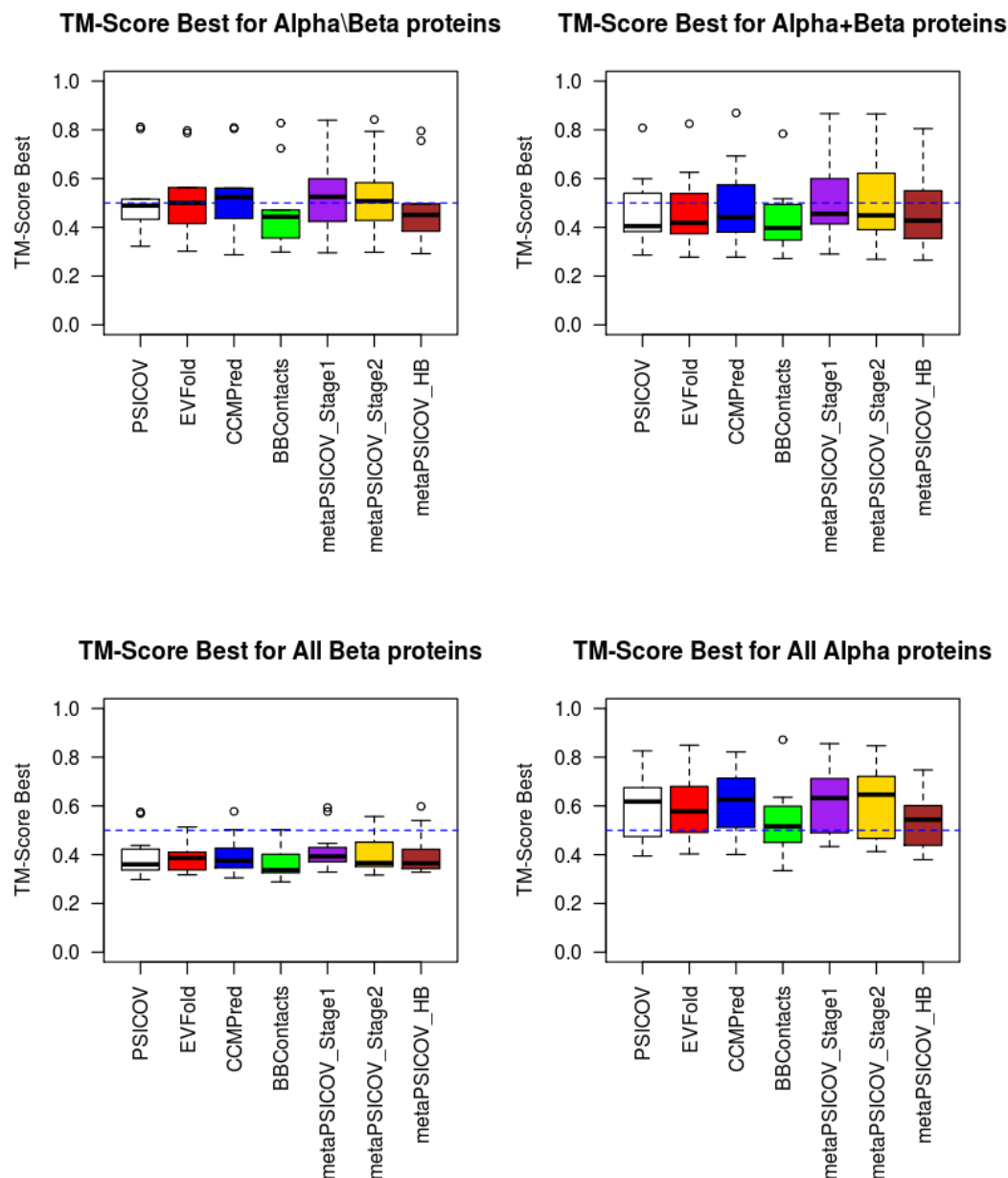


Fig. 3.14 TM-Score Best according to target's SCOP class obtained using contact predictions of seven different predictors as distance constraints in SAINT2. 10,000 decoys were generated using SAINT2 Cotranslational for each set of contact predictions and for each target in our PDB-Representative Validation Set.

3.3.3 Satisfied Predictions Contacts Correlate with Model Quality

Predicted contacts have been shown to be useful in performing quality assessment of generated decoys (Kamisetty et al., 2013). This is usually done by computing a score based on how many of the predicted contacts have been satisfied in a given model. There are two ways of considering these satisfied contacts. The first is to consider only the subset of predicted contacts that represent a true positive, i.e. predictions that represent true contacts in the native structure (satisfied **correct** predicted contacts.). The second is to consider all predictions, true contacts or otherwise, that are satisfied (satisfied predict contacts).

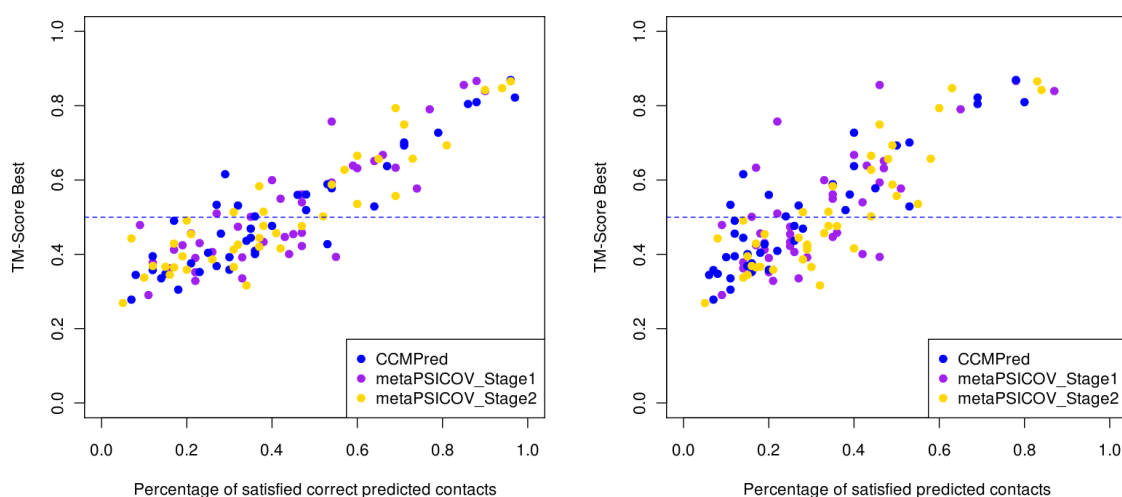


Fig. 3.15 Comparison between model quality and the number of correct predicted contacts (left) and overall predicted contacts (right) satisfied in the best model generated by SAINT2 for the 41 proteins in our PDB-Representative Validation data set. Models were produced by SAINT2 using contact predictions generated by CCMPred and metaPSICOV Stages 1 and 2.

We have assessed whether satisfied predicted contacts, correct or otherwise, show good correlation with model quality (Figure 3.15). For this analysis, we have selected only three contact predictors, CCMPred and metaPSICOV Stages 1 and 2, as their predictions led to the best models generated by SAINT2 in a majority of cases. Our results reveal that there is a strong correlation between the number of satisfied correct predicted contacts and model

quality. The correlation is slightly weaker, but still present when we consider all predicted contacts that are satisfied.

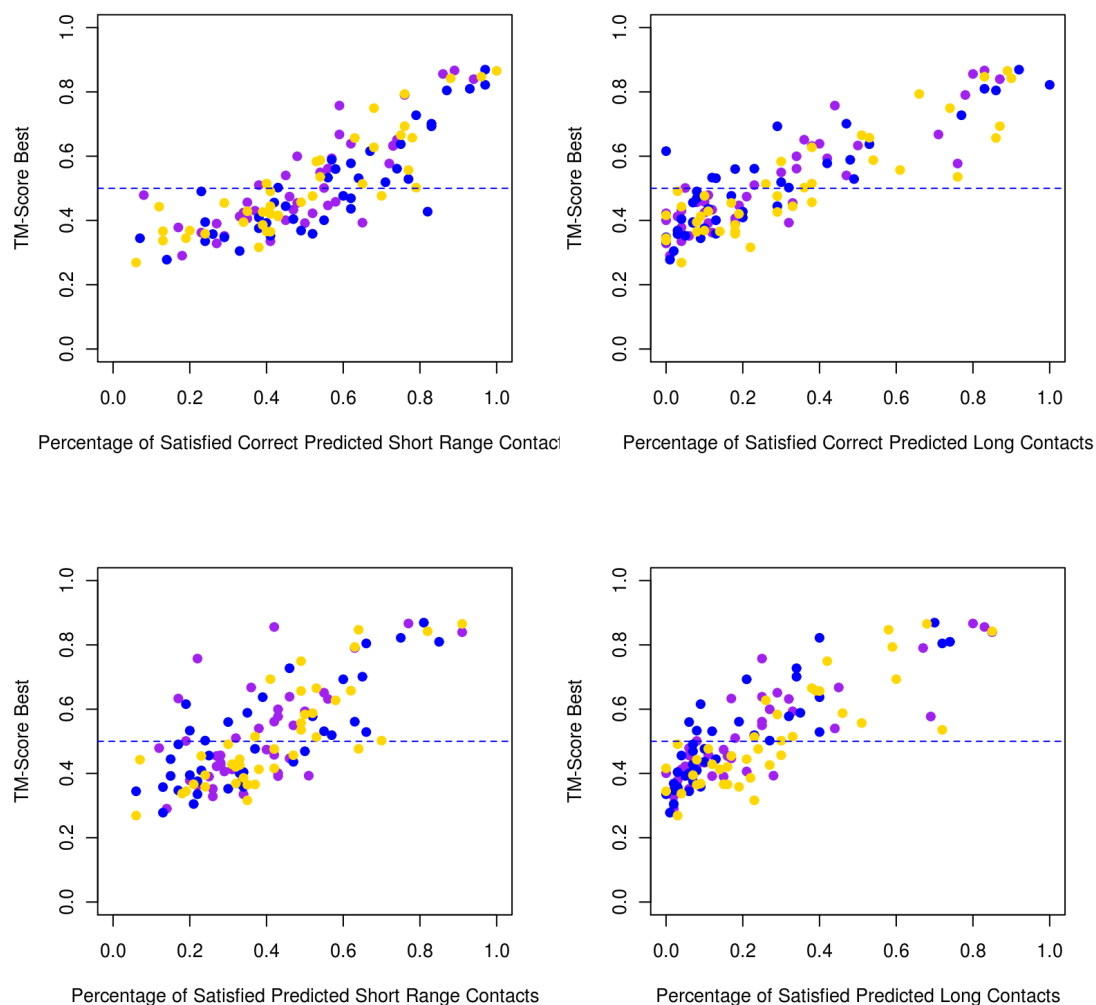


Fig. 3.16 Comparison between model quality and the number of short-range predicted contacts (left) and long-range predicted contacts (right) satisfied in the best model generated by SAINT2 for the 41 proteins in our PDB-Representative Validation data set. Results are shown for correct predicted contacts (top) and for all predicted contacts (bottom). Models were produced by SAINT2 using contact predictions generated by CCMPred and metaPSICOV Stages 1 and 2.

We have also assessed the correlation between satisfied predicted contacts and model quality in terms of short-range and long-range predicted contacts (Figure 3.16). Our results

show that long-range predicted contacts present a stronger correlation with model quality than short-range predicted contacts. This can be seen both for correct predicted contacts and for all predicted contacts.

Given that the spread of predicted contacts has also been reported to be crucial for accurate topology modelling (Kim et al., 2014), we investigated the correlation between model quality and the spread of satisfied long-range predicted contacts (Figure 3.17). Our results show no correlation between these two features, however there are few cases in which the predicted contacts are not evenly spread.

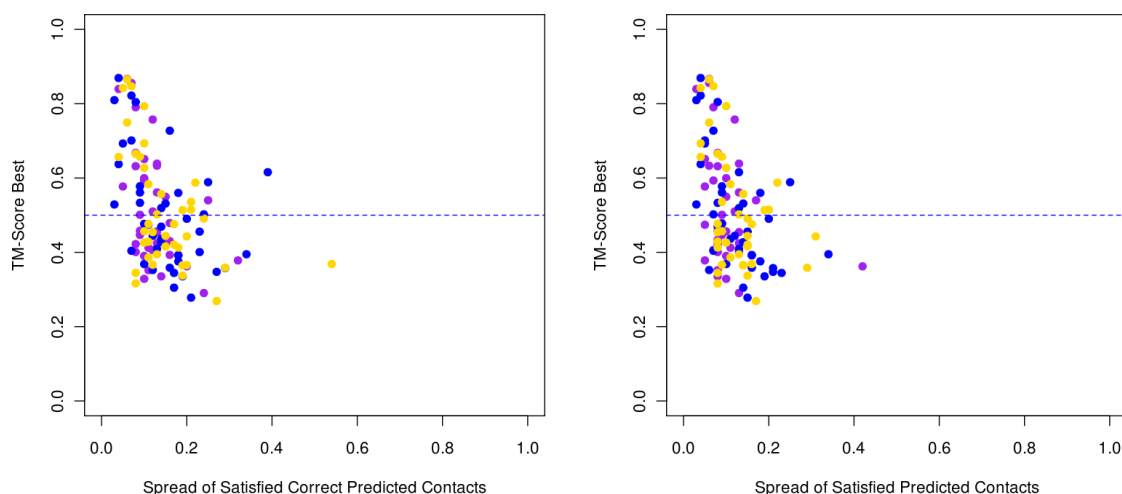


Fig. 3.17 Comparison between the spread of satisfied predicted long-range contacts as output by metaPSICOV Stage 1 and model quality of the best decoy output by SAINT2 for the 41 proteins in our PDB-Representative Validation data set.

3.3.4 Assessing the Usefulness of Predicted Contacts to Model Quality Assessment

To ascertain the effectiveness of long-range contact predictions to assist in model quality assessment, we have selected two test cases, proteins 1AIU and 2J9V, and compared the model quality of the 10,000 decoys generated by SAINT2 for each case against the number

of satisfied long-range predicted contacts as output by metaPSICOV Stage 1. We have chosen these proteins as they have similar lengths (approximately 100 residues), yet present significantly different long-range predicted contact precisions (86% for 1AIU and 40% for 2J9V). Our results suggest that we can use a cutoff of satisfied long-range predictions to enrich the final decoy ensemble with correct answers even in the case where the precision of predicted long-range contacts was low.

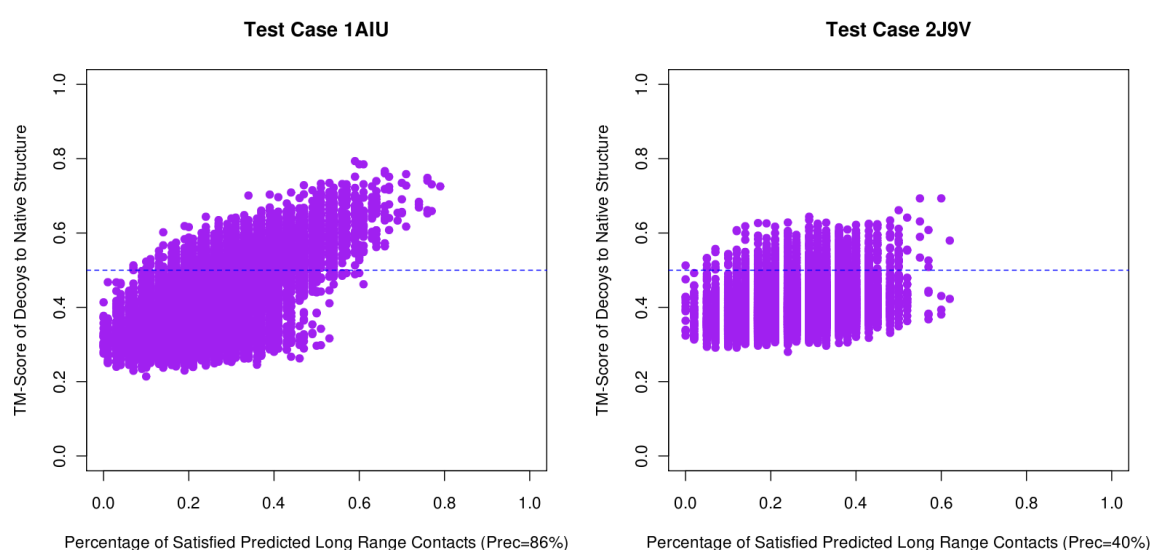


Fig. 3.18 Comparison between model quality and the number of satisfied long-range predicted contacts for 10,000 models generated by SAINT2 for proteins 1AIU (left) and 2J9V (right) using predicted contacts output by metaPSICOV Stage 1

To further investigate how long-range predicted contacts can be used to perform model quality assessment, we assessed the relationship between model quality and the number of satisfied predicted long-range contacts (as output by metaPSICOV Stage 1) for all 10,000 decoys generated by SAINT2 and for each of the 41 proteins in our PDB-Representative Validation data set (Figure 3.19). We have compared the distribution of satisfied predicted long-range contacts for two groups of decoys: correct models (TM-Score ≥ 0.5) and remaining models (TM-Score < 0.5). Our results show that a naive bayesian classifier can be built based on the proportion of satisfied predicted long-range contacts to classify decoys

between our two groups. We select a cutoff of at least 29% of predicted long-range contacts that need to be satisfied in order for a model not to be discarded from SAINT2's final ensemble.

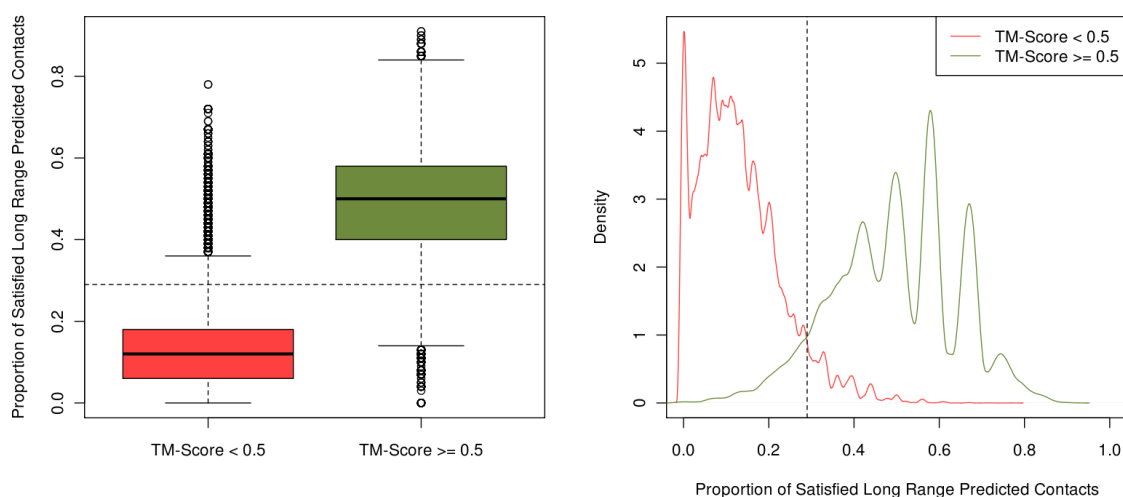


Fig. 3.19 Comparison between the proportion of satisfied long-range predicted contacts as output by metaPSICOV Stage 1 for two groups of decoys: correct models (TM-Score ≥ 0.5) and remaining models (TM-Score < 0.5). Data shown is based on 10,000 decoys generated by SAINT2 using metaPSICOV Stage 1 predictions for the 41 targets in our PDB-Representative Validation data set.

Based on our previous result, we compared the TM-Score Top-5 (see Section 4.2.3) of all decoys generated by SAINT2 for the 41 proteins in our PDB-Representative Validation data set against the TM-Score Top-5 of the subset of decoys that satisfied at least 29% of predicted long-range contacts (Figure 3.20). We have tested two ways of ranking decoys: according to SAINT2's energy function and according to the number of satisfied predicted long-range contacts. TM-Score Top-5 is the TM-Score to native structure of the best model out of the top-5 scoring models according to either ranking. There were eight cases for which no decoy satisfied more than 29% of the predicted long-range contacts and, for all of these cases, the TM-Score to native structure of the best model was less than 0.4, suggesting that we are correct to discard these results. In the remaining 33 cases, we did not observe

any significant difference as a result of discarding the decoys with low number of satisfied long-range predicted contacts. This is not surprising considering that the contact potential implemented in SAINT2 penalises decoys that do not satisfy many predicted contacts. Ranking decoys solely according to the number of satisfied long-range predicted contacts produced significantly worse results compared to ranking them according to SAINT2's scoring function.

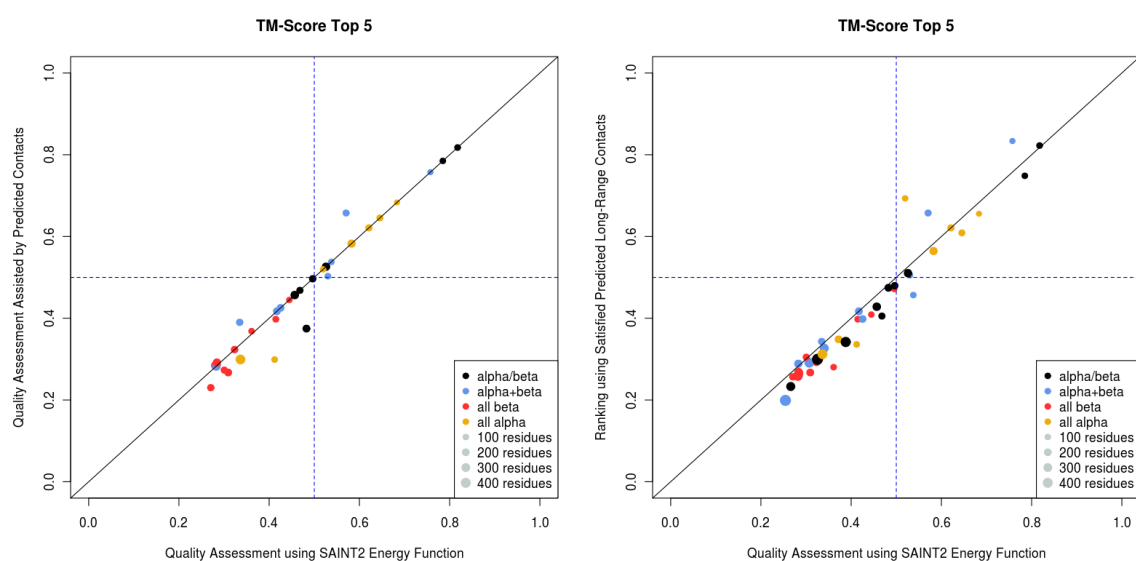


Fig. 3.20 Comparison between the TM-Score Top 5 of all decoys ranked according to SAINT2's energy function against the TM-Score Top 5 of the subset of decoys that satisfy at least 29% of predicted long-range contacts. Two ranking methods were tested for the quality assessment assisted by predicted contacts: SAINT2's energy function and ranking according to the number of satisfied predicted long-range contacts. Data shown is based on 10,000 decoys generated by SAINT2 using metaPSICOV Stage 1 predictions for the 41 targets in our PDB-Representative Validation data set.

3.3.5 Searching for evolutionary signal in predicted residue-residue contacts

It has been suggested that the score of the couplings computed by different contact predictors reflects the strength of the evolutionary signal of a given co-evolving pair of residues

(Kamisetty et al., 2013; Marks et al., 2011). Therefore, predicted contacts with higher scores are thought to be more important for protein stability and function.

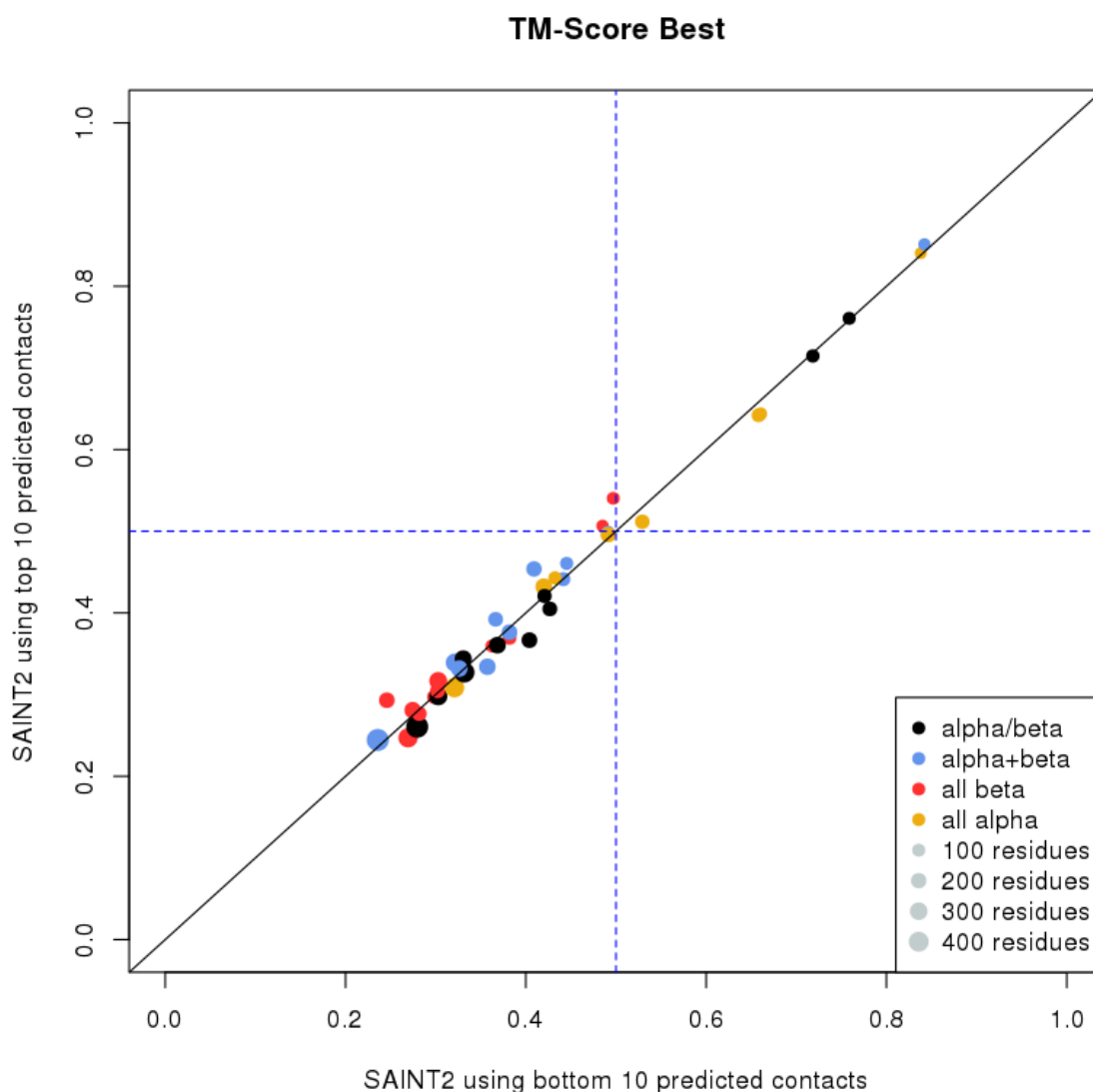


Fig. 3.21 Comparison of the best model generated by SAINT2 using the top ten and the bottom ten correct contact predictions output by metaPSICOV Stage 2. Data is shown for 10,000 decoys generated by SAINT2 for the 41 proteins in our PDB-Representative Validation data set. Proteins are coloured according to their SCOP class and the size of the points scales with protein length.

To test this hypothesis, we selected two subsets of predictions out of the correct predicted contacts output by metaPSICOV stage 2. We have chosen metaPSICOV Stage 2 for this

test given that it had the best overall precision amongst all tested contact predictors. The first subset contains the ten correct predicted contacts with the highest scores amongst the predicted contacts (top 10), whereas the second subset contains the ten correct predicted contacts with the lowest scores amongst the predictions output by metaPSICOV stage 2 (bottom 10). We then used SAINT2 to generate decoys using each subset of predicted contacts. For each subset, we generated 10,000 decoys for the 41 proteins in our PDB-Representative Validation data set. Our results show no significant difference between the best models produced by SAINT2 using the top 10 correct predicted contacts and the bottom 10 correct predicted contacts output by metaPSICOV Stage2.

3.4 Discussion

Our results show that metaPSICOV Stage 2 has the highest average precision amongst tested methods. As expected, we observe a decrease in precision for all methods when more contact predictions are considered. It is important to highlight that bbcontacts, which is designed to predict contacts between β -strands, and MetaPSICOV-HB, which is designed to detect residues that form backbone hydrogen bonds, aim to output a smaller number of predictions. Both methods present the biggest variation between the precision of the top $L/10$ and top L predicted contacts.

Given that long-range contacts have been reported to be more useful for *de novo* protein structure prediction, we have reassessed the precision of each contact predictor in terms of short-range and long-range contacts. MetaPSICOV Stage 1 has the highest precision for long-range predicted contacts and its long-range predicted contacts are overall more precise than its short-range predicted contacts. Yet, metaPSICOV Stage 1 outputs a smaller proportion of long-range contacts when compared to other methods such as metaPSICOV Stage 2 and PConsC2. Our results also show that PSICOV tends to output more long-range

predicted contacts than short-ranged ones and its proportion of long-range contacts was the highest amongst all methods.

We have provided a rough estimate of how many sequences are required by each contact predictor to obtain a precision of at least 50% for their top- L predicted contacts. This estimate was based on a linear regression using the predicted contacts computed by each method for our PDB-Representative Validation data set. The estimate suggests that the contact predictors PConsC2 and metaPSICOV Stage 2 can generate useful predictions using MSAs with as few as 200 sequences. Several assumptions that may not have been satisfied were made in order to perform the linear regression and the sample size is too small to generalize these results. However, our findings still show that the estimate of the proportion of *de novo* modelling cases that can benefit from contact predictions (Kamisetty et al., 2013) may need to be re-evaluated.

In our analyses, we observed that all methods present a significantly lower precision for predicted contacts output for proteins belonging to All α SCOP class. This reflects the lower precisions observed for predicted contacts between α -helical residues (helix-helix SS category). We hypothesize that the lower precision occurs because α -helical residues cannot form backbone hydrogen bonds other than the ones formed with adjacent residues in the helix (which are explicitly not considered in this analysis). According to this hypothesis, the strength of evolutionary signal for α -helical contacts is weaker, which would explain why they are harder to predict.

Our results suggest that there is room for improvement in contact predictions for α -helical proteins, potentially using similar strategies to the ones employed by bbcontacts. Additional information such as the frequencies of contacts between α -helices or the distribution of the number of contacting helices observed in the PDB could be used to improve these predictions further. Bbcontacts generates a high proportion of short-range predicted contacts formed mostly by at least one β -strand residue. Although the precision of these predictions is, on

average, lower than the precision of predictions output by methods such as metaPSICOV Stage 2, the concept of bbcontacts highlights the potential to further extend existing predictors to improve their precision for contacts predicted between residues of a specific secondary structure.

We have also shown that metaPSICOV-HB can successfully predict backbone atoms that form hydrogen bonds. This result highlights the potential of co-evolution methods to translate sequence information into accurate distance constraints. MetaPSICOV-HB achieves, on average, more than 50% precision for backbone hydrogen bond contacts, which suggests that these distance constraints can be used to improve *de novo* protein structure prediction.

When looking at the spread of correct predicted contacts, we observed that PConsC2 and metaPSICOV Stage 2 present a lower spread compared to other approaches. Both methods rely on the notion that a contact is much more likely to occur next to another contact in a (predicted) contact map, either explicitly (in the case of PConsC2) or implicitly as features of a neural network (in the case of metaPSICOV Stage 2). Given that a fixed number of predictions is often used in model generation (i.e. the top L predictions), enriching the final contact prediction in this manner is likely to reduce the spread of the top-L contacts.

We have used the contact predictions output by seven of the methods to assist SAINT2 Cotranslational in generating 10,000 decoys for each protein in our PDB-Representative Data Set. Our results show that metaPSICOV Stage 1 can be used to produce models that are consistently as good as the best models produced amongst all methods. This suggests that the spread of correct predicted contacts and the precision of predicted long-range contacts are also important for efficient model generation.

When considering the SCOP class of the targets, SAINT2 failed to produce good models for a majority of All- β proteins using any set of predictions, despite the fact that contact predictions for this SCOP class had the highest precision out of all four classes. This reveals

a limitation of SAINT2 concerning the accurate prediction of All- β protein structures, an issue that will need to be addressed in the future.

Our results show that the number of satisfied long-range predicted contacts correlates well with the quality of models generated by SAINT2. We explored this information to discard decoys that do not satisfy more than 29% of long-range predicted contacts. Discarding these decoys did not lead to any significant change in the quality of the final five decoys (TM-Score Top 5) selected in our protein structure prediction pipeline. Using this cutoff, we were able to detect eight modelling cases (out of 23) for which no correct answers were produced. In these cases, the modelling routine is failing to sample the conformational space efficiently, perhaps due to a poorly constructed fragment library. We have shown that predicted contact information can assist in model quality assessment, particularly in the identification of bad models.

If the strength of couplings computed during contact prediction correlates with the importance of a given contact to protein stability and function, we would expect higher scoring correct predicted contacts to be more useful for protein structure prediction than lower scoring correct predicted contacts. We have tested this evolutionary signal hypothesis by comparing models generated using the top ten and the bottom ten correct contact predictions as generated by metaPSICOV Stage 2. Our results reveal no significant difference between the models generated using either set of contacts, which suggests that the aggregated effect of multiple predicted contacts is far more important for model generation than individual signals/strength of couplings.

With the analyses described in this chapter, we have assessed the usefulness and the limitations of incorporating predicted contact information into a *de novo* protein structure prediction pipeline. Overall, we selected metaPSICOV Stage 1 as the contact predictor of choice to assist SAINT2 in generating models. We also identified SAINT2's limited ability to model All β proteins, given that modelling fails even for the cases where contact predictions

are very precise. Moreover, our results provide insights into how we will be able to improve our modelling results using predicted backbone hydrogen bond contacts and performing quality assessment mediated by predicted contacts.

Chapter 4

Biologically Inspired *de novo* Protein Structure Prediction: Methods

Software Availability and Authorship: the protein structure predictor SAINT2 was implemented by Saulo H.P. de Oliveira and will be made freely available after this work is published.

4.1 Introduction

Protein structure determination is expensive and may present technical difficulties in some cases (Terwilliger et al., 2009). Protein sequence information, on the other hand, is readily available and has grown exponentially in recent years (Kahn, 2011). A large number of tools have been developed to predict protein structure from sequence (see Moult et al. (2014, 2011a) for examples). These structure prediction methods are commonly categorized into two groups: template-based and template-free. In this Chapter, we will focus on template-free approaches, also known as *de novo* protein structure prediction.

In *de novo* structure prediction, model generation is treated as a global optimization problem, where heuristics are used to search the conformational space for a global energy

minimum (Baker and Sali, 2001). Despite efforts to improve structure prediction, *in silico* models generated *de novo* do not achieve experimental quality/resolution for a large number of cases (see Moult et al. (2014, 2011a)).

One of the leading methodologies for *de novo* protein structure prediction is known as fragment-assembly. In fragment-assembly protein structure prediction, a library of structural fragments is compiled for each position of a target protein. As the fragments in the library are taken from existing protein structures, the structures that are sampled for a given residue are those that are observed in nature. Generally, fragment-based approaches use a scoring function is thought to be representative of the energy landscape involved in protein folding and these methods attempt to find the conformation at the global minimum of this scoring function.

There are two main difficulties encountered during this process: the size of the conformational space and the imprecision of the scoring function. The conformational space, even with the use of fragment libraries, is so large that it cannot be explored even when powerful search heuristics and high computational resources are employed (Bradley et al., 2005). In terms of the scoring function, the potentials used are only estimations and are, therefore, imprecise (Kinch et al., 2011). In order to address these limitations, we make use of the mechanisms of the folding process *in vivo*.

However, existing tools are not designed to be investigative. *De novo* protein structure prediction software prioritize accuracy, i.e. they are implemented to generate an answer as close to the crystal structure as possible (Jones and McGuffin, 2003; Raman et al., 2009; Zhang et al., 2005). Initiatives such as CASP (Moult et al., 2014) and CAMEO (Haas et al., 2013) encourage this mentality. Here, we want to develop software that is accurate for structure prediction, but is also flexible enough to be able to investigate more information from the *in vivo* folding process.

When proteins fold *in vivo*, they also have to sample from a large combinatorial space, but are capable of folding efficiently and accurately. In order to better understand how proteins restrict the search space and attain their native-state conformation, it may be important to consider the biological context under which proteins fold. One hypothesis is that proteins fold as they are being translated, which is known as cotranslational protein folding (see Chapter 5).

SAINT (Sequential Algorithm Initiated at the Nitrogen Terminus) was originally developed to investigate the impact of such cotranslational protein folding on *de novo* protein structure prediction (Ellis et al., 2010b). SAINT was implemented as a modified version of ROSETTA (Raman et al., 2009) that performed structure prediction in a cotranslational fashion. It combined the heuristic fragment-based approach with a simulation, in which the peptide is gradually extruded from a virtual ribosome. Modelling results obtained using the SAINT showed that directional predictions starting from the nitrogen terminus generate better results than sequential predictions beginning in the carbon terminus (Ellis et al., 2010b). Nonetheless, SAINT was constrained by ROSETTA's complex code architecture, limiting its investigative power.

In order to fully investigate how cotranslational protein folding can be used to improve structure prediction, we have developed SAINT2, a completely independent cotranslational fragment-assembly structure predictor. SAINT2 differs from conventional fragment-assembly approaches because it performs predictions sequentially, in a cotranslational fashion. SAINT2's fragment libraries are able to be composed of fragments of different lengths, unlike the original SAINT. The effect of translation kinetics in protein folding can be explored using different move distributions. Furthermore, using SAINT2 we can calculate the effects of individual scoring potentials and how these effects vary with protein length.

In this chapter, we will describe the implementation and development of SAINT2, a biologically inspired *de novo* protein structure prediction software. We compared SAINT2

in its Cotranslational mode against ROSETTA for a Training set of 43 structurally diverse proteins. Our results show that SAINT2 generates correct answers for 24 cases and ROSETTA for 19 of 43 cases. No correct answer was produced by ROSETTA for proteins longer than 200 residues, whereas SAINT2 generated correct answers for 5 of these 21 longer proteins. We have also participated in the latest iteration of the Critical Assessment of protein Structure Prediction (CASP11) and shown that an initial version of SAINT2 produced correct models for 2 out of 13 cases, despite the fact that a reduced number of decoys was generated.

4.2 Methods

4.2.1 Sequence-to-Structure Pipeline

We have implemented a sequence-to-structure *de novo* protein structure prediction pipeline (Figure 4.1). Our protocol's first step is to predict the target's secondary structure and torsion angles from its sequence. We then generate a target-specific fragment library using Flib (see Chapter 2) and metaPSICOV (Jones et al., 2014) to predict residue-residue contacts for the target (see Chapter 3). All of this information is then fed into SAINT2, our cotranslational *de novo* structure prediction software, which performs structural modelling.

Secondary Structure and Torsion Angle Prediction

The first step in our sequence-to structure pipeline is the prediction of secondary structure using the target's sequence. To predict secondary structure (SS), our pipeline uses PSIPRED V3 (Jones, 1999) coupled with NCBI BLAST+ v2.2.27 (Camacho et al., 2009). The target's sequence is used as input for PSI-BLAST (Altschul et al., 1997), which generates a multiple sequence alignment (MSA) using the following parameters:

- Inclusion e-threshold = 0.001.

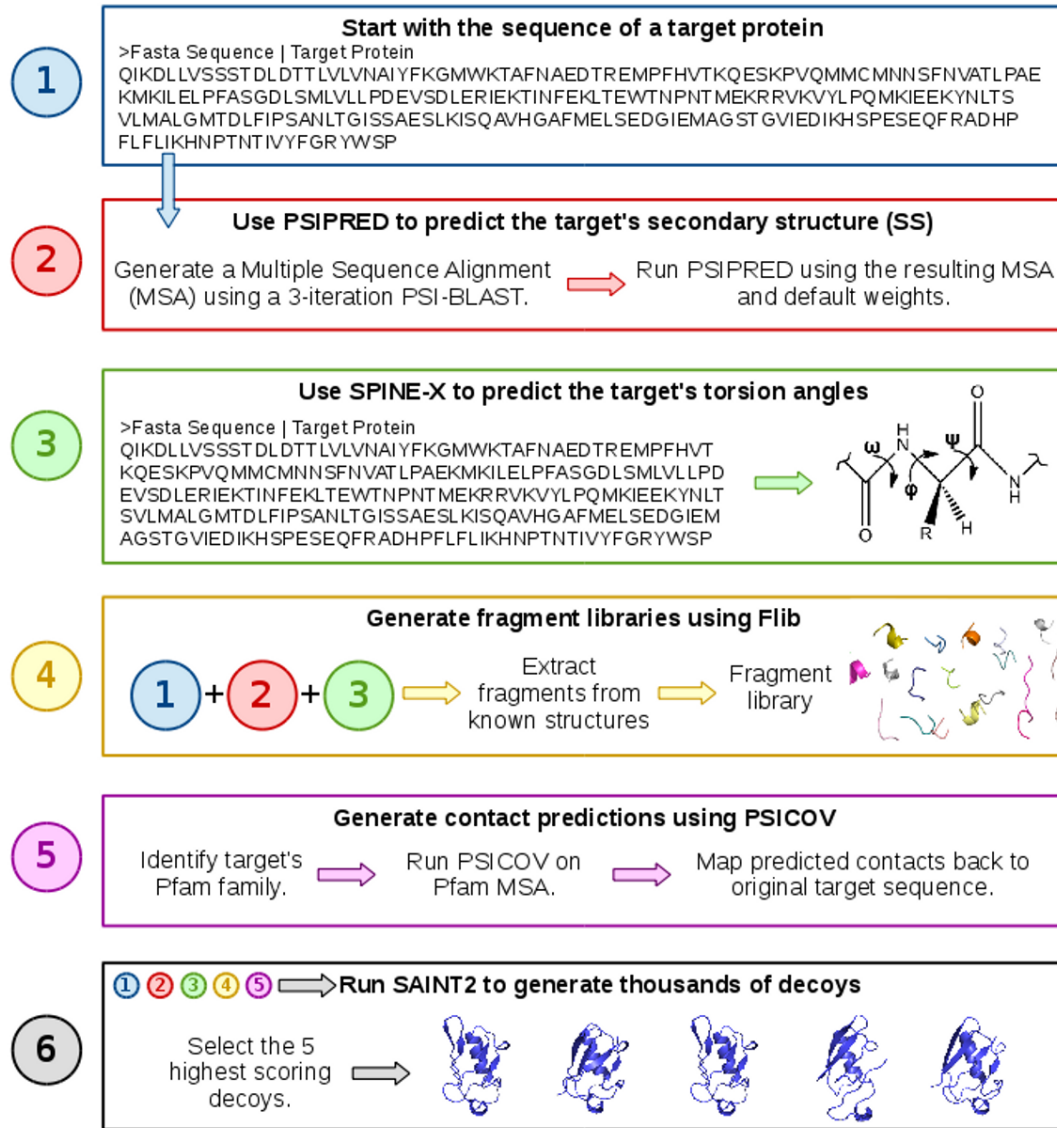


Fig. 4.1 In order to generate the files required for model generation, we embedded our software, SAINT2, into a sequence-to-structure pipeline. Our protocol takes as input a target sequence, which is then used to generate secondary structure predictions using PSIPRED (Jones, 1999), torsion angle predictions using SPINE-X (Faraggi et al., 2009, 2012), a fragment library using Flib (see Chapter 2 for more details), and contact predictions using metaPSICOV (Jones et al., 2014). The final step in our pipeline is to generate structure models using SAINT2. SAINT2 requires the output files of steps one to four to generate models. Step five is optional.

- Number of iterations = 3.

PSIPRED is then used to generate the SS predictions using the resulting MSA and default weights. Spine-X (version 2.0) (Faraggi et al., 2009, 2012) is used to perform torsion angle predictions based on the sequence of the target.

Fragment Library Generation

SAINT2 relies on a fragment library to reduce its conformational search space. For each target, we generate a custom fragment library using Flib (de Oliveira et al., 2015). This process is described in detail in Chapter 2. Fragments in this library are represented by a set of six angles per residue: three torsion angles (ϕ , ψ , and ω), and the three bond angles (between the main chain atoms N, C- α and C). For example, a nine-residue long fragment would be represented by $6 \times 9 = 54$ angles.

We used Flib (de Oliveira et al., 2015) to generate the fragment libraries used in SAINT2. Flib extracts fragments from a curated database of known structures. This database is a subset of the PDB (Berman et al., 2000) that is non-redundant (sequence identity $< 90\%$) and contains only high quality structures (resolution $< 2.5 \text{ \AA}$). HHSearch (Söding, 2005) is used to identify and remove homologues to the target from this database in order to represent a realistic *de novo* structure prediction scenario. Fragments are selected from structures in the database based on the target's sequence, predicted secondary structure and predicted torsion angles. On average, Flib generates ≈ 30 fragments per target position. The fragments can be 6 to 20 residues long.

Residue-Residue Contact Predictions

We use metaPSICOV (Jones et al., 2014) to generate residue-residue contact predictions for the target (see Chapter 3). metaPSICOV is a meta-predictor that combines inferred correlated mutations from three distinct approaches: PSICOV (Jones et al., 2012), mean field DCA as

implemented in the FreeContact package (Kaján et al., 2014), and CCMPred (Seemayer et al., 2014). No contact predictions are generated for MSAs that contain less than 200 sequences, as the precision for such cases has been reported to be low (Jones et al., 2014).

Once the predictions have been output, we map the predicted contacts back to the original sequence. Contact predictions that map back to a gap in the original sequence are deleted. We only consider predictions with an estimated precision (PPV) of 0.5 or higher.

4.2.2 Training Data Set

The Training Data Set used in this chapter was previously described in Table 2.1. It is comprised of 43 structurally diverse proteins extracted from the PDB. These proteins are all single chain, single domain proteins proportionally distributed into the four SCOP (Murzin et al., 1995) protein classes: all α , all β , α/β , and $\alpha + \beta$. They are also evenly spread in terms of length, ranging from 50 to 500 residues.

4.2.3 Validation

TM-Score (Xu and Zhang, 2010; Zhang and Skolnick, 2004) is used to evaluate the quality of the decoys generated by SAINT2. A TM-Score of 0.5 or higher is considered to be a correct answer (Xu and Zhang, 2010). Three different TM-Score based measures were defined to help assess our results:

- TM-Score of the best decoy (TM-Score Best): this measure is computed by selecting the decoy with the highest TM-Score compared to the target's native structure. This decoy may not necessarily be output as a prediction by SAINT2.
- TM-Score of the Top 5 decoys (TM-Score Top-5): this measure is the TM-Score of the best decoy amongst the five decoys output by SAINT2.

- Average TM-Score: this measure is the average TM-score for a given decoy population produced by SAINT2.

4.2.4 The Framework of SAINT2

SAINT2 is a fragment-assembly *de novo* structure prediction method. SAINT2 samples the conformational space using a library of fragments of known structure (Figure 4.2). Its energy function is a combination of several different knowledge-based potentials (see Section 4.2.9).

4.2.5 Peptide/Protein Representation within SAINT2

SAINT2 represents its 3D models as 3D Cartesian coordinates of the five main backbone atoms (C, N, O, C- α and C- β). These coordinates are calculated using the fragment library (see Chapter 2) and completed using ideal bond lengths and angles. SAINT2 does not currently consider side-chains explicitly.

4.2.6 SAINT2 Run Modes and Initialization

Three different modes have been implemented within SAINT2: Cotranslational, *In vitro*, and Reverse (Figure 4.3).

SAINT2 Cotranslational mode mimics the sequential aspect of cotranslational protein folding (see Chapter 5). The routine is initialized by selecting a fragment from the fragment library corresponding to the N-terminal residues of the target protein. In this mode, the peptide will grow as the simulation is executed. The direction of peptide extrusion is N-terminus to C-terminus.

The Reverse mode is analogous to the Cotranslational mode, but the initialization occurs in the opposite terminus (the C-terminus). In the Reverse mode, the peptide will also grow as the simulation is executed, but the direction of peptide extrusion is reversed (C-terminal to N-terminal).

SAINT2 Cotranslational Mode Framework

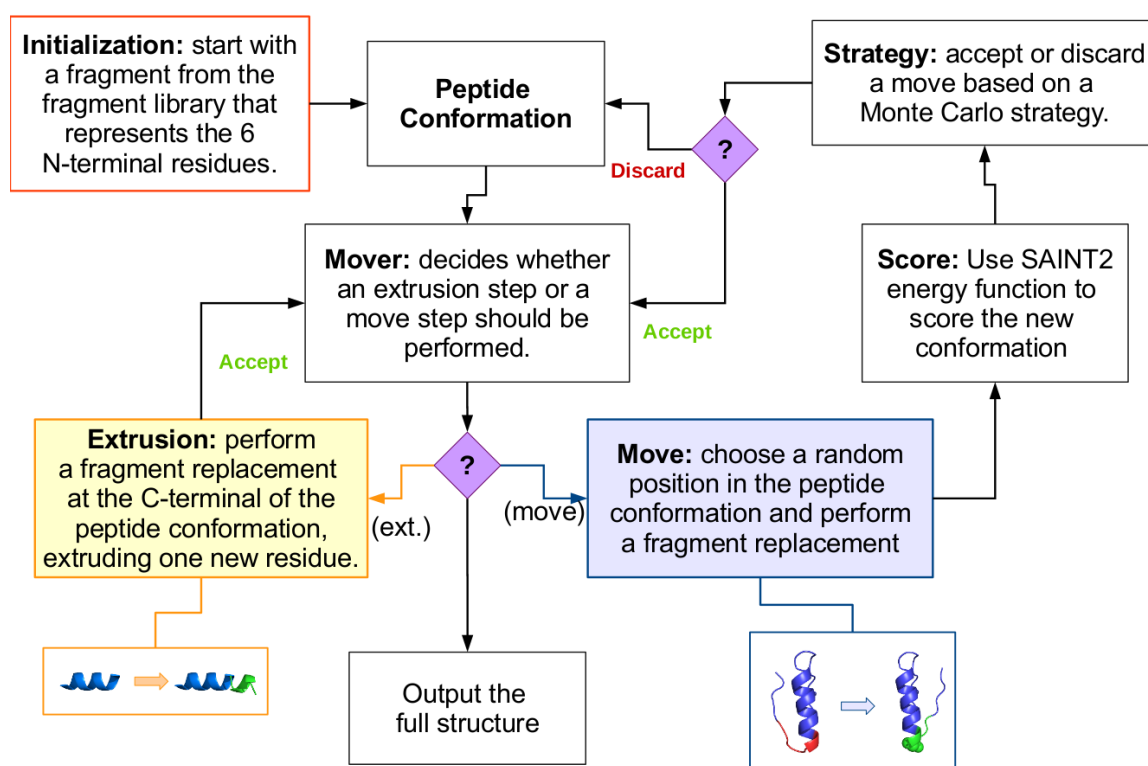


Fig. 4.2 The framework of the Cotranslational mode of SAINT2. The process is initiated by the selection of fragment from the library for the conformation of the six N-terminal residues. The Mover then decides whether moves or extrusions should be performed. In an extrusion, the peptide chain is extended by adding a fragment from the library to the C-terminus of the model. In a move, a fragment replacement is performed at a random position in the peptide. The resulting conformation is scored and either accepted or discarded according to a Monte Carlo strategy. Once the peptide has been fully extruded and the required number of moves has been performed, the full peptide conformation is output. This process generates a single decoy and, in practice, it is repeated thousands of times to generate a population of decoys.

SAINT2 can also perform fragment-assembly in a similar fashion to traditional approaches such as ROSETTA (Bonneau et al., 2002, 2001; Bradley et al., 2005; Raman et al., 2009). We refer to this as *in vitro* structure prediction, as it resembles the re-folding that occurs *in vitro* after protein denaturation. In the *In vitro* mode, SAINT2 is initialized with a fully extruded and fully elongated protein conformation. All the torsion angles are set to 180 degrees and ideal bond lengths and angles are used.

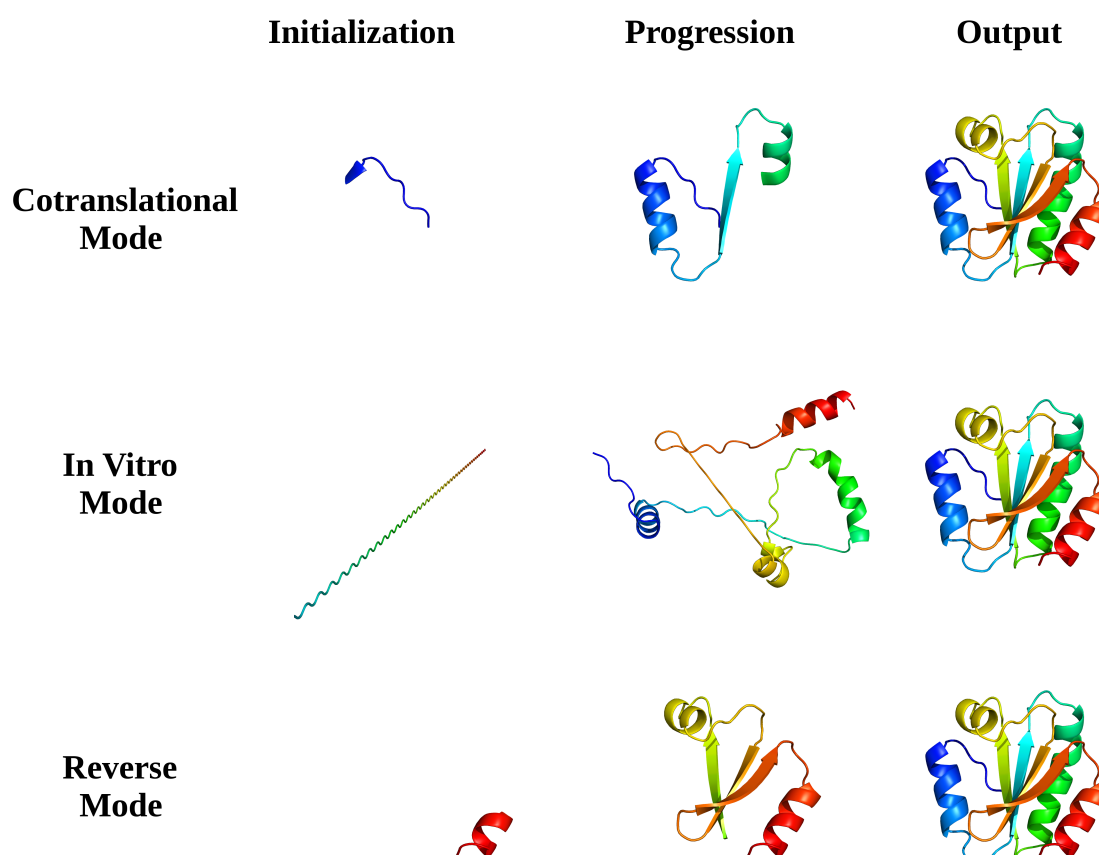


Fig. 4.3 Representation of the three modes implemented within SAINT2. The Cotranslational mode begins with residues representing the N-Terminal and grows the peptide as the simulation is executed. The Reverse mode is analogous to the Cotranslational mode, but predictions start at the C-Terminal and growth direction is reversed. The *In vitro* mode, based on traditional approaches, begins with a fully elongated sequence.

4.2.7 Fragment Replacements

During a fragment replacement step, fragments are selected at random from the fragment library. The probability of selecting a given fragment is proportional to the fragment score assigned by Flib, which is based on the Predicted Torsion Angle score described in Section 2.2.4. Unlike traditional approaches, SAINT2 employs two different types of fragment replacement: Extrusion steps and Move steps.

Extrusion Steps

These steps are a specific type of fragment replacement that always takes place at the end of the existing conformation, growing the peptide by one residue (Figure 4.4). In Cotranslational mode, an extrusion always occurs at the C-terminal end of the current peptide, in which a fragment representing the C-terminus is randomly selected from the fragment library. For the Reverse mode, extrusion occurs in an analogous fashion, but at the N-terminal end of the peptide. The new conformation resulting from an extrusion step is always accepted. No extrusions are performed for the *In vitro* mode, as in this mode the initial conformation already contains all the residues of the target.

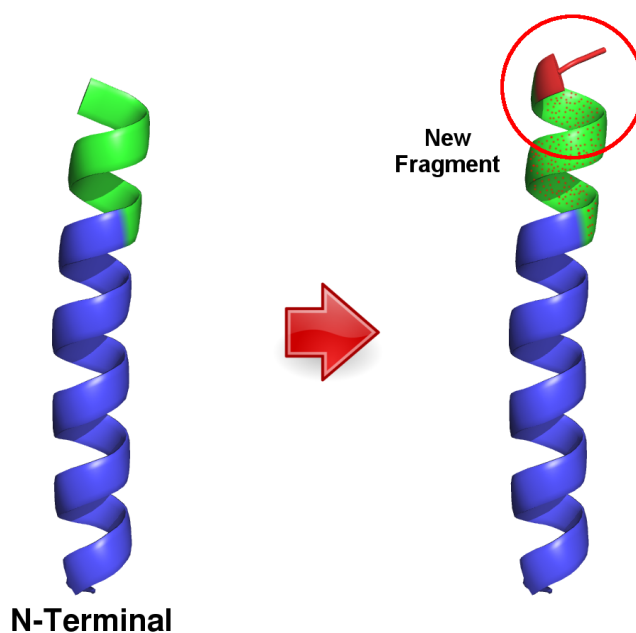


Fig. 4.4 Representation of an extrusion step of SAINT2 Cotranslational mode. An Extrusion step takes place in one of the termini of the target (C-Terminal for Cotranslational mode or N-Terminal for Reverse mode) and it grows the current peptide conformation by one residue.

Move Steps

These steps are fragment replacements that take place at random positions in the existing peptide conformation (Figure 4.5). Unlike extrusion steps, move steps do not append new

residues at the end of the sequence. Move steps are used in all of SAINT2 modes. Move steps can be accepted or discarded dependent of their score (see Section 4.2.20).

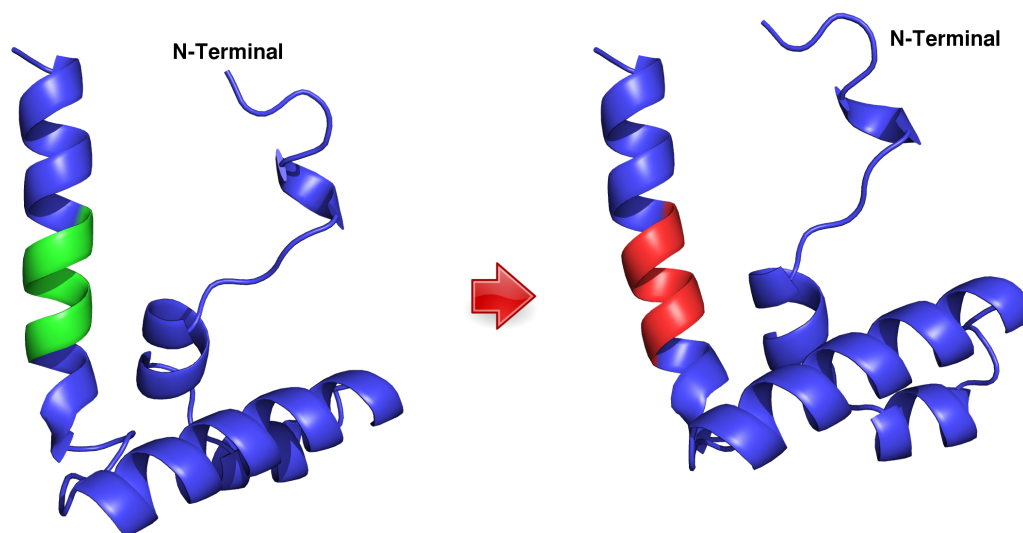


Fig. 4.5 Representation of a move step of SAINT2 Cotranslational mode. A move step occurs at a randomly selected position in the target. The torsion angles corresponding to that position (green) are replaced by the torsion angles of a randomly chosen fragment from the fragment library (red). All other torsion angles remain unchanged.

4.2.8 Mover

The Mover decides when to perform move steps and extrusion steps for the Cotranslational and Reverse modes of SAINT2. There are two Mover approaches currently implemented: Uniform Mover and Linear Mover.

The Uniform Mover puts an equal number of move steps between each extrusion. Using the Uniform Mover, $M/N - 6$ move steps are performed between each extrusion step, where M is the total number of move steps defined at input and N is the length of the target.

The Linear Mover increases the number of moves performed between each extrusion in a linear fashion. When only a small fraction of the target protein has been extruded, a small

number of moves is performed between each extrusion. As the peptide conformation grows, the number of move steps between extrusions increases linearly.

On top of the move steps that occur between extrusions, SAINT2 also performs a series of move steps after the peptide has been fully extruded. We refer to these move steps as post-extrusion move steps. For the analyses described here, 1,000 post-extrusion move steps were performed.

For the *In vitro* mode, only post-extrusion move steps are performed.

For the analyses described in this chapter, we have used a linear mover with 10,000 moves during extrusion and 1,000 moves post-extrusion (Cotranslational and Reverse modes). For analyses using the *In vitro* mode, 11,000 moves were used.

4.2.9 Score

SAINT2 uses a knowledge-based potential built from the following components:

- RAPDF potential (Samudrala and Moult, 1998).
- Lennard-Jones potential (Jones, 1924).
- Solvation potential, as implemented in Tosatto (2005).
- Predicted secondary structure potential.
- Predicted torsion angle potential.
- Predicted inter-residue contacts potential (see Chapter 3 for more details).
- Orientation potential, as implemented in Tosatto (2005).
- Radius of Gyration potential (Lobanov et al., 2008b).

SAINT2's score is a weighted sum of each of its components. The score components are normalized by chain length and to a similar scale in order to ensure that the scoring

weights reflect the contribution of each individual potential. This assists in comparing and understanding the contribution of each individual component to the final score.

The data set used to compute and train these potentials (PISCES Training Set) was derived from PISCES (Wang and Dunbrack, 2003), using the following criteria:

- Sequence Identity Cutoff = 30%.
- Resolution Cutoff = 2.0Å.
- R-Factor Cutoff = 0.2.

Chains containing missing or invalid data were excluded from this set.

4.2.10 Residue-specific All-atom conditional Probability Discriminatory Function

The Residue-specific All-atom conditional Probability Discriminatory Function (RAPDF) is a knowledge-based potential that is commonly used to discriminate between native and non-native protein structures (Samudrala and Moult, 1998; Tosatto, 2005). It is calculated using the observed frequencies of contacts between atom types. All residue-specific non-hydrogen atoms are treated as a different atom type, i.e. the C- α of a leucine is a different atom type than the C- α of an isoleucine. The RAPDF potential in SAINT2 has 99 atom types: one for each of the five backbone atoms (N, C- α , C, O and C- β) for the 20 canonical amino acids except for Glycine, which has no C- β .

We have computed the contact frequencies according to 18 distance bins (0-3 Å, 3-4 Å, 4-5 Å, ... , 19-20 Å) for all atom pairs in our PISCES Training set (Section 4.2.9). Contacts between atoms of the same residue were excluded from these calculations. These frequencies

were then used to compile a table of scores. Given a distance bin d and two atom types a and b , the RAPDF score is given by:

$$s(d_{ab}) = -\log\left(\frac{N(d_{ab})/\sum_d N(d_{ab})}{\sum_{ab} N_{ab}(d_{ab})/\sum_d \sum_{ab} N_{ab}(d_{ab})}\right) \quad (4.1)$$

Where $N(d_{ab})$ is the number of observed contacts between atom types a and b in a specific distance bin d , $\sum_d N(d_{ab})$ is the number of observed contacts between atom types a and b over all distance bins, $\sum_{ab} N_{ab}(d_{ab})$ is the number of observed contacts between any atom types in a specific distance bin d , and $\sum_d \sum_{ab} N_{ab}(d_{ab})$ is the number of observed contacts between any atom types over all distance bins.

This RAPDF score table can be used to assign a RAPDF score to a given pair of atoms. The RAPDF score of a peptide conformation is given by the sum of the individual RAPDF scores of every pair of atoms in that conformation. Atom pairs within the same residues are excluded from these calculations.

4.2.11 Solvation

The Solvation potential is a knowledge-based score that considers the propensity of a given residue to be exposed to solvent (water). The Solvation potential in SAINT2 is based on the one described in Tosatto (2005). We estimate solvent accessibility by counting the number of other C- β s (surrounding C- β s) inside a sphere of radius 10Å centred on a given residue's C- β (for glycine, C- α is used instead).

We have computed the frequencies of surrounding C- β s according to 35 exposure bins (0-3, 4, 5, ..., 36, and 37 or more surrounding C- β atoms) for the 20 canonical amino acids as observed in our PISCES Training set (Section 4.2.9). These frequencies were then used to

compile a table of scores. Given an amino acid *res* and an exposure bin *expo*, the Solvation score can be computed according to the formula:

$$s(res, expo) = -\log\left(\frac{Sur(res, expo) / \sum_{i=1}^{20} Sur(A_i, expo)}{\sum_{j=1}^{35} Sur(res, E_j) / \sum_{j=1}^{35} \sum_{i=1}^{20} Sur(A_i, E_j)}\right) \quad (4.2)$$

Where $Sur(res, expo)$ is the observed number of amino acids of type *a* in the exposure bin *expo*, $\sum_{i=1}^{20} Sur(A_i, expo)$ is the observed number of residues of any type in the exposure bin *expo*, $\sum_{j=1}^{35} Sur(res, E_j)$ is the total number of residues of type *res*, and $\sum_{j=1}^{35} \sum_{i=1}^{20} Sur(A_i, E_j)$ is the number of residues of any type over all exposure bins.

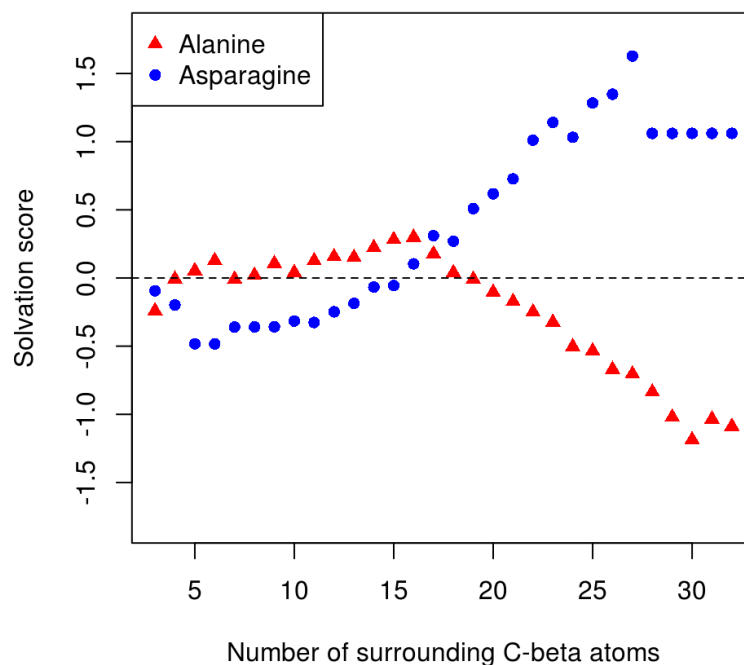


Fig. 4.6 Solvation scores (y-axis) for Alanine (red) and Asparagine (blue) given for each of the exposition bins defined in SAINT2 (x-axis). The number of surrounding C-βs is inversely proportional to a given residue's solvent exposure.

The Solvation scores for Alanine (apolar) and Asparagine (polar) are shown in Figure 4.6. Apolar residues such as Alanine present negative Solvation scores (favourable) when they are buried (high number of surrounding C- β s). On the other hand, polar residues such as Asparagine present negative Solvation scores when they are on the surface (low number of surrounding C- β s).

The Solvation potential in SAINT2 uses the pre-computed tables for all residues/bins. For missing or sparse values (frequency less than 7) we used the scores of their nearest bins. The Solvation potential is the sum of the Solvation score for all residues in a given conformation.

4.2.12 Lennard-Jones Potential

The Lennard-Jones potential is a mathematical model that approximately describes inter-atomic interactions (Jones, 1924). The Lennard-Jones potential is commonly expressed in terms of the distance r between two atoms, as given by the formula:

$$V_{LJ}(r) = 4\epsilon \times \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (4.3)$$

In this equation, V_{LJ} is the inter-atomic potential between the two atoms, ϵ is the depth of the potential well surrounding the function's global minimum, and σ is the distance at which the potential between the two atoms is zero (see Figure 4.7). The first term r^{-12} is known as the repulsion term as it represents the repulsion when electron orbitals are overlapping. The term $-r^{-6}$ is the attractive term and is an approximation of Van der Waals interactions.

SAINT2's Lennard-Jones score between two atoms of types a and b at a distance d is given by the formula:

$$LJ_{ab} = \frac{C_{ab}^{12}}{d^{12}} - \frac{C_{ab}^6}{d^6} \quad (4.4)$$

Where $C_{ab}^{12} = \epsilon_{ab} * \sigma_{ab}^{12}$ and $C_{ab}^6 = 2 \times \epsilon_{ab} \times \sigma_{ab}^6$. The ϵ and σ values were taken from the OPLS-AA force field (Jorgensen et al., 1996).

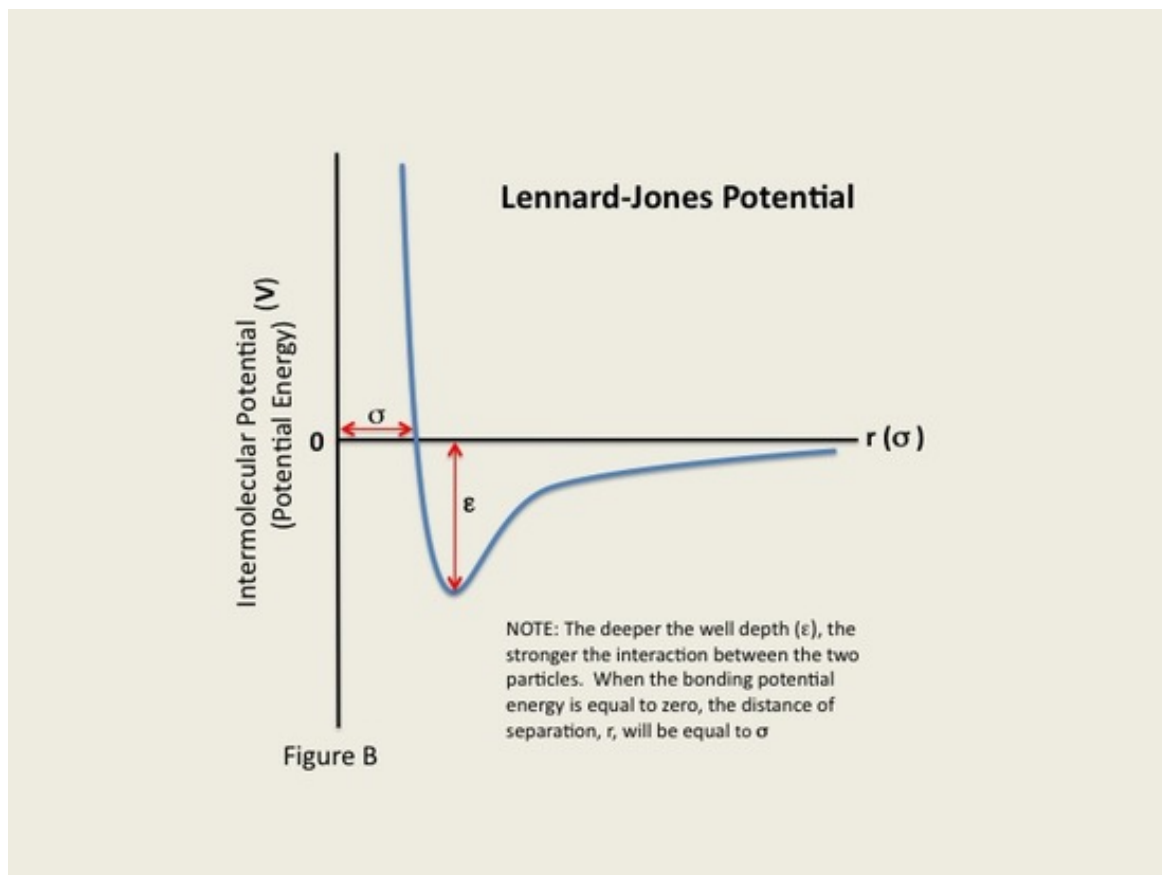


Fig. 4.7 Representation of the Lennard-Jones Potential.¹

The Lennard-Jones score in SAINT2 is assigned a value of zero when the distance between two atoms is greater than $2.5 \times \sigma$. The total Lennard-Jones score for a given peptide conformation is the sum of the Lennard-Jones score of all of its atom pairs, except the pairs in the same residue or the residues immediately before/after it in the protein sequence (residues $i - 1, i + 1$).

4.2.13 Orientation Potential

The Orientation Potential describes the frequencies of contacting residues for different distance bins and also considering the orientation of their side chains. The distance between

a pair of residues is calculated using their C- α atoms. In order to represent side chain orientation, three angles are used: $C\alpha_2-\hat{C}\alpha_1-C\beta_1$, $C\alpha_1-\hat{C}\alpha_2-C\beta_2$, and $C\beta_2-\hat{C}\alpha_1-C\beta_1$. We provide a schematic representation of these angles in Figure 4.8.

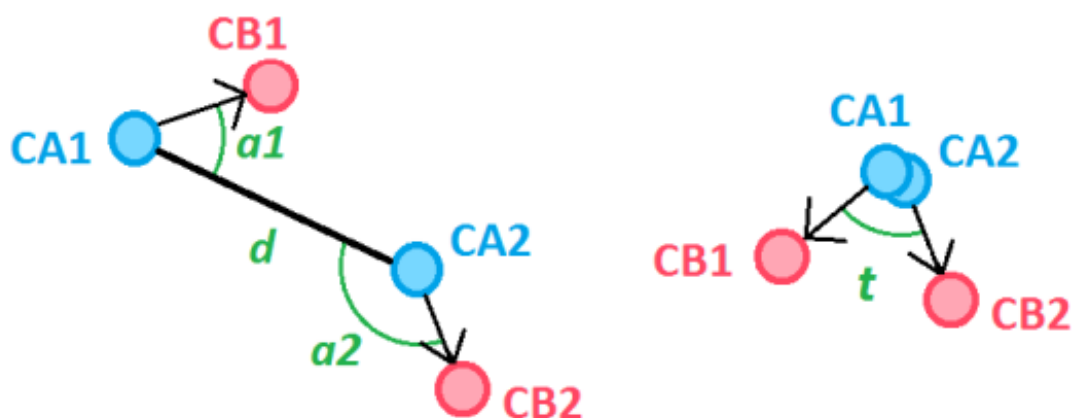


Fig. 4.8 Schematic representation of the angles used to compute the Orientation score. CA1 and CA2 represent the C- α s of two distinct residues. CB1 and CB2 represent the C- β s of these residues (for Glycine, we estimate the position of its C- β). d represents the distance between CA1 and CA2. The angles $a1$ and $a2$ represent the angles $C\alpha_2-\hat{C}\alpha_1-C\beta_1$ and $C\alpha_1-\hat{C}\alpha_2-C\beta_2$, respectively. The angle $C\beta_2-\hat{C}\alpha_1-C\beta_1$ is represented by t .

Table 4.1 The orientation bins used to classify residue pairs for the Orientation Potential

	$a1 < 45$	$45 \leq a1 < 90$		$a1 \geq 90$
		$t < 90$	$t \geq 90$	
$a2 < 45$	Bin 1	Bin 2		Bin 3
$45 \leq a2 < 90$	Bin 4	Bin 5	Bin 6	Bin 7
$a2 \geq 90$	Bin 8	Bin 9		Bin 10

We have used ten orientation bins to classify residue pairs according to the orientation of their side chains (Table 4.1) and the 18 distance bins defined in Section 4.2.10. Therefore, we have a total of $10 \times 18 = 180$ different distance-orientation bins. The frequency of all residue

¹Image by Anonymous Author under Creative Commons (CC BY-NC-SA 3.0 - <http://creativecommons.org/licenses/by-nc-sa/3.0>), via UCDavis ChemWiki.

pairs for each bin was computed for the PISCES Training set and used to compute a table of scores. The score of a pair of residues a and b in a distance-orientation bin do is given by:

$$s(do_{ab}) = -\log\left(\frac{N(do_{ab}) / \sum_{ab} N(do_{ab})}{\sum_{do} N(do_{ab}) / \sum_{do ab} \sum_{ab} N_{ab}(d_{ab})}\right) \quad (4.5)$$

Where:

$N(do_{ab})$ is the observed number of residue pairs (a, b) in the distance-orientation bin do ,
 $\sum_{i,j=1}^{20} N(do_{ij})$ is the observed number of all residue pairs in the distance-orientation bin do ,
 $\sum_{do} N(do_{ab})$ is the observed number of residue pairs (a, b) across all distance-orientation bins,
 and $\sum_{do ab} \sum_{ab} N_{ab}(d_{ab})$ is the observed number of all residue pairs across all distance-orientation bins. Missing or sparse values (frequency less than 7) are set to zero.

The final Orientation potential of a given peptide conformation is the sum of the Orientation scores of every pair of residues in that conformation. Pairs of residues immediately adjacent in the protein sequence (residues $i, i+1$) are excluded from this calculation.

4.2.14 Predicted Secondary Structure Potential

SAINT2's predicted secondary structure potential is determined by counting the number of residues that disagree with their predicted secondary structure. To calculate the potential, matches are given a predicted secondary structure score of 0 and mismatches are given a score of 10. Only residues that are predicted to be α -helical or β -strand are considered in this calculation. Secondary structure predictions were generated using PSIPRED (Jones, 1999) and default parameters.

4.2.15 Predicted Torsion Angle Potential

The predicted torsion angle potential is based the score described in Section 2.2.4. For this potential, the torsion angles (ϕ, ψ) of all residues in a given peptide/protein conformation are computed and compared to the predicted torsion angles output by SPINE-X (Faraggi et al., 2009, 2012).

The total predicted torsion angle score of a given peptide conformation of length L is given by:

$$Score = \sum_{i=1}^L \sqrt{(\phi_{predicted} - \phi_{conformation})^2} + \sqrt{(\psi_{predicted} - \psi_{conformation})^2} \quad (4.6)$$

4.2.16 Radius of Gyration Potential

The radius of gyration (R_g) is a measure of compactness of a protein conformation and is given by the formula:

$$R_g^2 \approx \sum_{i=1}^{N_{atoms}} (a_i - C)^2 / N_{atoms} \quad (4.7)$$

Where N_{atoms} is the total number of atoms, a_i is an individual atom and C is the geometric centre of the protein.

We assessed the distribution of radii of gyration for all protein in the PISCES Training set (Figure 4.9). We then performed a linear regression for the radius of gyration using protein length (Figure 4.10). This regression was based on the scaling law for the radius of gyration under physiological conditions (Hong and Lei, 2009).

We used the linear regression shown in Figure 4.10 to developed a radius of gyration potential for SAINT2. For this potential, we estimate the radius of gyration (predicted) of a peptide conformation using our linear model. We then compute the observed radius of

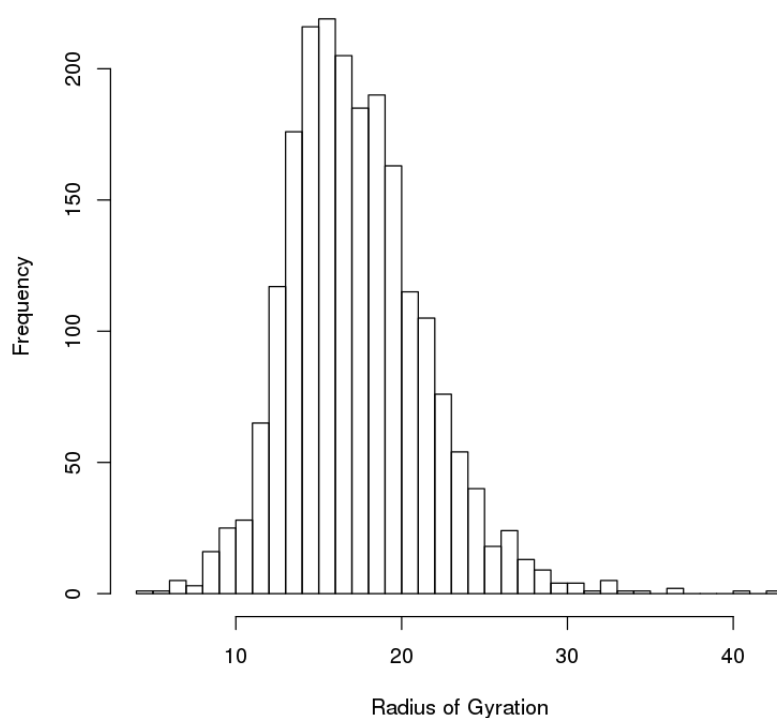


Fig. 4.9 Distribution of the radii of gyration for all proteins in the PISCES Training set.

gyration for the peptide as given by equation 4.7. The final score is given by the absolute value of the difference between the predicted and observed radii of gyration of a given conformation.

4.2.17 Predicted Contact Potential

We have based our predicted residue-residue contact potential on the potentials described in Kamisetty et al. (2013); Kosciolk and Jones (2014). SAINT2 penalizes residues that were predicted to be, but are not in contact. We define two residues to be in contact whenever their C- β s (C- α s for glycine) are within 8.0 Å from each other. As previously defined in Section 3.2.11, the score of a given pair of residues i, j predicted to be in contact is given by:

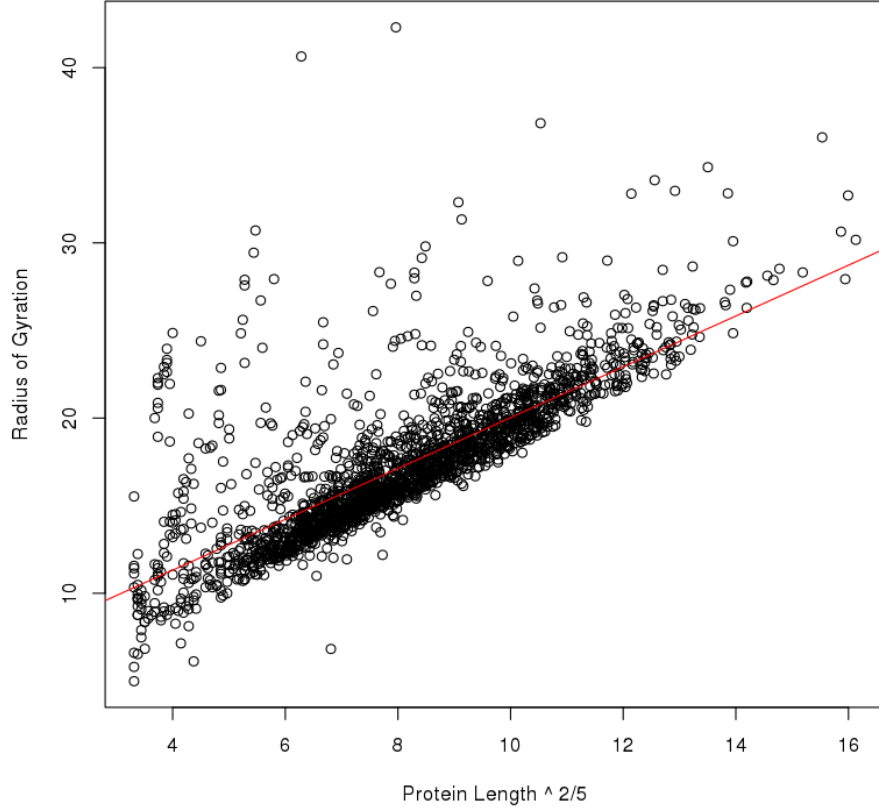


Fig. 4.10 Regression based on the scaling law of the radius of gyration for all proteins in the PISCES Training set. A linear regression was performed for the Radius of Gyration using $L^{2/5}$, where L is the protein length.

$$S_{ij}^{contact} = \begin{cases} 0, & \text{if } \|\mathbf{C}_\beta(i) - \mathbf{C}_\beta(j)\| = \sqrt{\sum_{\kappa=x,y,z} (C_\beta^\kappa(i) - C_\beta^\kappa(j))^2} < 8.0 \text{ \AA} \\ \|\mathbf{C}_\beta(i) - \mathbf{C}_\beta(j)\| - 8.0 \text{ \AA}, & \text{otherwise.} \end{cases}$$

Where $\mathbf{C}_\beta(i)$ and $\mathbf{C}_\beta(j)$ represent the coordinates of the C- β s (C- α s in the case of glycine) of residues i and j . The Predicted Contact Potential is given by the sum of the scores of every pair of residues predicted to be in contact.

4.2.18 Normalization of SAINT2 Score Components and Initial Scoring Weights

As RAPDF, Lennard-Jones, Solvation, and Orientation potentials have arbitrary scales, they were normalised to the same scale by the following procedure: 505 proteins were selected at random from the 5425 proteins in the PISCES Training set described in Section 4.2.9. This subset does not contain any homologues to the proteins in our Training and PDB-Representative Validation data set. For each of these protein, eight decoys were generated by perturbing the torsion angles of the native state to comprise a Decoy Training set. Steric clashes were not accepted when causing these perturbations. Perturbations were done in a way to ensure that the GDT-TS (Zemla, 2003b) to native structure of decoys in the Decoy Training set was evenly spaced between 40 and 100. The four potentials were calculated for each of the 4040 decoys and normalised to have a mean value of 100 (which is performed by adding a normalising constant λ_1) and to have the same minimum value as the RAPDF (which was performed by scaling using a constant λ_2).

The initial scoring weights for the RAPDF, Lennard-Jones, Solvation, and Orientation potentials were calculated using hill-climbing where we maximised the correlation between the weighted sum of the four scores and the GDT-TS of each decoy in the Decoy Training set. Weights were not permitted to go below zero. The native structure's score for each potential was subtracted from the score of every decoy to enable comparison of scores between different proteins. During the initial scoring weight estimation, we observed two distinct behaviour for proteins shorter than 150 residues and proteins longer than 150 residues. We decided to use two sets of weights according to protein length based on this observation.

The predicted secondary structure potential, predicted torsion angle potential, radius of gyration potential, and predicted contact potential were implemented at a later stage than the other potentials. For that reason, their normalisation and initial scoring weight estimation was performed differently. Ten different sets of weights per potential were tested. For this test, we

used SAINT2 Cotranslational mode to generate 1,000 decoys for each the 41 proteins in our Training Data set (see Table 2.1). We chose the weights that led to the best modelling results. Given the number of potentials that have been implemented in SAINT2, it has become necessary to reproduce the original scoring weight estimation using a procedure similar to the one described for the RAPDF, Lennard-Jones, Solvation, and Orientation potentials.

The initial scoring weights used in SAINT2 are described in Table 4.2.

Table 4.2 Initial Scoring Weights for the Score Components in SAINT2

Component	Length $\leq$ 150 residues	Length $>$ 150 residues
RAPDF	0.156	0.303
Solvation	0.262	0.282
Lennard-Jones	0.505	0.304
Orientation	0.077	0.111
Pred. Secondary Structure	0.2	0.2
Pred. Torsion Angle	0.1	0.1
Radius of Gyration	0.1	0.1
Pred. Residue-Residue Contacts	1.0	1.0

4.2.19 Optimizing Scoring Weights

In order to optimize the scoring weights, we performed several tests in which we altered one of the scoring weights and assessed the impact of that alteration on modelling accuracy (Figure 4.11). For each test, we generated 3,000 decoys for each of the 43 proteins in our Training Data Set. Then, we compared the 43 TM-Score Best values obtained using the altered scoring weights against the TM-Score Best values obtained using the scoring weights described in Section 4.2.18. We define $\Delta_{TM-Score}$ as the difference between the TM-Score Best with the altered weights and the TM-Score Best with the initial weights.

Our results show that SAINT2 is robust to small perturbations to its scoring weights. Certain adjustments (i.e. RAPDF $\times 4$ and Solvation $\times 4$) show improvements over the initial scoring weights for some proteins, but this improvements are offset by other proteins for which results are significantly worse. Considering no scoring weight adjustment revealed

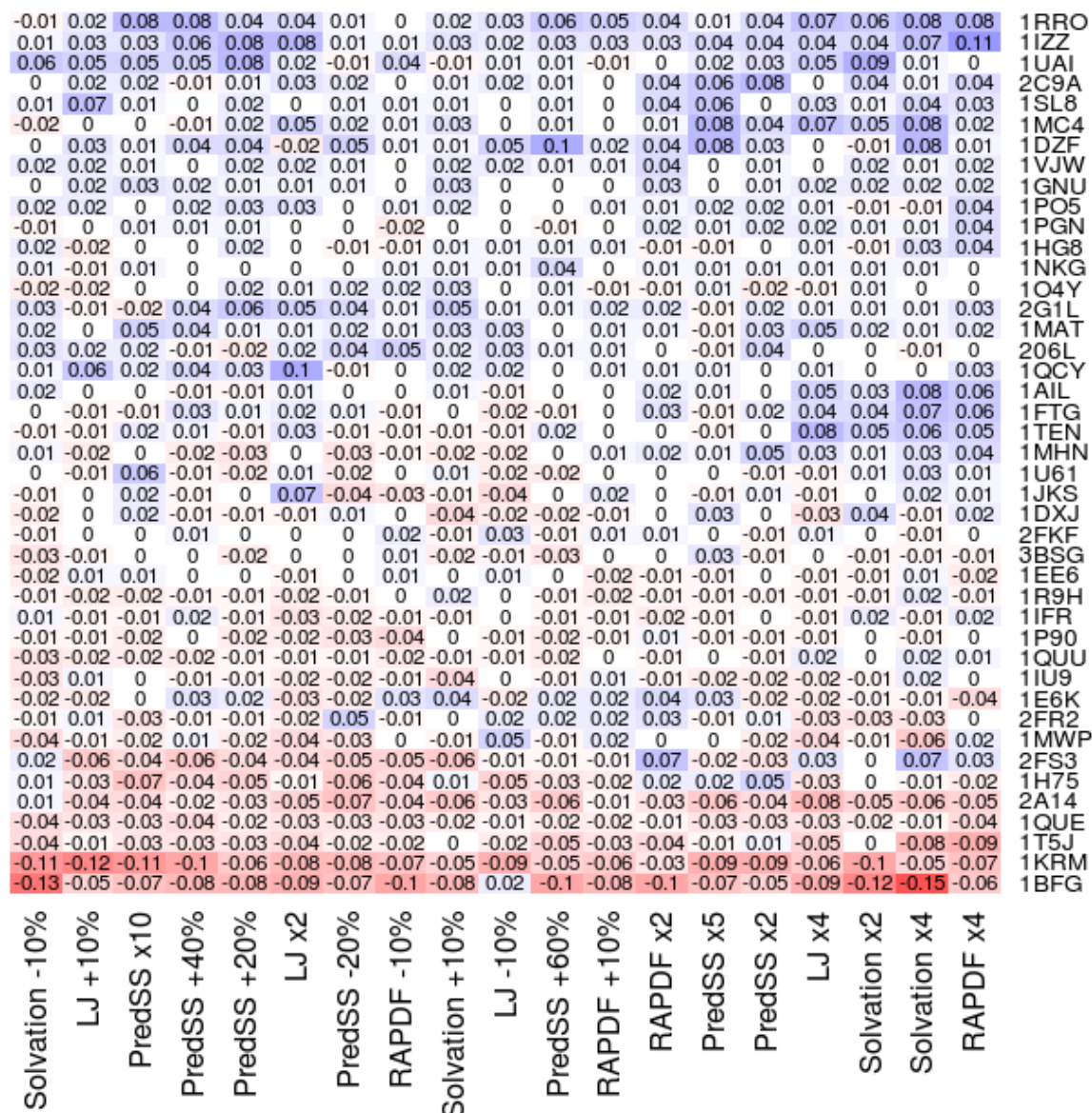


Fig. 4.11 Heatmap of the $\Delta_{TM-Score}$ for the scoring weight tests performed on the 43 proteins in our Training Data Set. For each test (columns), we generated 3,000 decoys using the adjusted scoring weight (x-axis labels) and compared the results with the ones obtained using the initial scoring weights described in Section 4.2.18. We define $\Delta_{TM-Score}$ as the difference between the TM-Score Best with the altered weights and the TM-Score Best with the initial weights. The colouring indicates whether the change in scoring weight led to an improvement (blue) or a decrease (red) in TM-Score Best.

a clear improvement for the majority of proteins in our Training Data Set, we decided to maintain the initial scoring weights for subsequent analyses.

4.2.20 Strategy

The Strategy is the criterion used to decide whether peptide conformation originating from a move step should be accepted or discarded. There are different strategies implemented within SAINT2, but for the purpose of the analyses described in this Thesis, only the Monte Carlo strategy has been used.

The Monte Carlo criterion consists of accepting new conformations according to a probability. This probability depends on the difference between the score of a new conformation and the score of the previous conformation, referred to as Δ . The acceptance probability of the Monte-Carlo criterion, based on a synthetic temperature T defined at input, is given by:

$$P = e^{-\Delta/T} \quad (4.8)$$

4.2.21 Selection and Refinement

SAINT2 samples the conformational space by generating thousands of decoys for each target. Decoys are ranked according to our knowledge-based potential (see Section 4.2.9) and the top-5 scoring decoys are output by our sequence-to-structure pipeline.

Currently, we do not perform any refinement step on the decoys output by SAINT2.

4.2.22 Performance

De novo protein structure prediction is known to be computationally intensive. For instance, state-of-the-art software such as ROSETTA can take thousands to hundreds of thousands of core hours to generate predictions for a single target protein. Therefore, efficient use of resources is crucial for *de novo* structure prediction. Given the investigative character of SAINT2, the software's architecture was designed to be modular. This modularity has a performance cost. For example, scoring potentials were originally implemented

independently despite performing redundant operations (i.e. calculating distances between atoms).

The most time consuming step in a run of SAINT2 is the scoring of decoys. The time complexity of the scoring function is determined by its potentials. Potentials such as the RAPDF and Lennard-Jones perform an all atom pairwise comparison, which has a time complexity in the order of $O(L^2)$, where L is the number of residues. The Solvation and Orientation potentials are also quadratic on the protein length, but are faster than the RAPDF and Lennard-Jones as only one atom per residue is used in their calculations. The remaining potentials run in linear time ($O(L)$). Performing a fragment replacement and generating a new conformation also runs in linear time ($O(L)$). For a run of SAINT2, M conformations are generated and scored. Therefore, the time complexity of SAINT2 is in the order of $O(M \times L^2)$, where M is the total number of moves and L is the length of the protein. Given that we have used a constant number of moves for the analyses described in this work, the time complexity of SAINT2 is quadratic.

Our initial assessment of the speed profile of a run of SAINT2 revealed that the run time for all operations of SAINT2 were negligible compared to its scoring function (Figure 4.12). We proceeded to profile the run time of each score component of SAINT2 using the Co-translational mode to perform predictions for a specific target (1H75). Run times of each score component were averaged over ten runs and results are shown in Figure 4.12. The speed profile revealed that the RAPDF and the Lennard-Jones potentials are, as expected, the limiting factors in terms of SAINT2's run time.

To optimize SAINT2's performance, we decided to merge the RAPDF and the Lennard-Jones potentials into a Core potential. We eliminated unnecessary conditional branching and implemented an all-atom distance table to be accessed by all potentials. The resulting run time profile is shown in Figure 4.13.

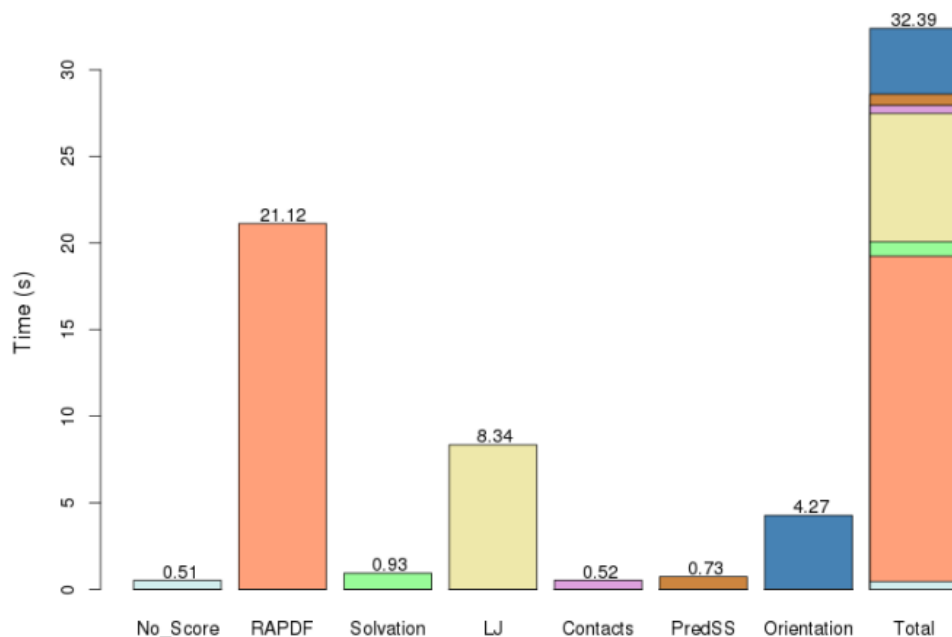


Fig. 4.12 Run time profile of each score component of SAINT2 before performance optimization. Run times are averaged over ten runs of SAINT2 Cotranslational mode for protein 1H75.

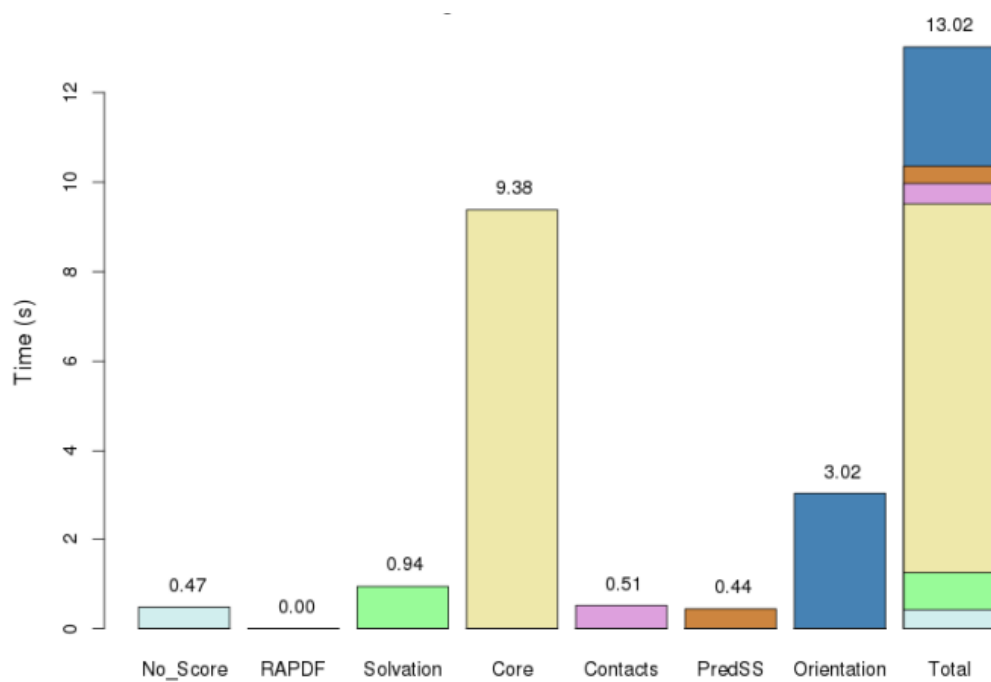


Fig. 4.13 Run time profile of each score component of SAINT2 after speed optimization. Run times are averaged over ten runs of SAINT2 Cotranslational mode for protein 1H75.

After optimization, a run of Cotranslational mode of SAINT2 for protein 1H75 takes about 13 seconds on a AMD OpteronTM Processor 6272 (2100 MHz). We were able to reduce SAINT2's run time by $\approx 66\%$. We compared the run time of an individual run of SAINT2 against an individual run of ROSETTA (build 3.5), one of the leading methodologies for *de novo* protein structure prediction. We performed this comparison for the 43 proteins in our Training data set (Figure 4.14). We aimed for a fair comparison and used the best possible set of parameters for both programmes. It is important to highlight that ROSETTA performs full-atom predictions and that in its ideal settings, 360,000 moves are used. This number is significantly higher than the 11,000 moves performed by SAINT2. For this reason, when using the ideal settings of both predictors, we would expect ROSETTA to be significantly slower than SAINT2. Indeed, our results show that SAINT2 can generate a decoy up to 20 to 80 times faster than ROSETTA. To generate 1,000 decoys for each protein in our Training Data set, it takes ≈ 1000 CPU hours using SAINT2 and ≈ 32000 CPU hours using ROSETTA. However, given that the ideal number of moves used in ROSETTA is more than 30 times larger than the ideal number of moves used in SAINT2 we would also expect ROSETTA to perform significantly better than SAINT2. Yet, when given the same amount of computational time, ROSETTA's modelling results were consistently worse than the ones obtained by SAINT2 Cotranslational (refer to section 4.3 for more details).

4.2.23 Critical Assessment of protein Structure Prediction

We submitted predictions for 13 targets in the 11th edition of the Critical Assessment of protein Structure Prediction (CASP11). The list of targets for which predictions were submitted is given in Table 4.3. For each of the targets, we generated 1,000 decoys using our sequence-to-structure pipeline and the Cotranslational Mode of SAINT2. We generated this number of decoys due to limited computational resources and because, at the time, we had

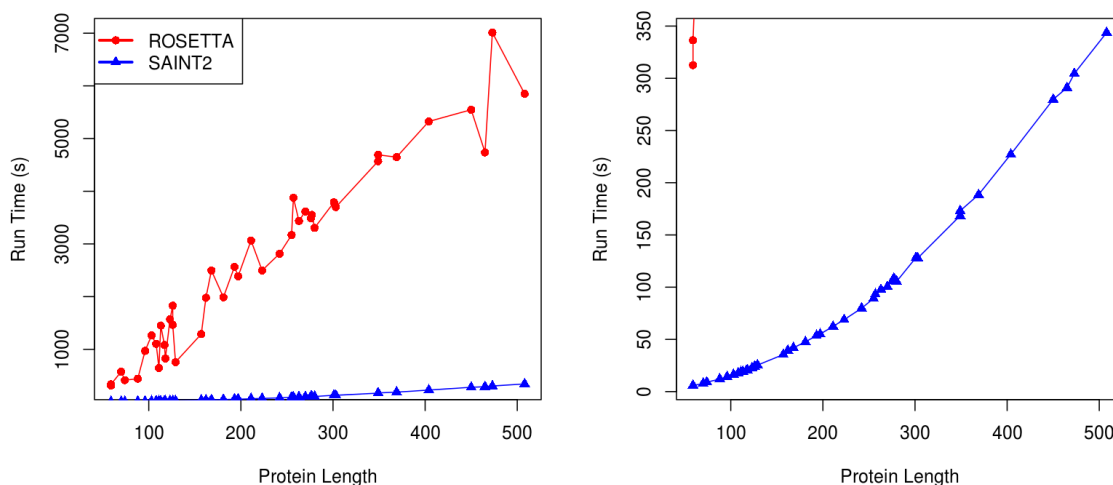


Fig. 4.14 Run times for an individual run of SAINT2 (blue) and ROSETTA (red) for the 43 proteins in our Training Data Set.

not yet estimated the number of decoys required by SAINT2 to produce a correct answer (see Section 5.3.1).

The score potentials Predicted Torsion Angle, Predicted Secondary Structure, Predicted Residue-Residue Contacts, and Radius of Gyration had not been fully implemented and tested within SAINT2 when CASP11 took place and, therefore, were not used for the predictions. For that reason, contact predictions were also excluded from the sequence-to-structure prediction pipeline for the CASP11 targets. The scoring weights for the other potentials are the same as described in Section 4.2.18

4.3 Results

Our aim when designing SAINT2 was to develop a tool that would allow the incorporation of more biologically realistic features into protein structure prediction. To this end, SAINT2 was designed as a highly modular tool. New features can easily be incorporated and tested and existing features can be disabled by the user at input. Nonetheless, to be able to test our

hypotheses with adequate scientific rigour, it was important to ensure that SAINT2 produced results that were comparable to state-of-the-art protein structure prediction software.

We have chosen to compare our software SAINT2 to ROSETTA (Bonneau et al., 2002, 2001; Bradley et al., 2005; Raman et al., 2009), one of the methods that achieved the best results for the free modelling category of the last iteration of CASP.

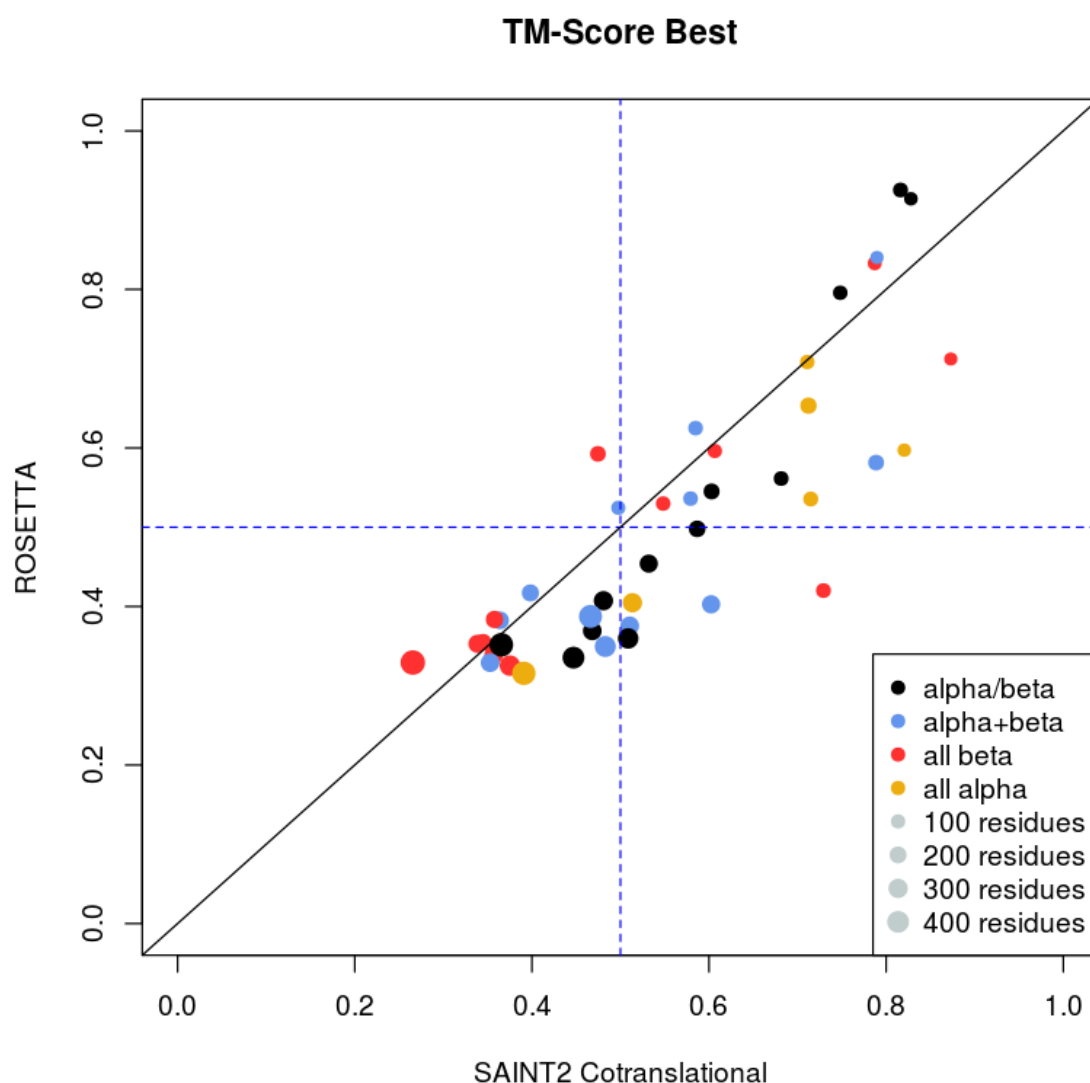


Fig. 4.15 Comparison between the TM-Score Best of SAINT2 against ROSETTA for the 43 proteins in our Training Data Set. A TM-Score greater than 0.5 (blue line) is considered to be a correct answer. Proteins are coloured according to their SCOP class and sorted according to length.

We compared the modelling results of SAINT2 Cotranslational against ROSETTA for a Training Data Set of 43 structurally diverse proteins (Figure 4.15). As there is no published optimal number of decoys necessary to produce a correct/good ROSETTA model, we chose to use the CPU time necessary to generate 10,000 decoys with SAINT2 Cotranslational, its optimal setting. SAINT2 Cotranslational is significantly faster than ROSETTA (as seen in section 4.2.22) so, in this benchmark, ROSETTA produced fewer than 10,000 decoys for each case. ROSETTA, in its ideal settings, performs a much higher number of moves than SAINT2. Therefore, the fairest way to compare both programmes is to use the same amount of CPU time for both of them. It is possible that if ROSETTA was run for longer, its results could approach or even overtake those of SAINT2, but the results of SAINT2 would also improve with extra computational time. Another important consideration comes from the fact that ROSETTA is generating full-atom models whereas SAINT2 is only predicting backbone atom coordinates. ROSETTA and its scoring potentials were developed and trained to model full-atom structures. Disabling these capabilities would increase the speed performance of ROSETTA at a cost of final model quality. Therefore, we maintain our argument that the fairest comparison between the two programmes is to use the same amount of CPU time for each software and their ideal settings.

Considering the TM-Score of the best decoy (TM-Score Best), SAINT2 produced a correct answer for 24 out of the 43 cases and ROSETTA produced a correct answer for 19 of the 43 cases. There were seven cases for which SAINT2 generated a correct answer and ROSETTA did not, and only two cases where a correct answer was produced by ROSETTA and not by SAINT2. For proteins up to 200 residues, both SAINT2 and ROSETTA produced a correct answer for 19 out of 22 cases. Yet, ROSETTA did not produce a correct answer for proteins longer than 200 residues. If we consider the SCOP class of the proteins, SAINT2 consistently generated better predictions than ROSETTA for α/β proteins longer than 100 residues.

4.3.1 CASP

As described in Section 4.2.23, we used SAINT2 to generate 1,000 decoys for 13 of the targets in CASP11. For these predictions, we used a preliminary version of our pipeline that did not include predicted contact information and did not use several potentials (Predicted Secondary Structure, Predicted Torsion Angles and Radius of Gyration potentials). Out of the 13 targets, 4 targets were cancelled as their structure could not be resolved experimentally during the duration of CASP11. The results for the remaining 9 targets are shown in Table 4.3.

Table 4.3 CASP11 Results

Target	PDB ID	Length	TM-Score Best	TM-Score Top 5	Best CASP Model
T0855*	2MQD	119	0.41	0.41	0.60
T0853	2MQB	152	0.50	0.35	0.65
T0850	-	190	-	-	-
T0848	4R4G	354	0.33	0.24	0.39
T0846	-	146	-	-	-
T0844	-	145	-	-	-
T0842	-	146	-	-	-
T0838	-	154	0.32	0.33	0.70
T0837*	-	128	0.44	0.44	0.79
T0836*	-	204	0.51	0.48	0.60
T0835	-	424	0.30	0.31	0.76
T0834*	4R7Q	219	0.47	0.40	0.43
T0832	4RD8	257	0.31	0.36	0.44

Out of the nine valid CASP11 targets for which predictions were submitted, SAINT2 produced correct answers for two (TM-Score Best > 0.5). However, no correct answer was submitted (TM-Score Top 5 > 0.5). The native structure and the model are shown for the CASP target T0836 in Figure 4.16.

4.4 Discussion

In this Chapter, we have described the implementation of SAINT2, a cotranslational *de novo* protein structure prediction method. In SAINT2, we have implemented three different run

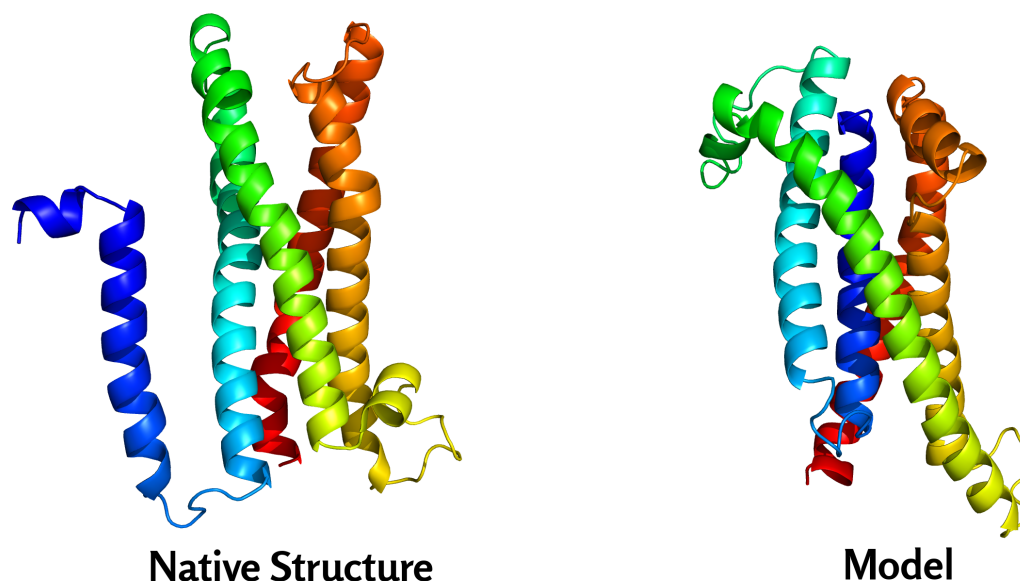


Fig. 4.16 Native structure and the best model generated by SAINT2 for CASP11 target T0836.

modes to test our hypothesis that cotranslational protein folding can be used to improve protein structure prediction. In SAINT2 Cotranslational mode, we start predictions with a short peptide that gets extended throughout the simulation. To our knowledge, no other current method performs prediction in this cotranslational fashion.

We have defined two different fragment replacement types, moves and extrusions, to assist in model generation. This is a unique feature of SAINT2. The order in which these moves and extrusions are performed is determined by a Mover. We have implemented two different Mover profiles, but the extension to other Mover profiles is trivial and could be used to reflect translation kinetics (see Chapter 5 for further details).

We performed a comparison between SAINT2 and ROSETTA and found that SAINT2 is capable of producing an individual decoy up to 80 times faster than ROSETTA. Our modelling results also show that, in general, SAINT2 produces better models than ROSETTA. When we consider the cases where a correct answer was produced, SAINT2 produced correct answers for 24 of these cases and ROSETTA produced correct answers for 19. This suggests

that our sequential approach is capable of sampling conformations that are explored by a non-sequential approach such as ROSETTA.

We also found that SAINT2 generates consistently better results than ROSETTA for α/β proteins longer than 100 residues. Proteins belonging to α/β SCOP class have been suggested to fold in a cotranslational fashion. This is another indication that cotranslational aspects may be useful for *de novo* protein structure prediction.

One of the main limitations of our sequence-to-structure pipeline is the lack of a refinement step. Results in the refinement category of the past two iterations of CASP have shown that several refinement approaches can be used to consistently improve modelling results. The incorporation of a refinement step into our pipeline is necessary if we aim to participate in the next iterations of CASP. However, the main focus of this work has been to use additional biological information to improve model generation. If we can generate higher quality models, we can expect that refining these models would lead them even closer to the native structure.

As expected, our results in CASP11 are worse than the results obtained for our Training Data Set. CASP targets are known to be difficult to model. Several of these targets were longer than 300 residues and SAINT2, like all *de novo* protein structure prediction software, tends to perform better for shorter proteins. The number of decoys generated for CASP using SAINT2 was relatively low and several potentials were not used in the generation of these decoys. Furthermore, we do not employ any clustering routine to perform model quality assessment. Clustering routines have been shown to be better for decoy selection than the single decoy approach used in our pipeline.

Our results in CASP are not comparable to the best models submitted by other groups, but this is not surprising given that we entered the competition with a preliminary version of our pipeline. Overall, the fact that we were capable of generating some correct answers for hard targets using this preliminary version of SAINT2 was a positive result. More importantly,

there were several targets for which no correct answer was produced amongst the more than 200 CASP11 participants. This suggests that other aspects, such as the context under which proteins fold, should be explored in protein structure prediction if we aim to predict protein structures consistently and accurately.

SAINT2 has been tested and shown to generate correct answers for some cases. With that in mind, we have confidence that SAINT2 can be used to test interesting hypotheses and to provide us with insights on how we can use biology as a source of inspiration to improve *de novo* protein structure prediction.

Chapter 5

Biologically Inspired *de novo* Protein Structure Prediction

5.1 Introduction

Average proteins are composed of hundreds of amino acids and each of these residues can assume many conformations (Avbelj and Baldwin, 2003). The number of possible conformations that a protein can assume is therefore very large. Yet proteins are capable of folding accurately and reaching their native state in microseconds (Debès et al., 2013; Hartl and Hayer-Hartl, 2009; Ivankov et al., 2003; Zou et al., 2014).

It is widely accepted that proteins do not sample every possible conformation when folding. Instead, the process of folding is thought to be directed by local interactions, coordinating the formation of intermediates that facilitate folding (Dill and Chan, 1997; Hartl and Hayer-Hartl, 2009; Zwanzig et al., 1992). In this chapter, we explore whether the same conformational search strategies employed *in vivo* could be used to improve protein structure prediction.

As discussed in Chapter 4, *in silico* models generated *de novo* do not achieve experimental quality/resolution for a large number of cases (Moult et al., 2014, 2011a). One of the problems

encountered when predicting protein structures *de novo* is the size of the conformational search space (Dill and Chan, 1997; Onuchic et al., 1997; Zwanzig et al., 1992). As described above, during folding *in vivo*, protein also have to sample from a large combinatorial space, but are capable of folding efficiently and accurately. In order to better understand how proteins restrict the search space and attain their native-state conformation, it may be important to consider the biological context under which proteins fold. There is a growing body of evidence that suggests this context is mediated by protein synthesis (e.g. Basharov (2000, 2003); Fedorov and Baldwin (1997); Giglione et al. (2009); Kadokura and Beckwith (2009); Kolb (2001)).

One hypothesis is that proteins fold as they are being translated (cotranslational protein folding). Protein synthesis is known to occur on longer time scales than protein folding. In *E. coli*, for instance, protein synthesis occurs at approximately 0.05s/codon (Pedersen, 1984; Sørensen and Pedersen, 1991) whereas secondary structure formation is thought to occur in microseconds (Debès et al., 2013; Hartl and Hayer-Hartl, 2009; Ivankov et al., 2003; Zou et al., 2014). This suggests folding must occur during translation as there is no evidence of a mechanism that prevents partially extruded peptides from assuming conformations as they wait for the rest of the protein to be translated.

Some fundamental questions can be posed to understand the relationship between protein folding and translation. Is cotranslational protein folding necessary for folding? More importantly, is it necessary for fast and efficient folding? Some globular proteins are capable of refolding *in vitro* after being denatured (e.g. Anfinsen (1972); Anfinsen et al. (1961)), which suggests that cotranslational protein folding is not essential for folding. However, there is growing evidence to support the potential importance of cotranslational protein folding for fast and efficient folding. Several proteins cannot refold after denaturation and refolding often leads to lower yields due to aggregation and misfolding (e.g. Cabrita and Bottomley (2004); Clark (2001); Eiberle and Jungbauer (2010)). There is also evidence that proteins

tend to fold orders of magnitude faster than they refold (e.g. Fedorov and Baldwin (1999); Seckler et al. (1989)).

Another biological feature that has been suggested as a key player in cotranslational protein folding is the way in which different codons that encode for the same amino acid can regulate protein synthesis kinetics (Han et al., 2012; Komar, 2009; O'Brien et al., 2012; O'Brien et al., 2014a,b; Sander et al., 2014; Yang et al., 2014). Translational pauses caused by rare or less available codons are thought to allow partially extruded peptides sufficient time to fold into energetically favourable intermediates. Low availability of a given codon have been shown to slow down and even halt translation (Angov et al., 2008; Clarke et al., 2008; Spencer et al., 2012; Tsai et al., 2008; Zhang et al., 2009). This has an effect on the folding of multi-domain proteins (Angov et al., 2008; Clarke et al., 2008; Komar, 2009; Spencer et al., 2012; Tsai et al., 2008; Zhang et al., 2009). Synonymous codon mutations have also been associated with protein misfolding (Clarke et al., 2008; Spencer et al., 2012) and shown to enhance protein expression (Angov et al., 2008). There is evidence that codon optimality is conserved, which suggests a relationship between synonymous codons and protein structure and function (Pechmann and Frydman, 2013). The degeneracy of the genetic code, therefore, supports the notion that proteins fold as they are being synthesized.

The underlying hypothesis behind cotranslational protein folding is that it promotes the formation of energetically favourable intermediates, which help a protein to sample its conformational space efficiently. These cotranslational protein folding intermediates have been observed experimentally. Both cryo-electron microscopy and nuclear magnetic resonance spectroscopy of proteins reveal that nascent chains attached to the ribosome can resemble the native state (Cabrita et al., 2009; Eichmann et al., 2010; Hsu et al., 2007; Knoops et al., 2012; Lin et al., 2012). In the experiments of Lin et al. (2012), the intermediates were shown to regulate the recruitment of molecules associated with translation and folding, such as the signal recognition particle and its receptor FsY (Lin et al., 2012). Other studies have

assessed whether nascent chains are functionally active. Certain enzymes, such as viral proteases (Nicola et al., 1999; Sánchez et al., 2004), show catalytic activity when they are partially extruded from the ribosome, which indicates that the shape of the intermediates is similar enough to the native-state to confer enzymatic activity to the nascent chain. Similar results were observed when assessing whether ribosome-bound nascent chains of globin were capable of binding to heme (Komar et al., 1997).

There are many aspects of cotranslational protein folding that could promote the formation of energetically favourable intermediates, i.e. the interaction of the extruding peptide with the ribosome tunnel (Lim and Spirin, 1986), crowding effects of the cellular environment and the action of chaperone proteins that mediate protein folding (Duttler et al., 2013; Md et al., 2011; Oh et al., 2011; Zhang et al., 2014). The sequential aspect of translation in itself could be sufficient to promote formation of these intermediates, thus shaping the energy landscape of cotranslational folding.

Several computational methods have been used to explore cotranslational aspects of protein folding (Elcock, 2006; Ellis et al., 2010b,c; Huard et al., 2006; Jefferys et al., 2010; Lu and Liang, 2008; Qin and Zhou, 2014; Srinivasan and Rose, 1995). Sequential molecular dynamics revealed the existence of alternative fold pathways and peptide intermediates that resemble the native structure (Elcock, 2006), corroborating experimental data. Modelling results obtained using the original SAINT (Sequential Algorithm Initiated at the Nitrogen Terminus) showed that directional predictions starting from the nitrogen terminus generate better results than sequential predictions beginning from the carbon terminus (Ellis et al., 2010b). This suggests that building models sequentially is not sufficient to observe modelling improvements, but that the biological direction of translation needs to be taken into account for better model generation. This corroborates the notion that translation kinetics play a crucial role in protein folding.

In order to investigate how cotranslational protein folding can be used to improve structure prediction, we have developed SAINT2, a completely independent cotranslational fragment-assembly structure predictor. SAINT2 differs from conventional fragment-assembly approaches as it is able to perform structure predictions sequentially, in a cotranslational fashion. SAINT2 uses fragment libraries composed of fragments of different lengths (refer to Chapter 2) and a predicted residue-residue contact potential (refer to Chapter 3).

We have used SAINT2 to generate 100,000 decoys for each of the 43 proteins in our Training Data set. A correct answer (TM-Score > 0.5) was produced for 25 of these proteins. We find that SAINT2 requires fewer than 10,000 decoys to generate a correct model for 20 cases. When assessing the number of decoys required to generate a "best" model (i.e. a model nearly as good as the best in the 100,000 set), SAINT2 requires fewer than 20,000 decoys for 18 cases. Contrary to our expectations, the number of decoys required to produce a "best" model was independent of protein length.

To investigate the effect of cotranslational protein folding in structure prediction, we have generated 10,000 decoys for a validation set of 41 proteins, using three different approaches: a sequential approach from N to C-terminus (SAINT2 Cotranslational), a non-sequential approach (SAINT2 *In vitro*), and a reverse sequential approach from C to N-terminus (SAINT2 Reverse). We show that cotranslational predictions generate better results when compared to *in vitro* and reverse predictions. Compared to the *in vitro* version, SAINT2 Cotranslational generated individual decoys between 50% to 250% faster. SAINT2 Cotranslational generates a correct answer in 17 of the 41 cases (Validation set) whereas SAINT2 *In vitro* generates correct answers for only 13 cases. The observed improvement using SAINT2 Cotranslational is consistent throughout all protein lengths in our validation set and is most pronounced for the proteins in the set that belong to the α/β SCOP class.

We find that the observed improvements are direction-dependent. Predictions performed sequentially in a non-biological direction (SAINT2 Reverse) also produced correct answers

for 17 cases, out of which 11 were worse when compared to SAINT2 Cotranslational. This corroborates previous findings (Ellis et al., 2010b).

We also tested whether the benefits of performing *de novo* protein structure prediction cotranslationally are consistent for different proteins of the same protein family. We generated predictions using SAINT2 Cotranslational and SAINT2 *In vitro* for nine proteins belonging to the Pfam family *TatD related DNAses*. We observed that the difference between the TM-Score of the best decoy generated using SAINT2 Cotranslational and SAINT2 *In vitro* is consistent for all nine proteins in this data set.

Overall, our findings suggest that cotranslational aspects of protein folding play a part in restricting the conformational space and, therefore, can be used to improve *de novo* protein structure prediction. Our cotranslational approach built individual models faster and, in general, of better quality compared to our non-sequential approach. Our results corroborate the cotranslational protein folding hypothesis and highlight the importance of considering additional biological information to improve model generation.

We investigated the relationship between four structural signatures for cotranslational protein folding and the benefits of cotranslational structure prediction for 41 proteins in our Validation Data Set. No correlation was observed between these structural signatures and the difference between the TM-Score of the best decoy generated by SAINT2 Cotranslational and SAINT2 *In vitro*.

5.2 Methods

5.2.1 Data Sets

For the analyses described in this chapter, we used the two data sets (Training Data Set and PDB-Representative Validation Data Set) that were previously described in Section 2.2.1.

These data sets are comprised of structurally diverse single-domain proteins selected from the PDB (Berman et al., 2000).

We have also used a smaller data set comprised of nine proteins belonging to the Pfam family *TatD related DNAses* (Table 5.1). These proteins were selected based on the availability of a contiguous, high-quality single-domain protein structure for members of the *TatD related DNAses* family.

Table 5.1 *TatD related DNAses* Data Set

SCOP Class α/β			
PDB ID	Pfam Family ID	Protein Length	Resolution
1XWY	PF01026	264 Residues	2.00 Å
3GG7	PF01026	254 Residues	1.50 Å
1J6O	PF01026	268 Residues	1.80 Å
1ZZM	PF01026	259 Residues	1.80 Å
4PE8	PF01026	260 Residues	2.89 Å
4P5U	PF01026	262 Residues	2.00 Å
1YIX	PF01026	265 Residues	1.90 Å
2XIO	PF01026	301 Residues	1.19 Å
3IPW	PF01026	325 Residues	1.95 Å

The nine proteins comprising the *TatD related DNAses* data set. Proteins are single-domain, single chain, and belong to the same Pfam family.

5.2.2 Decoy Generation

Sequence-to-structure Pipeline Parameters

We have used the same parameters described in Chapter 4 within our sequence-to-structure pipeline. Secondary structure predictions were generated using PSIPRED V3 (Jones, 1999). Torsion angle predictions were generated using Spine-X (version 2.0) (Faraggi et al., 2009, 2012). Fragment libraries were generated using Flib (de Oliveira et al., 2015). Contact predictions were generated using metaPSICOV (Jones et al., 2014) when enough sequence information was available. For more details, refer to Section 4.2.1.

SAINT2 Parameters

For the analyses described in this chapter, we have used a linear mover with 10,000 moves during extrusion and 1,000 moves post-extrusion (SAINT2 Cotranslational and Reverse modes). For analyses using the SAINT2 *In vitro* mode, 11,000 moves were used. The scoring weights used to run SAINT2 are described in Section 4.2.18.

5.2.3 Validation

TM-Score (Xu and Zhang, 2010; Zhang and Skolnick, 2004) is used to evaluate the quality of the decoys generated by SAINT2. A TM-Score of 0.5 or higher is considered to be a correct answer (Xu and Zhang, 2010).

In Section 4.2.3, we described three quality measures (TM-Score Best, TM-Score Top-5, and Average TM-Score) that were used in our subsequent analyses to evaluate the quality of decoy ensembles generated by SAINT2.

5.2.4 Signatures of Cotranslational Protein Folding

In this section, we describe four different structure measures have previously been developed to identify proteins that fold cotranslationally (Deane et al., 2007; Saunders et al., 2011).

Mean Central Residue

The Mean Central Residue (MCR) is an estimate of the residue position closest to the centroid of a given protein structure. The MCR is normalized between 0 and 1. An MCR of less than 0.5 was suggested to identify proteins that fold in a cotranslational fashion. The mean central residue is a weighted average of the sequence positions of the eight protein residues closest to its centroid. The MCR is given by the formula:

$$MCR = \frac{\sum_{i=1}^8 \frac{Closest(i)}{\delta(R_{Closest(i)}, Cen)}}{n \times \sum_{i=1}^8 \frac{1}{\delta(R_{Closest(i)}, Cen)}} \quad (5.1)$$

Where:

- $Closest(i)$ is the sequential index of the i th closest residue to the protein's centroid (Cen).
- $\delta(R_{Closest(i)}, Cen)$ is the distance (Å) between the i th closest residue and the protein's centroid.
- n is the protein's length.

Relative distance of the N and C termini to the protein's centroid

The relative distance of the N and C termini to the protein's centroid (NC_{cent}) compares the centrality of the protein's termini. A negative NC_{cent} is indicative of an N-terminus that is more centrally oriented than the C-terminus. A negative NC_{cent} was suggested to

identify proteins that fold cotranslationally (Saunders et al., 2011). The NC_{cent} is given by the formula:

$$NC_{cent} = \log \frac{\delta(N-terminal, Cen)}{\delta(C-terminal, Cen)} \quad (5.2)$$

Where $\delta(N-terminal, Cen)$ and $\delta(C-terminal, Cen)$ are the distances between the centroid and the N-terminus and C-terminus, respectively.

Hydrophobic Moment of Inertia

The Hydrophobic Moment of Inertia (MoI) is a measure of structural compactness. High protein compactness is thought to be associated with cotranslational protein folding (Saunders et al., 2011). Smaller hydrophobic MoIs are indicative of higher protein compactness. To compute the hydrophobic MoI we only consider the hydrophobic residues, as given by:

$$MoI = \frac{1}{N_{hydro}} \sum_i^{N_{hydro}} \delta(HR_i, Cen)^2 \quad (5.3)$$

Where:

- N_{hydro} is the number of hydrophobic residues of the target.
- HR_i is the i -th hydrophobic residue of the target.
- $\delta(HR_i, Cen)$ is the distance (Å) between the i -th hydrophobic residue and the protein's centroid.

Sum of Logarithmic Ratios of protein contacts

If a given protein folds in a cotranslational fashion, one would expect a residue to be more likely to form contacts with residues that have already been extruded. The Sum of

Logarithmic Ratios of protein contacts (SLR) compares the likelihood a given residue i to form a contact with a lower index residue (sequentially closer to the N-terminus) against how likely the residue $n-i$ is to form a contact with a residue with a higher index (sequentially closer to the C-terminus). A positive SLR was suggested to indicate that a protein folds in a cotranslational fashion.

In this case, a previous contact of residue i (from the N-terminus) is defined as a residue j , where $j < i - 5$, and where the distance between residues i and j is lesser than $13\text{\AA}$. For this calculation, we enforce $i > 6$ because previous contacts for the first five N-terminal residues occur solely due to sequence proximity. We define A_i^N as the total number of previous contacts for residue i . Analogously, we define a forward contact of residue $n - i$ as a residue j , where $j > n - i - 5$, and where the distance between residues $n - i$ and j is lesser than $13\text{\AA}$. Once again, we enforce $i < n - 5$ to eliminate trivial forward contacts (a trivial contact is established between two residues that are less than 5 residues apart). The total number of forward contacts can be defined as A_{n-i}^C .

The chain is parsed simultaneously and symmetrically from both termini. To prevent division by zero in our formula, we group residues whenever the sums of both A_i^N and A_{n-i}^C are non-zero. We define G as the number of groups, k as the index of the groups, and I_k as the number residues of group k . The Sum of Logarithmic Ratios of protein contacts is given by:

$$SLR = \frac{1}{G} \sum_{k=1}^G (\log(\sum_{i=1}^{I_k} A_i^N) - \log(\sum_{i=1}^{I_k} A_{n-i}^C)) \quad (5.4)$$

5.3 Results

5.3.1 Determining the Number of Decoys Required by SAINT2

The general protocol for fragment-based *de novo* protein structure prediction is to generate as many decoys as possible for a target. There has been little analysis done to assess how many decoys are needed for each target in order to obtain a good answer. Most papers suggest that the longer the protein, the larger the number of decoys needed (Kim et al., 2014; Simoncini and Zhang, 2013; Xu and Zhang, 2012b).

SAINT2, in its standard cotranslational mode, was used to generate 100,000 decoys for each of the 43 proteins in our Training Data set. Correct answers (TM-Score to native structure > 0.5) were generated for 25 targets. These 25 cases were used to estimate how many decoys are necessary to obtain a correct answer and a "best" model (Figure 5.1). A "best" model is a decoy within 0.05 TM-Score units of the best possible decoy in the 100,000 decoy ensemble.

To estimate these numbers, we took samples of varying sizes from the decoy population. We varied the sample size from 100 to 100,000 by increments of 100. For each sample size, we take 100 samples and count how many of these samples contain a correct answer or a "best" model. For each protein, we identify the smallest sample size for which 95% of the samples contain a correct answer and use that sample size as an estimate for the number of decoys required by SAINT2 to produce a correct answer. We estimate the number of decoys required by SAINT2 to produce a best model in an analogous fashion, by identifying the smallest sample size for which 95% of the samples contain a "best" model.

Our results show that when fewer than 10,000 decoys are generated, SAINT2 produces a correct answer for 20 targets. For 18 out of 25 targets in our Training Data set, SAINT2 produces a model as good as any in the 100,000 ensemble after generating fewer than 20,000 decoys.

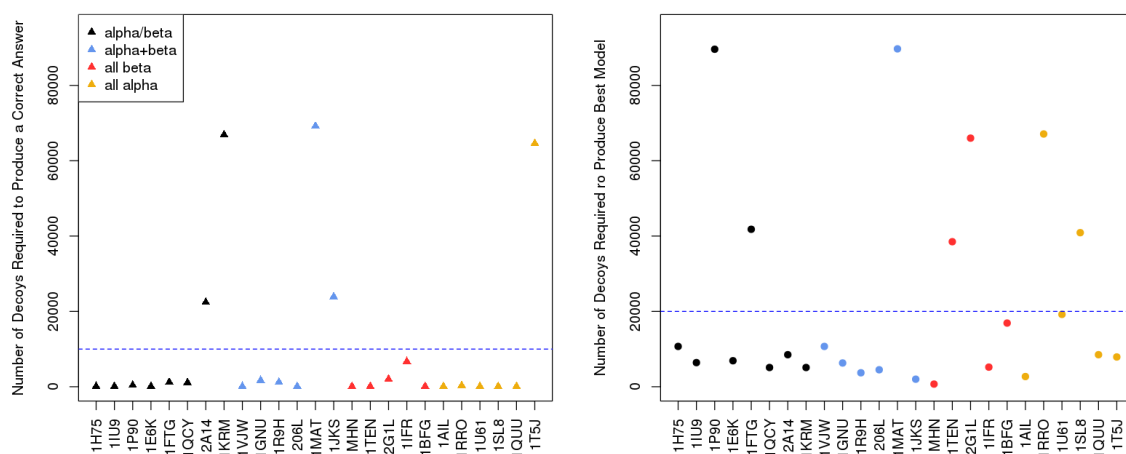


Fig. 5.1 We used SAINT2 to generate 100,000 decoys for each target in our Training Data set and have estimated both the number of decoys required to produce a correct answer (left) and to produce a "best" model (right). A correct answer is one with TM-Score to the native structure greater than 0.5 and a "best" model is a decoy within 0.05 TM-Score units of the best possible solution produced in the 100,000 decoy ensemble. Proteins are coloured according to their SCOP classes.

We analysed whether sequence features could be used to estimate the number of decoys required to produce a correct answer (Figure 5.2). SAINT2 generates a correct answer for all proteins in our training data set shorter than 250 residues when fewer than 10,000 decoys have been produced. Results for the SCOP classes show that SAINT2 can generate a correct answer for most All α and All β proteins in our data set when fewer than 5,000 decoys are produced. However, no correct answer was produced for All β proteins longer than 250 residues.

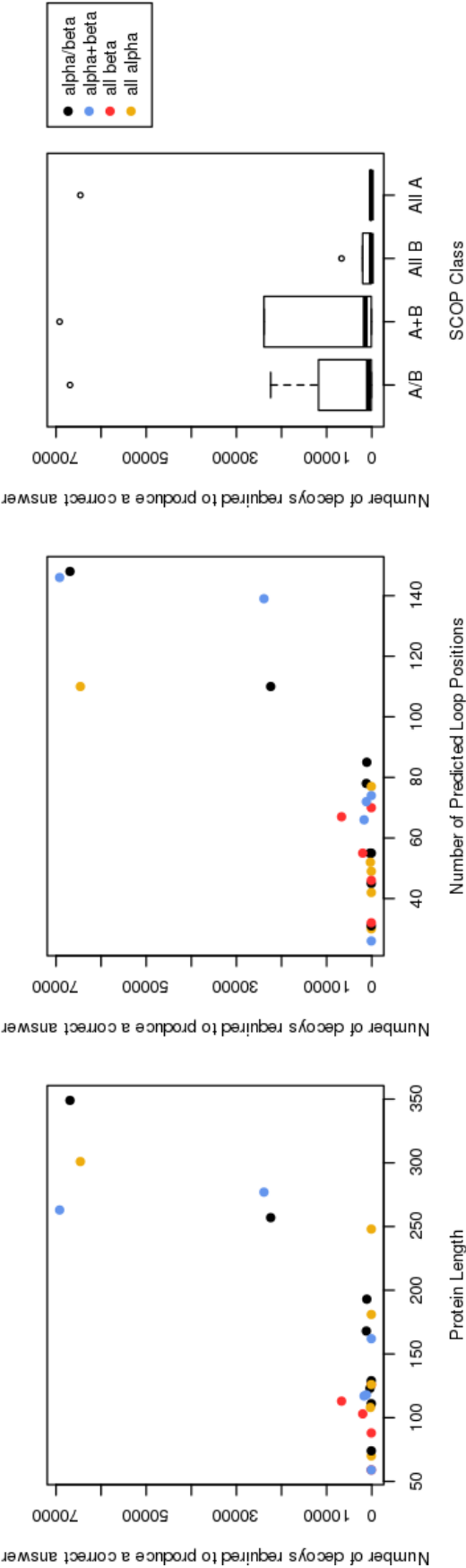


Fig. 5.2 Correlation between the number of decoys required by SAINT2 to produce a correct answer and three sequence-based features. The x-axis represents each of our sequential features (protein length, number of predicted loop positions and SCOP class). The y-axis is the number of decoys required to generate a correct answer for 25 targets in our Training data set. SAINT2 did not produce a correct answer for the remaining 18 targets.

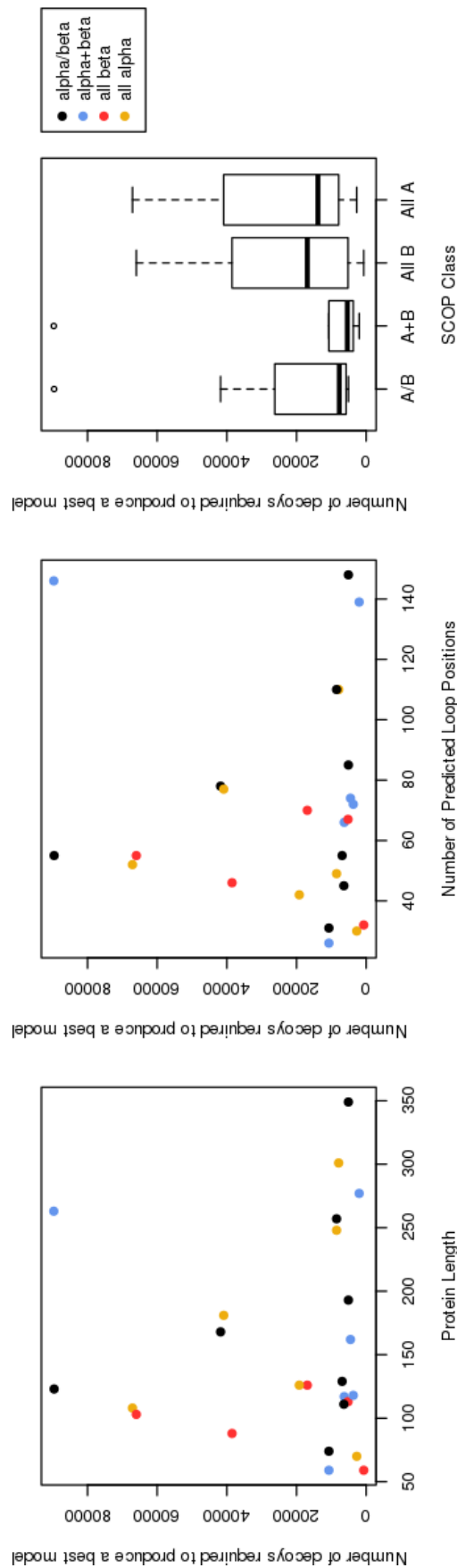


Fig. 5.3 Correlation between the number of decoys required by SAINT2 to produce a "best" model and three sequence-based features. The x-axis represents each of our sequential features (protein length, number of predicted loop positions and SCOP class). The y-axis is the number of decoy required by SAINT2 to produce a "best" model for the 25 targets in our Training data set for which SAINT2 was capable of generating a correct answer.

These sequence features were also compared against the number of decoys required to produce a "best" model for each protein in our Training Data Set (Figure 5.3). We observe no correlation between our estimate for the number of decoys required to produce a "best" model and protein length or number of loop positions. Results for the SCOP classes show that $\alpha + \beta$ proteins tend to require fewer decoys compared to other SCOP classes in order for a "best" model to be generated.

We also tested for a relationship between the number of decoys required to produce a correct answer with fragment library quality and the precision of predicted residue-residue contacts and observed no correlations (Figures 5.4, 5.5). No correlation was observed when performing a similar comparison using the number of decoys required to produce a "best" model (data not shown).

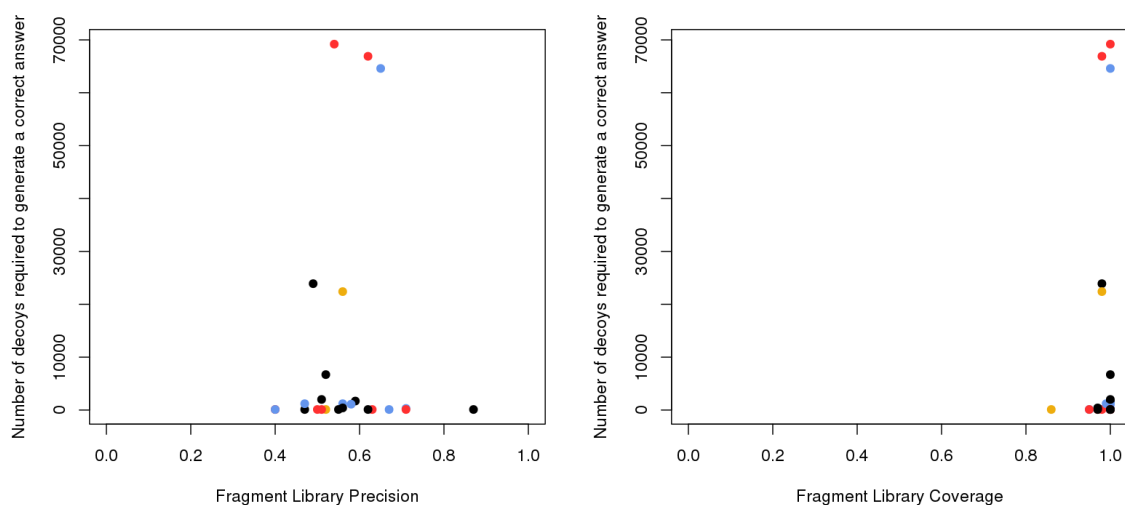


Fig. 5.4 Relationship between fragment library precision/coverage and the number of decoys required to produce a correct answer using SAINT2. Data is shown for the 25 proteins in our training data set. Colours represent each of the SCOP classes.

We have assessed the number of decoys that need to be generated by SAINT2 so that a correct answer or a "best" model is produced. Our results reveal that generating 10,000 decoys is sufficient to ensure that a correct answer is produced for most cases and that

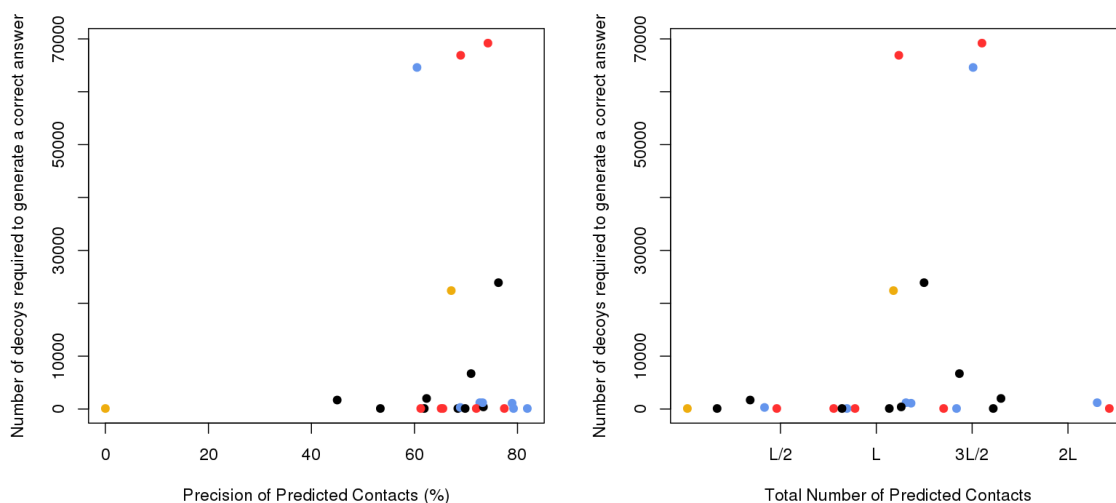


Fig. 5.5 Relationship between precision of predicted contacts and the number of decoys required to produce a correct answer using SAINT2. Data is shown for the 25 proteins in our training data set. Colours represent each of the SCOP classes.

generating more than 20,000 decoys does not lead to significant improvements in 18 out of 25 cases. We have also shown that the number of decoys required to produce a "best" model is independent of protein length and is lower for proteins belonging to $\alpha + \beta$ SCOP classes. These results allow us to estimate the number of decoys that SAINT2 should generate for a given target.

5.3.2 Impact of Sequentiality on the Quality of Models Generated

If the underlying hypothesis of cotranslational protein folding is correct, SAINT2 Cotranslational mode should be more efficient at searching the conformational space than the other modes of SAINT2. We used our three versions of SAINT2 to evaluate whether such an effect can be observed in a structure prediction context (Figures 5.6, 5.7).

We generated 10,000 decoys for each of the 41 targets in our validation set using both SAINT2 Cotranslational and SAINT2 *In vitro*. For both, identical fragment libraries, predicted residue-residue contacts and number of moves to generate a decoy were used. Decoys

generated using SAINT2 Cotranslational Mode tend to present a higher TM-Score Best when compared to SAINT2 *In vitro* mode. SAINT2 Cotranslational produced a correct answer (TM-Score > 0.5) for 17 cases whereas SAINT2 *In vitro* produced a correct answer for only 13 cases. There are no cases where a correct answer was generated using the SAINT2 *In vitro* mode, but was not produced using SAINT2 Cotranslational mode. SAINT2 Cotranslational predictions were better in 10 out of the 13 cases for which SAINT2 *In vitro* also generated a correct answer.

These trends are reproduced if we look at the Top 5 predictions (TM-Score Top 5) as opposed to the best TMScore in the set. SAINT2 Cotranslational gives a higher TM-Score in its Top 5 for 33 of the 41 cases, generating 11 correct answers (TM-Score Top 5 > 0.5). SAINT2 *In vitro* has a correct answer in the Top 5 for 4 of the 41 cases and in no cases where SAINT2 Cotranslational did not generate a correct answer.

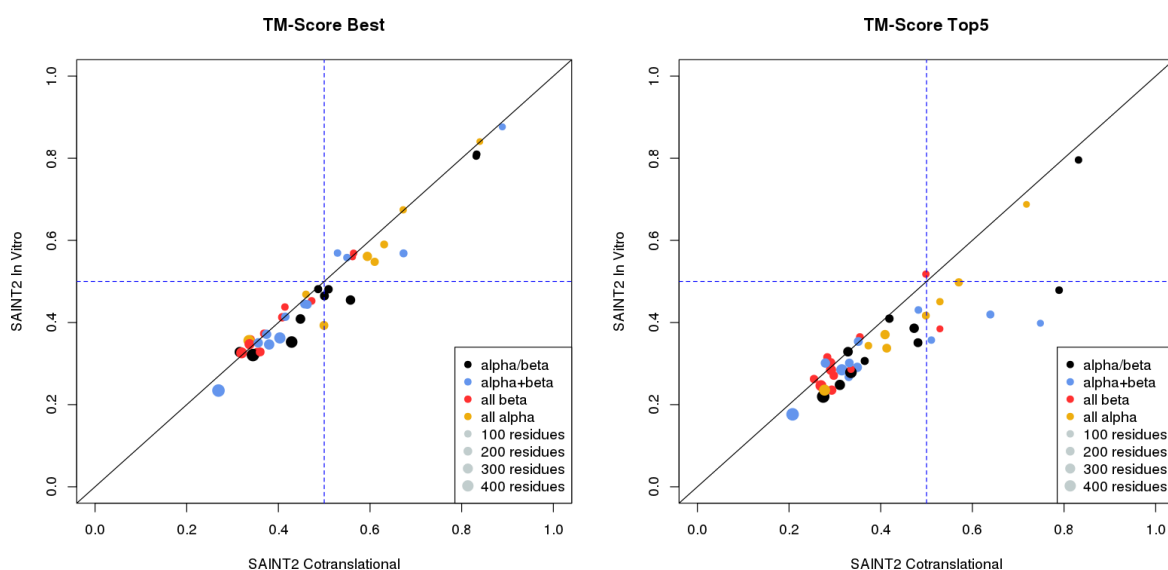


Fig. 5.6 Comparison of the TM-Score Best (left) and TM-Score Top 5 (right) of the 41 proteins in our validation set obtained using SAINT2 Cotranslational (x-axis) against SAINT2 *In vitro* (y-axis). Points below the black line indicate cases where SAINT2 Cotranslational performs better than SAINT2 *In vitro*. Each point is coloured according to the target protein's SCOP class and its size indicates protein length.

Protein synthesis has a direction; residues are extruded starting from the nitrogen terminus and ending at the carbon terminus. We assessed how this directionality affects the results

generated by SAINT2. We compared SAINT2 Cotranslational mode to SAINT2 Reverse mode (Figure 5.7). SAINT2 Reverse performs predictions in a sequential fashion, but in the reverse direction to that observed in nature. SAINT2 Reverse was used to generate 10,000 decoys for each of the 41 targets in our validation data set. Decoys generated using SAINT2 Cotranslational achieve a higher TM-Score Best for 27 out of 41 cases compared to SAINT2 Reverse. Out of the 41 cases, SAINT2 Reverse also produced a correct answer (TM-Score Best > 0.5) for 17 cases. However, when we compare the results for the Top-5 TM-Score, SAINT2 Cotranslational performs better in 35 of the 41 cases, generating 11 correct answers (Top-5 TM-Score > 0.5). SAINT2 Reverse generates a correct answer in 4 of the 41 cases.

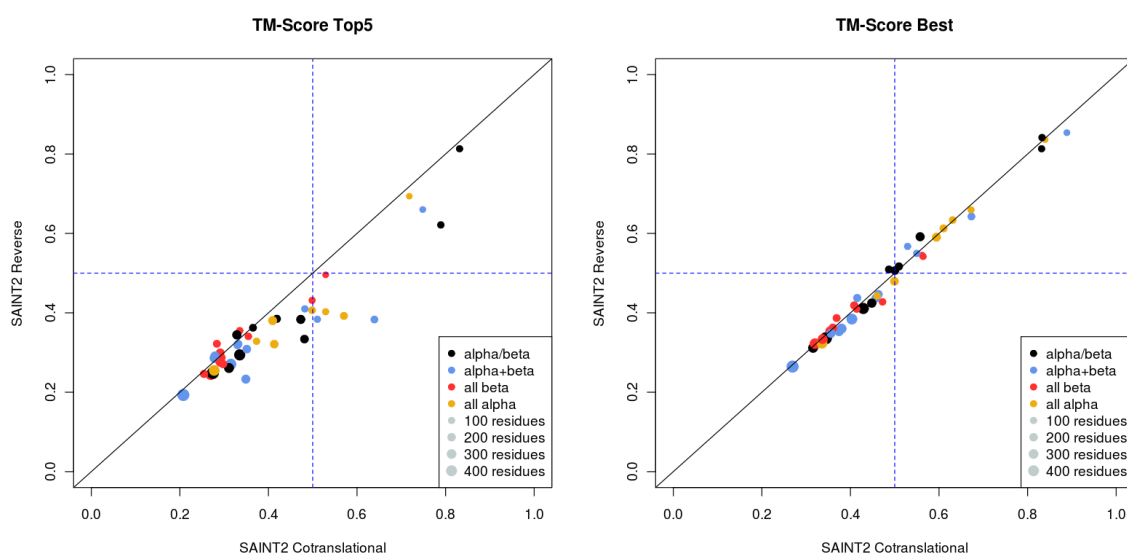


Fig. 5.7 Comparison of the TOP-5 TM-Score (left) and TM-Score Best (right) of the 41 proteins in our validation set obtained using SAINT2 Cotranslational (x-axis) against SAINT2 Reverse (y-axis). Points below the black line indicate cases where SAINT2 Cotranslational performs better than SAINT2 Reverse. Each point is coloured according to the target protein's SCOP class.

It has been suggested that proteins belonging to the α/β SCOP class and longer proteins are more likely to fold cotranslationally (Deane et al., 2007; Saunders et al., 2011). We used our SAINT2 versions to assess the relationship between length/SCOP class and the improvements observed by performing predictions cotranslationally. SAINT2 Cotranslational

presents a higher TM-Score Best for 9 out of 10 α/β proteins in our validation set (as seen in Figure 5.6). Furthermore, the effect seems to be stronger (i.e. the differences between the TM-Score Best of SAINT2 Cotranslational and SAINT2 *In Vitro* are larger) for α/β . We observe no relationship between the model improvement and protein length.

We have also compared the run time of our different versions of SAINT2 (Figure 5.8). The time complexity of our fragment assembly approach is quadratic on the number of atoms (see Section 4.2.22). Given that SAINT2 Cotranslational performs many moves on a reduced number of atoms (a portion of the full protein chain), it can generate decoys up to 2.5 times faster than SAINT2 *In Vitro*.

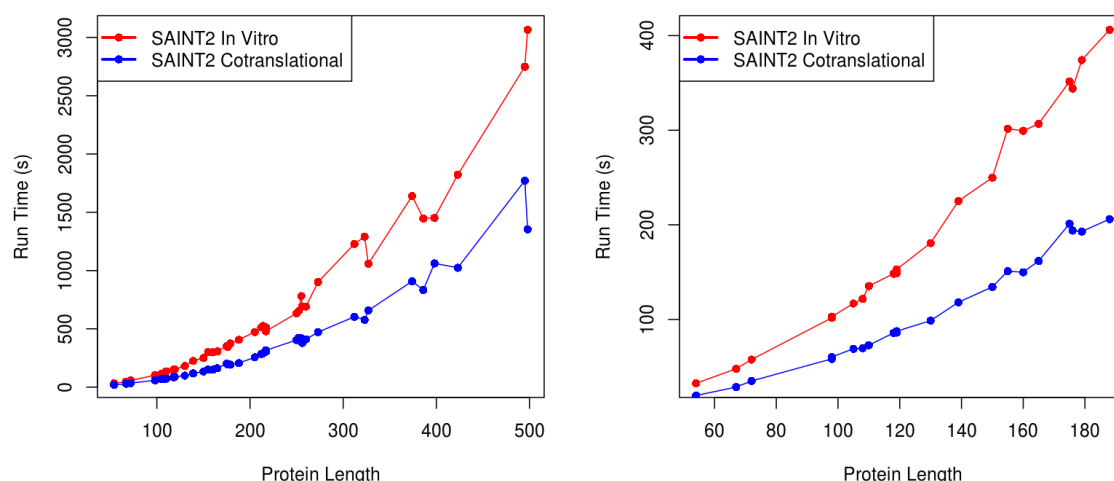


Fig. 5.8 Run times for an individual run of SAINT2 Cotranslational (blue) and SAINT2 *In vitro* (red) for the 41 proteins in our Validation Data Set.

Overall, our results show that SAINT2 Cotranslational employs an efficient search approach; it is faster and produces better models for a majority of modelling cases. There are no cases where a conventional search strategy (SAINT2 *In Vitro*) is capable of producing a model that is significantly better than the ones generated by SAINT2 Cotranslational. Yet, there are several cases where a correct answer was produced cotranslationally that would not

have been generated otherwise. The improvement in modelling results was observed across all protein lengths and across all SCOP classes represented in our Validation Data set.

5.3.3 Improvement Cotranslational is Consistent in a Protein Family

We have assessed the improvement of predictions generated by SAINT2 Cotranslational against SAINT2 *In vitro* for nine proteins belonging to the same protein family. We have chosen the Pfam family of protein 1XWY (TatD related DNases family) as this protein is a long (264 residues) α/β protein for which a correct answer was produced using SAINT2 Cotranslational and not produced using SAINT2 *In vitro*.

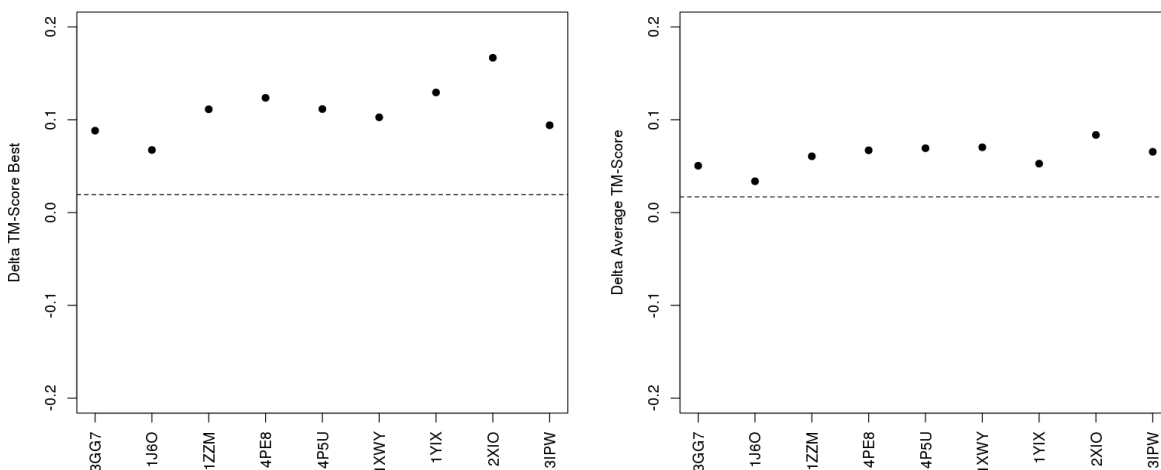


Fig. 5.9 Δ TM-Score Best (left) and Δ Average TM-Score (right) for nine proteins belonging to the Pfam family *TatD related DNases* as generated by SAINT2. We generated 1,000 decoys for the nine proteins in the *TatD related DNases* data set and computed the difference of TM-Score Best and Average TM-Score of the decoys generated by SAINT2 Cotranslational and SAINT2 *In vitro*. Proteins are sorted according to their length. The observed improvements obtained by performing predictions cotranslationally are consistent and significant for all nine proteins in this family. The dashed lines represent the mean values of Δ TM-Score Best and Δ Average TM-Score as computed for the 41 proteins in our Validation Data set.

For each of the nine proteins in the TatD related DNases data set, we have generated 1,000 decoys using SAINT2 Cotranslational and SAINT2 *In vitro*. We use only 1,000 decoys for this analysis as less than 1,000 decoys were required to generate a correct model for 1XWY. For all cases, the TM-Score Best obtained by SAINT2 Cotranslational is significantly larger

than the TM-Score Best obtained by SAINT2 *In Vitro* (Figure 5.9). SAINT2 Cotranslational consistently generates better models for all cases, which suggests that cotranslational protein folding is a conserved folding mechanism for proteins in this family.

5.3.4 Exploring Signatures of Cotranslational Protein Folding

Given SAINT2's cotranslational character and our underlying hypotheses, one would expect SAINT2 Cotranslational to produce better results than SAINT2 *In vitro* for proteins that attain their native structure in a cotranslational manner. However, to our knowledge, there is no curated database that lists proteins that have been shown to fold in that fashion. For that reason, we have used structural descriptors described in Deane et al. (2007); Saunders et al. (2011) to try and predict which proteins in our Validation Data set fold cotranslationally, thus predicting which targets would be better modelled by SAINT2 Cotranslational.

We computed four measures (MCR, NCcent, MoI, and SLR) based on the structure of the 41 targets in our Validation Data Set. We compared the correlation between each of these measures and the Δ TM-Score Best (Figure 5.10) for all targets in the Validation Data Set. Our results show no clear association between the Δ TM-Score Best and MCR/MoI/NCcent. All of the targets with a Δ TM-Score Best greater than 0.05 presented a positive SLR, yet this could be an artefact due to the small sample size.

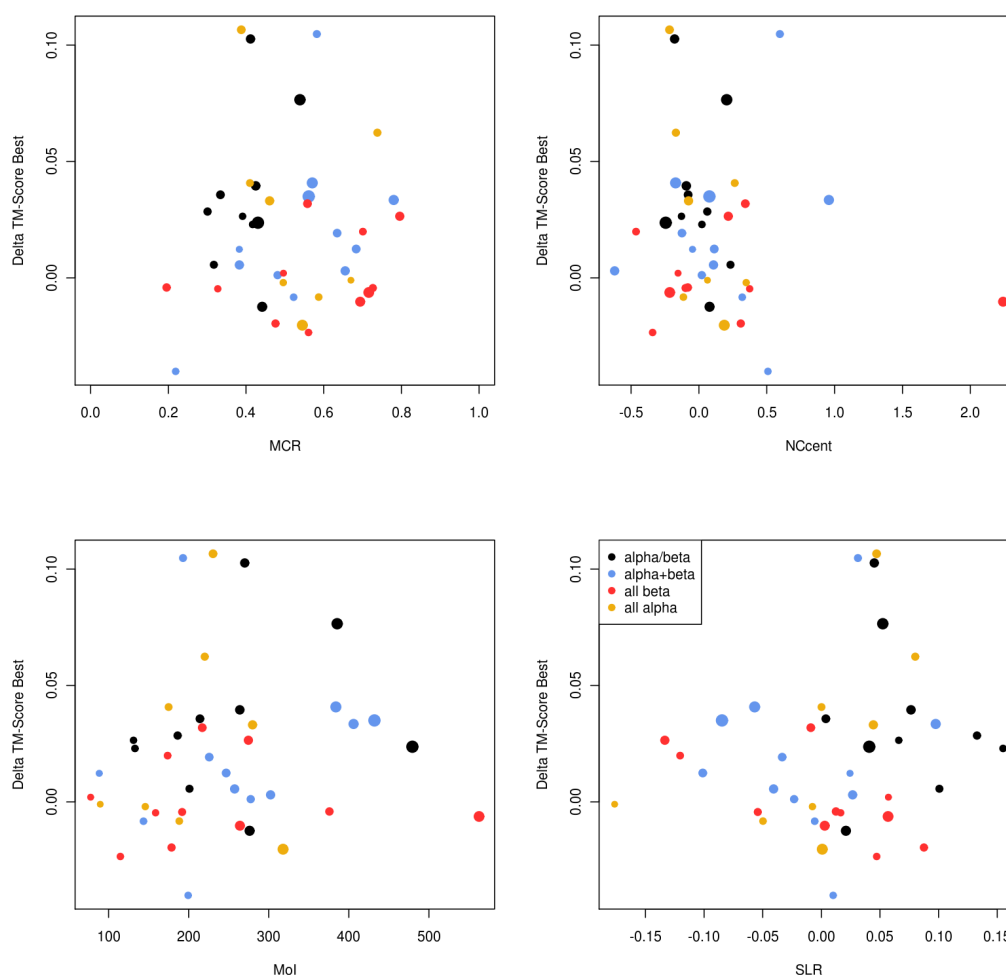


Fig. 5.10 Relationship between four structural signatures for cotranslational protein folding and the Δ TM-Score Best obtained for 41 proteins in our Validation Data Set. Proteins are coloured according to their SCOP classes. MCR, NCcent and MoI are thought to be inversely proportional to cotranslational protein folding (i.e. smaller values of MCR/NCcent/MoI suggest that the target folds in a cotranslational fashion). No correlation is observed between these three measures and Δ TM-Score Best. SLR values greater than 0.5 suggest that the target is folding in a cotranslational fashion, which we observe for all targets with a significant Δ TM-Score Best (>0.05).

5.4 Discussion

In this Chapter, we have investigated the behaviour of a cotranslational fragment-based *de novo* structure prediction method SAINT2. Our initial studies found that the number of decoys required by SAINT2 to generate a "best" model is not dependent on protein length. However, the number of decoys required to produce a correct answer does show some dependence on protein length. This follows the general perception that longer targets require a larger number of decoys in order for a good answer to be generated but not that a larger number of decoys are required for longer proteins to generate a high quality prediction (Kim et al., 2014; Simoncini and Zhang, 2013; Xu and Zhang, 2012b). No correct answer was produced for any of the proteins longer than 350 residues in our set, therefore our results cannot be extrapolated for longer proteins.

SAINT2 Cotranslational, in most cases where a correct answer is produced at all, is able to produce it when less than 10,000 decoys are generated. This is far fewer than the 100,000 decoys or more many methods generate for each target (Kim et al., 2014; Simoncini and Zhang, 2013; Xu and Zhang, 2012b). This suggests that SAINT2 is either more efficient at sampling the conformational space or that other methods are generating an excessive number of decoys. Assuming that first to be true, these results corroborate the idea that cotranslational aspects can be used to improve search heuristics.

Experimental results indicate that folding *in vivo* is faster than refolding, thus cotranslational protein folding is thought to help proteins to search the conformational space efficiently. This is also seen computationally as SAINT2 Cotranslational not only generates better predictions, but is also faster than SAINT2 *In Vitro*. *In vitro* predictors always perform moves and score a full protein conformation. SAINT2 Cotranslational optimizes performance by performing moves and scoring a peptide that is shorter than the target protein. In the analyses in this manuscript, where we use the same number of moves for the two methods, SAINT2

Cotranslational is capable of generating a decoy between ≈ 1.5 and 2.5 times faster than SAINT2 *In Vitro*.

One possible issue with using a cotranslational protein structure predictor is the idea of local entrapment, that by folding the N-terminal residues before the rest of the protein, they could become trapped in a local minimum that is not relevant for the global fold. This type of entrapment does not appear to influence our methodology as for our validation set, SAINT2 Cotranslational generates more correct models and better models than SAINT2 *In Vitro*. There are no cases where SAINT2 *In Vitro* predictions are significantly (TM-Score difference > 0.05) better than SAINT2 Cotranslational.

The fact that SAINT2 Cotranslational generates better predictions than SAINT2 *In Vitro* provides further evidence that some proteins may fold cotranslationally. Instead of considering SAINT2 as a structure predictor, it could be used to assess if a protein is folding cotranslationally by evaluating the improvement observed for cotranslational predictions when compared to reverse and *in vitro* predictions. We observed a consistent improvement across nine proteins belonging to the same protein family. This suggests that cotranslational protein folding may be a conserved folding mechanism for protein families.

Our results further corroborate the cotranslational protein folding hypothesis. Furthermore, they suggest that *de novo* structure prediction could be improved by the use of cotranslation. Performing *de novo* structure prediction cotranslationally produces better results, significantly reduces the number of decoys required to generate a good answer and reduces the time needed to generate each decoy.

Chapter 6

Future Work

In this thesis, we have explored how biology can be used as a source of inspiration to improve *de novo* protein structure prediction. In the following sections, potential future directions of this work will be discussed.

6.1 Building even better fragment libraries

It has recently been reported that deep learning can be used to improve the prediction of one dimensional protein features such as secondary structure and torsion angles (Heffernan et al., 2015). The improvement observed was small and it would be interesting to assess the effect of using more accurate predicted secondary structure and torsion angles to build fragment libraries.

In this work, we varied the RMSD cutoff that defines what a good fragment is. Several works (Gront et al., 2011; Kalev and Habeck, 2011) suggest that a $1.5\text{\AA}$ is a good cutoff to ensure that a fragment will correctly represent the native structure. Nonetheless, no formal assessment has been performed to investigate whether this is true. A recent study suggests that the aggregate effect of small errors in torsion angles can lead to large deviations from the native structure (Heffernan et al., 2015). Therefore, it would be interesting to investigate

whether we can build protein models from fragments that deviate by approximately 1.5Å from the native structure. Our preliminary results show that this is not necessarily the case, which suggests that further investigation is required. In light of this hypothesis, if a cutoff of 1.5Å is, indeed, too lenient to define what a good fragment is, the completeness of the fragment space would need to be reassessed in terms of a more stringent fragment RMSD cutoff.

An initial assessment revealed that some cases in our Training data set would not be correctly modelled even if we chose the best fragments generated by Flib to predict the target's structure. Therefore, we would like to assess the proportion of *de novo* protein structure prediction cases that can be correctly modelled using Flib in conjunction with any fragment-assembly approach. This could be performed by extending our work to a larger, non-redundant data set such as PISCES (Wang and Dunbrack, 2003). We could examine the fragment libraries generated by Flib for this larger data set to estimate how many proteins of known structure can be correctly modelled by Flib.

We have reported that loop regions present a challenge for fragment library generation. We performed a preliminary assessment using native fragments (fragments extracted from the native structure) to represent a portion of a protein's loop region and shown that this is sufficient to ensure accurate topology prediction. These results suggest that loop regions may be the reason why structure prediction fails in a large number of cases. Therefore It would be interesting to assess how this limitation could be overcome, i.e. using more fragments to represent loop regions, dedicating more time to sample fragments in those regions, building fragments *in silico* that are not observed in nature.

6.2 Exploring co-evolution information to improve model generation

We estimated that both metaPSICOV and PCONSC2 can generate accurate contact predictions when MSAs have as few as ≈ 500 hundred sequences. Although a more statistically rigorous estimation of this number needs to be performed, it would be interesting to re-evaluate the estimate in Kamisetty et al. (2013) of the number of protein families with no known homologues for which we can produce contact predictions.

Our results have shown that metaPSICOV-HB can be used to perform accurate backbone hydrogen bond contact predictions. Perhaps, one of the most interesting future directions of this work would be to incorporate a predicted backbone hydrogen bond contact potential into SAINT2. This task could be easily performed and tested given SAINT2's highly modular architecture. Given that we will be exploring our ability to predict hydrogen bonds and its usefulness to *de novo* structure prediction, further investigation is also required to assess how many of the non-backbone hydrogen bonds are detected by existing contact predictors and whether similar strategies employed by metaPSICOV-HB could be used to infer these contacts with higher accuracy.

We found that the proportion of satisfied predicted long-range contacts is a good indicator of whether we have generated a correct answer for a given target or not. It would be interesting to investigate why we cannot satisfy a majority of the predicted long-range contacts for some cases. This suggests that SAINT2 is failing to sample near-native conformations in these cases. There are several reasons why these conformations may not be sampled, such as bad fragment libraries and imprecision in our scoring function. If we can determine the cause of this sampling bias, we will be one step closer to overcoming this limitation.

We implemented a naive classifier to discriminate between good and bad decoys. Another future direction of this work would be to explore more powerful techniques to use predicted contacts to perform model quality assessment.

6.3 Improving SAINT2

One of the main limitations of SAINT2, at the moment, is its poor ability to model All β proteins. This limitation needs to be addressed in order to ensure that SAINT2 achieves comparable results to state-of-the-art protein structure prediction software. To address this issue, we could test potentials that enforce the correct β -pairing for β -sheets. This could be facilitated by the information provided by bbcontacts and metaPSICOV-HB. Another potential way of addressing this issue is to implement a hydrogen bonding potential, given the importance of hydrogen bonding to β -sheet formation. Note that the lack of a hydrogen bonding potential is not a limiting factor for the accurate modelling of α -helices as their backbone hydrogen bonds are modelled by the structural fragments used in SAINT2.

Our sequence-to-structure prediction pipeline could benefit from using a clustering method to perform model quality assessment. These methods have been shown to produce better results than the single decoy approach used described in this Thesis. Furthermore, side chain reconstruction and a refinement step will be crucial if we want to participate in further iteration of CASP. The focus of this work has been improving model generation, given that if model generation fails, model quality assessment becomes pointless and model refinement is unlikely to bring these poor quality models any closer to the actual native structure.

Several potentials have been implemented into SAINT2 since the initial estimation of it scoring weights. For that reason, a more thorough initial scoring weight estimation needs to be carried out. We propose to carry this estimation out by selecting a larger subset of the PISCES Training set and performing permutations to generate a sufficiently large number of decoys evenly spread across the range of possible TM-Scores. This larger Decoy Training

set would then be used to estimate the scoring weights that maximise the correlation between our scoring function and the observed TM-Scores. Instead of using a Downhill search to perform this, more powerful machine learning techniques should be used. More importantly, a 10-fold cross validation would need to be performed to ensure that we are not overtraining.

6.4 Exploring other cotranslational features in *de novo* protein structure prediction

One interesting feature that has not been explored in this work is biologically inspired Mover profiles. As discussed in Chapter 5, codon information can be used to determine translation kinetics. This feature should be explored in SAINT2. For instance, the effect of codons that can slow down and even pause translation could be simulated by decreasing the rate at which we perform extrusion steps in between move steps. To perform this assessment, however, we would need a precise way of determining which codons have an effect on translation kinetics. More importantly, we would need to compile a set of proteins that are heavily dependent on cotranslational protein folding to achieve their native structure. Neither of these tasks is trivial.

We have performed an initial assessment of the folding trajectories of decoys generated by SAINT2. Surprisingly, this assessment has shown that the decoys that lead to the best possible models are not necessarily the ones with the best possible intermediate peptide conformations. However, we did observe that intermediates that are correct (TM-Score to native structure > 0.5) are more likely to lead to a final correct model. If we can identify which intermediates are incorrect along the simulation, we could prevent SAINT2 from spending further computational resources on a run that is less likely to succeed. As mentioned previously, the number of satisfied long-range predicted contacts is a good indication of whether a given conformation is bad. Interesting strategies could be explored, where if an

incorrect intermediate is detected, SAINT2 could swap that intermediate conformation for another (i.e. one where more predicted long-range contacts are satisfied).

Another interesting result from our initial assessment of the folding trajectories of SAINT2 is that we observe correct intermediate conformations for several targets for which a correct final model was not produced. We need to ascertain how we can make sure that these correct conformations are sustained throughout a run of SAINT2, thus leading to higher number of cases with a correct answer.

It would also be interesting to explore SAINT2's potential to identify proteins that fold in a cotranslational fashion. At the moment, as far as we know, there is no clear experimental procedure to detect whether a protein needs to attain its native structure cotranslationally. SAINT2, therefore, could be a powerful tool to assist in detecting these proteins by assessing the difference in the TM-Score Best of decoys generated by SAINT2 Cotranslational and SAINT2 *In vivo*. Nonetheless, to assess SAINT2's predictive power we would need to know which of the proteins in a training or validation data set are thought to fold cotranslationally and such a data set could prove difficult to compile. In particular, we know that all transmembrane proteins fold in a cotranslational fashion, so this could potentially be used as an initial test of SAINT2's predictive capacities.

The effect of chaperones, molecular crowding and interaction with the ribosome are also interesting cotranslational features that have not been explored in this work. As our understanding of the molecular mechanisms of cotranslational protein folding grows, we will be able to incorporate each of these features into SAINT2, assessing the impact of considering each of these features to *de novo* protein structure prediction.

References

- Abbass, J. and Nebel, J.-C. (2015). Customised fragments libraries for protein structure prediction based on structural class annotations. BMC Bioinformatics, 16(1):136.
- Agashe, V. R., Shastry, M., and Udgaonkar, J. B. (1995). Initial hydrophobic collapse in the folding of barstar. Nature, 377(6551):754–757.
- 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.
- Andreani, J. and Söding, J. (2015). bbcontacts: prediction of β -strand pairing from direct coupling patterns. Bioinformatics, page btv041.
- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., and Murzin, A. G. (2014). Scop2 prototype: a new approach to protein structure mining. Nucleic acids research, 42(D1):D310–D314.
- Anfinsen, C. B. (1972). Studies on the principles that govern the folding of protein chains.
- Anfinsen, C. B., Haber, E., Sela, M., and White Jr, F. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. Proceedings of the National Academy of Sciences, 47(9):1309.
- Angov, E., Hillier, C. J., Kincaid, R. L., Lyon, J. A., et al. (2008). Heterologous protein expression is enhanced by harmonizing the codon usage frequencies of the target gene with those of the expression host. PloS one, 3(5):e2189.
- Argos, P., Schwarz, J., and Schwarz, J. (1976). An assessment of protein secondary structure prediction methods based on amino acid sequence. Biochimica et Biophysica Acta (BBA)-Protein Structure, 439(2):261–273.
- Avbelj, F. and Baldwin, R. L. (2003). Role of backbone solvation and electrostatics in generating preferred peptide backbone conformations: distributions of phi. Proceedings of the National Academy of Sciences, 100(10):5742–5747.
- Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. (2005). The universal protein resource (uniprot). Nucleic acids research, 33(suppl 1):D154–D159.
- Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. Science, 294(5540):93–96.

- Basharov, M. (2000). Cotranslational folding of proteins. *Biochemistry*, 65(12):1380–1384.
- Basharov, M. (2003). Protein folding. *Journal of cellular and molecular medicine*, 7(3):223–237.
- Berkenpas, M. B., Lawrence, D. A., and Ginsburg, D. (1995). Molecular evolution of plasminogen activator inhibitor-1 functional stability. *The EMBO journal*, 14(13):2969.
- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.
- Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. (1978). The protein data bank: a computer-based archival file for macromolecular structures. *Archives of biochemistry and biophysics*, 185(2):584–591.
- Betancourt, M. R. and Skolnick, J. (2001). Universal similarity measure for comparing protein structures. *Biopolymers*, 59(5):305–309.
- Bevivino, A. E. and Loll, P. J. (2001). An expanded glutamine repeat destabilizes native ataxin-3 structure and mediates formation of parallel β -fibrils. *Proceedings of the National Academy of Sciences*, 98(21):11955–11960.
- Blundell, T., Sternberg, M., and Thornton, J. (1987). Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326:347–352.
- Bonneau, R., Strauss, C. E., Rohl, C. A., Chivian, D., Bradley, P., Malmström, L., Robertson, T., and Baker, D. (2002). De novo prediction of three-dimensional structures for major protein families. *Journal of molecular biology*, 322(1):65–78.
- Bonneau, R., Tsai, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C. E., and Baker, D. (2001). Rosetta in casp4: progress in ab initio protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 45(S5):119–126.
- Bradley, P., Misura, K. M., and Baker, D. (2005). Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871.
- Brenner, S., Chothia, C., Hubbard, T. J., and Murzin, A. G. (1996). *Understanding protein structure: Using scop for fold interpretation*. Elsevier.
- Briët, J. and Harremoës, P. (2009). Properties of classical and quantum jensen-shannon divergence. *Physical review A*, 79(5):052311.
- Browne, W., North, A., Phillips, D., Brew, K., Vanaman, T. C., and Hill, R. L. (1969). A possible three-dimensional structure of bovine α -lactalbumin based on that of hen's egg-white lysozyme. *Journal of molecular biology*, 42(1):65–86.
- Bryngelson, J. D., Onuchic, J. N., Socci, N. D., and Wolynes, P. G. (1995). Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195.
- Cabrita, L. D. and Bottomley, S. P. (2004). Protein expression and refolding—a practical guide to getting the most out of inclusion bodies. *Biotechnology annual review*, 10:31–50.

- Cabrita, L. D., Hsu, S.-T. D., Launay, H., Dobson, C. M., and Christodoulou, J. (2009). Probing ribosome-nascent chain complexes produced in vivo by nmr spectroscopy. Proceedings of the National Academy of Sciences, 106(52):22239–22244.
- Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). Blast+: architecture and applications. BMC bioinformatics, 10(1):421.
- Choi, Y., Agarwal, S., and Deane, C. M. (2013). How long is a piece of loop? PeerJ, 1:e1.
- Choi, Y. and Deane, C. M. (2010). Fread revisited: accurate loop structure prediction using a database search algorithm. Proteins: Structure, Function, and Bioinformatics, 78(6):1431–1440.
- Chothia, C. and Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. The EMBO journal, 5(4):823.
- Chothia, C. and Levitt, M. (1976). Structural patterns in globular proteins. Nature, 261:552–558.
- Chou, P. Y. and Fasman, G. D. (1977). Secondary structural prediction of proteins from their amino acid sequence. Trends in Biochemical Sciences, 2(6):128–131.
- Clark, E. D. B. (2001). Protein refolding for industrial processes. Current Opinion in Biotechnology, 12(2):202–207.
- Clarke, T. F., Clark, P. L., et al. (2008). Rare codons cluster. PloS one, 3(10):e3412.
- Cline, M. S., Karplus, K., Lathrop, R. H., Smith, T. F., Rogers, R. G., and Haussler, D. (2002). Information-theoretic dissection of pairwise contact potentials. Proteins: Structure, Function, and Bioinformatics, 49(1):7–14.
- Cohen, B., Presnell, S., and Cohen, F. (1991). Pattern-based approaches to protein structure prediction. Methods in Enzymology, 202:252–268.
- Cohen, F., Arbanel, R., Kuntz, I., and Fletterick, R. J. (1986). Turn prediction in proteins using a pattern-matching approach. Biochemistry, 25(1):266–275.
- Cost, S. and Salzberg, S. (1990). Exemplar-based learning to predict protein folding. In Symp. Comp. Appl. Medical Care, Washington, DC.
- Dayhoff, M. O. and Schwartz, R. M. (1978). A model of evolutionary change in proteins. In In Atlas of protein sequence and structure. Citeseer.
- de Oliveira, S. H., Shi, J., and Deane, C. M. (2015). Building a better fragment library for de novo protein structure prediction. PloS one, 10(4):e0123998.
- Deane, C. M., Dong, M., Huard, F. P., Lance, B. K., and Wood, G. R. (2007). Cotranslational protein folding—fact or fiction? Bioinformatics, 23(13):i142–i148.
- Debès, C., Wang, M., Caetano-Anollés, G., and Gräter, F. (2013). Evolutionary optimization of protein folding. PloS Computational Biology, 9(1):e1002861.

- Deleage, G. and Roux, B. (1987). An algorithm for protein secondary structure prediction based on class prediction. Protein Engineering, 1(4):289–294.
- Desiraju, G. R. and Steiner, T. (2001). The weak hydrogen bond: in structural chemistry and biology, volume 9. Oxford university press.
- Dill, K. A. (1990). Dominant forces in protein folding. Biochemistry, 29(31):7133–7155.
- Dill, K. A. and Chan, H. S. (1997). From levinthal to pathways to funnels. Nature structural biology, 4(1):10–19.
- Dor, O. and Zhou, Y. (2007). Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. Proteins: Structure, Function, and Bioinformatics, 66(4):838–845.
- Dudek, M. J. and Scheraga, H. A. (1990). Protein structure prediction using a combination of sequence homology and global energy minimization i. global energy minimization of surface loops. Journal of Computational Chemistry, 11(1):121–151.
- Dunbrack, R. L. and Karplus, M. (1993). Backbone-dependent rotamer library for proteins application to side-chain prediction. Journal of molecular biology, 230(2):543–574.
- 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, 24(3):333–340.
- Duttler, S., Pechmann, S., and Frydman, J. (2013). Principles of cotranslational ubiquitination and quality control at the ribosome. Molecular cell, 50(3):379–393.
- Eddy, S. R. (2011). Accelerated profile hmm searches. PloS Computational Biology, 7(10):e1002195.
- Eiberle, M. and Jungbauer, A. (2010). Technical refolding of proteins: do we have freedom to operate? Biotechnology journal, 5(6):547.
- Eichmann, C., Preissler, S., Riek, R., and Deuerling, E. (2010). Cotranslational structure acquisition of nascent polypeptides monitored by nmr spectroscopy. Proceedings of the National Academy of Sciences, 107(20):9111–9116.
- 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.
- Elcock, A. H. (2006). Molecular simulations of cotranslational protein folding: fragment stabilities, folding cooperativity, and trapping in the ribosome. PloS one.
- Ellis, J. J., Huard, F. P., Deane, C. M., Srivastava, S., and Wood, G. R. (2010a). Directionality in protein fold prediction. BMC bioinformatics, 11(1):172.
- Ellis, J. J., Huard, F. P., Deane, C. M., Srivastava, S., and Wood, G. R. (2010b). Directionality in protein fold prediction. BMC bioinformatics, 11(1):172.

- Ellis, J. J., Huard, F. P., Deane, C. M., Srivastava, S., and Wood, G. R. (2010c). Directionality in protein fold prediction. *BMC bioinformatics*, 11(1):172.
- Faraggi, E., Yang, Y., Zhang, S., and Zhou, Y. (2009). Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure*, 17(11):1515–1527.
- 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):259–267.
- Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001a). Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, 14(11):835–843.
- Fariselli, P., Olmea, O., Valencia, A., and Casadio, R. (2001b). Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins: Structure, Function, and Bioinformatics*, 45(S5):157–162.
- Fedorov, A. N. and Baldwin, T. O. (1997). Cotranslational protein folding. *Journal of Biological Chemistry*, 272(52):32715–32718.
- Fedorov, A. N. and Baldwin, T. O. (1999). Process of biosynthetic protein folding determines the rapid formation of native structure. *Journal of molecular biology*, 294(2):579–586.
- Fox, N. K., Brenner, S. E., and Chandonia, J.-M. (2014). Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.
- Garnier, J., Osguthorpe, D., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, 120(1):97–120.
- Geourjon, C. and Deleage, G. (1995). Sopma: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments. *Computer applications in the biosciences: CABIOS*, 11(6):681–684.
- Gibrat, J.-F., Garnier, J., and Robson, B. (1987). Further developments of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *Journal of Molecular Biology*, 198(3):425–443.
- Giglione, C., Fieulaine, S., and Meinnel, T. (2009). Cotranslational processing mechanisms: towards a dynamic 3d model. *Trends in biochemical sciences*, 34(8):417–426.
- Gobel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function and Bioinformatics*, 18(4):309–317.

- Goldkorn, T. and Schejter, A. (1976). The redox potential of cytochrome c-552 from euglena gracilis: a thermodynamic study. Archives of biochemistry and biophysics, 177(1):39–45.
- Govindarajan, S. and Goldstein, R. A. (1998). On the thermodynamic hypothesis of protein folding. Proceedings of the National Academy of Sciences, 95(10):5545–5549.
- Griep, S. and Hobohm, U. (2009). Pdbselect 1992–2009 and pdbfilter-select. Nucleic acids research, page gkp786.
- Gront, D., Kulp, D. W., Vernon, R. M., Strauss, C. E., and Baker, D. (2011). Generalized fragment picking in rosetta: design, protocols and applications. PloS one, 6(8):e23294.
- Haas, J., Roth, S., Arnold, K., Kiefer, F., Schmidt, T., Bordoli, L., and Schwede, T. (2013). The protein model portal—a comprehensive resource for protein structure and model information. Database, 2013:bat031.
- Han, Y., David, A., Liu, B., Magadán, J. G., Bennink, J. R., Yewdell, J. W., and Qian, S.-B. (2012). Monitoring cotranslational protein folding in mammalian cells at codon resolution. Proceedings of the National Academy of Sciences, 109(31):12467–12472.
- Handl, J., Knowles, J., Vernon, R., Baker, D., and Lovell, S. C. (2012). The dual role of fragments in fragment-assembly methods for de novo protein structure prediction. Proteins: Structure, Function, and Bioinformatics, 80(2):490–504.
- Harder, T., Boomsma, W., Paluszewski, M., Frellsen, J., Johansson, K. E., and Hamelryck, T. (2010). Beyond rotamers: a generative, probabilistic model of side chains in proteins. BMC bioinformatics, 11(1):306.
- Hartl, F. U. and Hayer-Hartl, M. (2009). Converging concepts of protein folding in vitro and in vivo. Nature structural & molecular biology, 16(6):574–581.
- Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Yang, Y., and Zhou, Y. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. Scientific reports, 5.
- Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences, 89(22):10915–10919.
- Hildebrand, A., Remmert, M., Biegert, A., and Söding, J. (2009). Fast and accurate automatic structure prediction with hhpred. Proteins: Structure, Function, and Bioinformatics, 77(S9):128–132.
- Hinds, D. and Levitt, M. (1992). A lattice model for protein structure prediction at low resolution. Proceedings of the National Academy of Sciences, 89(7):2536–2540.
- Hinds, D. A. and Levitt, M. (1994). Exploring conformational space with a simple lattice model for protein structure. Journal of Molecular Biology, 243(4):668–682.
- 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.

- Holley, L. H. and Karplus, M. (1989). Protein secondary structure prediction with a neural network. Proceedings of the National Academy of Sciences, 86(1):152–156.
- Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. Journal of molecular biology, 233(1):123–138.
- Holmes, J. B. and Tsai, J. (2004). Some fundamental aspects of building protein structures from fragment libraries. Protein Science, 13(6):1636–1650.
- Hong, L. and Lei, J. (2009). Scaling law for the radius of gyration of proteins and its dependence on hydrophobicity. Journal of Polymer Science Part B: Polymer Physics, 47(2):207–214.
- 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):1607–1621.
- 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.
- Hsu, S.-T. D., Fucini, P., Cabrita, L. D., Launay, H., Dobson, C. M., and Christodoulou, J. (2007). Structure and dynamics of a ribosome-bound nascent chain by nmr spectroscopy. Proceedings of the National Academy of Sciences, 104(42):16516–16521.
- Huard, F. P., Deane, C. M., and Wood, G. R. (2006). Modelling sequential protein folding under kinetic control. Bioinformatics, 22(14):e203–e210.
- Ivankov, D. N., Garbuzynskiy, S. O., Alm, E., Plaxco, K. W., Baker, D., and Finkelstein, A. V. (2003). Contact order revisited: influence of protein size on the folding rate. Protein science, 12(9):2057–2062.
- Jefferys, B. R., Kelley, L. A., and Sternberg, M. J. (2010). Protein folding requires crowd control in a simulated cell. Journal of molecular biology, 397(5):1329–1338.
- Johnson, L. S., Eddy, S. R., and Portugaly, E. (2010). Hidden markov model speed heuristic and iterative hmm search procedure. BMC bioinformatics, 11(1):431.
- Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. Journal of molecular biology, 292(2):195–202.
- Jones, D. T., Buchan, D. W., Cozzetto, D., and Pontil, M. (2012). Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics, 28(2):184–190.
- Jones, D. T. and McGuffin, L. J. (2003). Assembling novel protein folds from super-secondary structural fragments. Proteins: Structure, Function, and Bioinformatics, 53(S6):480–485.
- Jones, D. T., Singh, T., Kosciolk, T., and Tetchner, S. (2014). Metapsicov: Combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics, page btu791.

- Jones, J. E. (1924). On the determination of molecular fields. ii. from the equation of state of a gas. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 106, pages 463–477. The Royal Society.
- Joo, K., Joung, I., Young Lee, S., Yun Kim, J., Cheng, Q., Manavalan, B., Young Joung, J., Heo, S., Lee, J., Nam, M., et al. (2015). Template based protein structure modeling by global optimization in casp11. Proteins: Structure, Function, and Bioinformatics.
- Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. Journal of the American Chemical Society, 118(45):11225–11236.
- Kabsch, W. (1978). A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography, 34(5):827–828.
- Kabsch, W. and Sander, C. (1983a). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers, 22(12):2577–2637.
- Kabsch, W. and Sander, C. (1983b). How good are predictions of protein secondary structure? FEBS letters, 155(2):179–182.
- Kadokura, H. and Beckwith, J. (2009). Detecting folding intermediates of a protein as it passes through the bacterial translocation channel. Cell, 138(6):1164–1173.
- Kahn, S. D. (2011). On the future of genomic data. Science, 331(6018):728–729.
- Kaján, L., Hopf, T. A., Marks, D. S., Rost, B., et al. (2014). Freecontact: fast and free software for protein contact prediction from residue co-evolution. BMC bioinformatics, 15(1):85.
- Kalev, I. and Habeck, M. (2011). Hhfrag: Hmm-based fragment detection using hhpred. Bioinformatics, 27(22):3110–3116.
- Kamisetty, H., Ovchinnikov, S., and Baker, D. (2013). Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. Proceedings of the National Academy of Sciences, 110(39):15674–15679.
- Kim, D. E., DiMaio, F., Yu-Ruei Wang, R., Song, Y., and Baker, D. (2014). One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. Proteins: Structure, Function, and Bioinformatics, 82(S2):208–218.
- Kinch, L., Yong Shi, S., Cong, Q., Cheng, H., Liao, Y., and Grishin, N. V. (2011). Casp9 assessment of free modeling target predictions. Proteins: Structure, Function, and Bioinformatics, 79(S10):59–73.
- Klein, P. and DeLisi, C. (1986). Classifying proteins into structural groups. Biopolymers, 25:1659–1672.
- Kneller, D., Cohen, F., and Langridge, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network. Journal of molecular biology, 214(1):171–182.

- Knoops, K., Schoehn, G., and Schaffitzel, C. (2012). Cryo-electron microscopy of ribosomal complexes in cotranslational folding, targeting, and translocation. Wiley Interdisciplinary Reviews: RNA, 3(3):429–441.
- Kolb, V. (2001). Cotranslational protein folding. Molecular Biology, 35(4):584–590.
- Kolodny, R., Koehl, P., Guibas, L., and Levitt, M. (2002). Small libraries of protein fragments model native protein structures accurately. Journal of molecular biology, 323(2):297–307.
- Komar, A. A. (2009). A pause for thought along the co-translational folding pathway. Trends in biochemical sciences, 34(1):16–24.
- Komar, A. A., Kommer, A., Krasheninnikov, I. A., and Spirin, A. S. (1997). Cotranslational folding of globin. Journal of Biological Chemistry, 272(16):10646–10651.
- Kosciolek, T. and Jones, D. T. (2014). De novo structure prediction of globular proteins aided by sequence variation-derived contacts. PloS One, 9(3):e92197.
- Kretsinger, R. H. and Barry, C. D. (1975). The predicted structure of the calcium-binding component of troponin. Biochimica et Biophysica Acta (BBA)-Protein Structure, 405(1):40–52.
- Kryshtafovych, A., Barbato, A., Fidelis, K., Monastyrskyy, B., Schwede, T., and Tramontano, A. (2014). Assessment of the assessment: evaluation of the model quality estimates in casp10. Proteins: Structure, Function, and Bioinformatics, 82(S2):112–126.
- Kuntz, I., Crippen, G., Kollman, P., and Kimelman, D. (1976). Calculation of protein tertiary structure. Journal of molecular biology, 106(4):983–994.
- Larsson, P., Wallner, B., Lindahl, E., and Elofsson, A. (2008). Using multiple templates to improve quality of homology models in automated homology modeling. Protein Science, 17(6):990–1002.
- Lee, J., Kim, S.-Y., and Lee, J. (2005). Protein structure prediction based on fragment assembly and parameter optimization. Biophysical chemistry, 115(2):209–214.
- Levinthal, C. (1968). Are there pathways for protein folding. J. Chem. phys, 65(1):44–45.
- Levitt, M. (1976). A simplified representation of protein conformations for rapid simulation of protein folding. Journal of molecular biology, 104(1):59–107.
- Levitt, M. and Warshel, A. (1975). Computer simulation of protein folding. Nature, 253(5494):694–698.
- Li, S. C., Bu, D., Gao, X., Xu, J., and Li, M. (2008). Designing succinct structural alphabets. Bioinformatics, 24(13):i182–i189.
- Lim, V. and Spirin, A. (1986). Stereochemical analysis of ribosomal transpeptidation conformation of nascent peptide. Journal of molecular biology, 188(4):565–574.
- Lin, K.-F., Sun, C.-S., Huang, Y.-C., Chan, S. I., Koubek, J., Wu, T.-H., and Huang, J. J.-T. (2012). Cotranslational protein folding within the ribosome tunnel influences trigger-factor recruitment. Biophysical journal, 102(12):2818–2827.

- Liu, Y. and Bahar, I. (2012). Sequence evolution correlates with structural dynamics. Molecular biology and evolution, 29(9):2253–2263.
- Lobanov, M. Y., Bogatyreva, N., and Galzitskaya, O. (2008a). Radius of gyration as an indicator of protein structure compactness. Molecular Biology, 42(4):623–628.
- Lobanov, M. Y., Bogatyreva, N., and Galzitskaya, O. (2008b). Radius of gyration as an indicator of protein structure compactness. Molecular Biology, 42(4):623–628.
- Lu, H.-M. and Liang, J. (2008). A model study of protein nascent chain and cotranslational folding using hydrophobic-polar residues. Proteins: Structure, Function, and Bioinformatics, 70(2):442–449.
- 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.
- Marks, D. S., Hopf, T. A., and Sander, C. (2012). Protein structure prediction from sequence variation. Nature biotechnology, 30(11):1072–1080.
- McGregor, M. J., Islam, S. A., and Sternberg, M. J. (1987). Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. Journal of molecular biology, 198(2):295–310.
- Md, A., Hogan, D., Pechmann, S., Albanese, V., Brown, P., et al. (2011). Defining the specificity of cotranslationally acting chaperones by systematic analysis of mrnas associated with ribosome-nascent chain complexes. PloS Biology, 9(7):e1001100.
- Mirjalili, V., Noyes, K., and Feig, M. (2014). Physics-based protein structure refinement through multiple molecular dynamics trajectories and structure averaging. Proteins: Structure, Function, and Bioinformatics, 82(S2):196–207.
- 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, 108(49):E1293–E1301.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2014). Critical assessment of methods of protein structure prediction (casp)—round x. Proteins: Structure, Function, and Bioinformatics, 82(S2):1–6.
- Moult, J., Fidelis, K., Kryshtafovych, A., and Tramontano, A. (2011a). Critical assessment of methods of protein structure prediction (casp)—round ix. Proteins: Structure, Function, and Bioinformatics, 79(S10):1–5.
- Moult, J., Fidelis, K., Kryshtafovych, A., and Tramontano, A. (2011b). Critical assessment of methods of protein structure prediction (casp)—round ix. Proteins: Structure, Function, and Bioinformatics, 79(S10):1–5.
- Moult, J., Pedersen, J. T., Judson, R., and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. Proteins: Structure, Function, and Bioinformatics, 23(3):ii–iv.

- Murzin, A. G., Brenner, S. E., 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.
- Nakashima, H., Nishikawa, K., and Ooi, T. (1986). The folding type of a protein is relevant to the amino acid composition. Journal of Biochemistry, 99:153–162.
- 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.
- Nicola, A. V., Chen, W., and Helenius, A. (1999). Co-translational folding of an alphavirus capsid protein in the cytosol of living cells. Nature cell biology, 1(6):341–345.
- Nissen, P., Hansen, J., Ban, N., Moore, P. B., and Steitz, T. A. (2000). The structural basis of ribosome activity in peptide bond synthesis. Science, 289(5481):920–930.
- Nugent, T., Cozzetto, D., and Jones, D. T. (2014). Evaluation of predictions in the casp10 model refinement category. Proteins: Structure, Function, and Bioinformatics, 82(S2):98–111.
- O’Brien, E. P., Vendruscolo, M., and Dobson, C. M. (2012). Prediction of variable translation rate effects on cotranslational protein folding. Nature communications, 3:868.
- Oh, E., Becker, A. H., Sandikci, A., Huber, D., Chaba, R., Gloge, F., Nichols, R. J., Typas, A., Gross, C. A., Kramer, G., et al. (2011). Selective ribosome profiling reveals the cotranslational chaperone action of trigger factor in vivo. Cell, 147(6):1295–1308.
- Olmea, O. and Valencia, A. (1997). Improving contact predictions by the combination of correlated mutations and other sources of sequence information. Folding and design, 2:S25–S32.
- Onuchic, J. N., Luthey-Schulten, Z., and Wolynes, P. G. (1997). Theory of protein folding: the energy landscape perspective. Annual review of physical chemistry, 48(1):545–600.
- Overington, J., Donnelly, D., Johnson, M. S., Šali, A., and Blundell, T. L. (1992). Environment-specific amino acid substitution tables: tertiary templates and prediction of protein folds. Protein Science, 1(2):216–226.
- O’Brien, E. P., Ciryam, P., Vendruscolo, M., and Dobson, C. M. (2014a). Understanding the influence of codon translation rates on cotranslational protein folding. Accounts of chemical research, 47(5):1536–1544.
- O’Brien, E. P., Vendruscolo, M., and Dobson, C. M. (2014b). Kinetic modelling indicates that fast-translating codons can coordinate cotranslational protein folding by avoiding misfolded intermediates. Nature communications, 5.
- Pazos, F., Olmea, O., and Valencia, A. (1997). A graphical interface for correlated mutations and other protein structure prediction methods. Computer Applications in the Biosciences, 13(3):319–321.

- Pechmann, S. and Frydman, J. (2013). Evolutionary conservation of codon optimality reveals hidden signatures of cotranslational folding. Nature structural & molecular biology, 20(2):237–243.
- Pedersen, J. T. and Moult, J. (1996). Genetic algorithms for protein structure prediction. Current Opinion in Structural Biology, 6(2):227–231.
- Pedersen, S. (1984). Escherichia coli ribosomes translate in vivo with variable rate. The EMBO journal, 3(12):2895.
- Ponder, J. W. and Richards, F. M. (1987). Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. Journal of molecular biology, 193(4):775–791.
- Punta, M., Coghill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., et al. (2011). The pfam protein families database. Nucleic acids research, page gkr1065.
- Qin, S. and Zhou, H.-X. (2014). Further development of the fft-based method for atomistic modeling of protein folding and binding under crowding: Optimization of accuracy and speed. Journal of chemical theory and computation, 10(7):2824–2835.
- R., K. (1987). An inductive learning approach to the problem of predicting a protein's secondary structure from its amino acid sequence. In Progress in Machine Learning EWSL 2., pages 230–250. Sigma Press.
- Ramachandran, G., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. Journal of molecular biology, 7(1):95–99.
- Raman, S., Vernon, R., Thompson, J., Tyka, M., Sadreyev, R., Pei, J., Kim, D., Kellogg, E., DiMaio, F., and Lange, O. (2009). Structure prediction for casp8 with all-atom refinement using rosetta. Proteins: Structure, Function, and Bioinformatics, 77(S9):89–99.
- 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):173–175.
- Ripphausen, P., Nisius, B., Peltason, L., and Bajorath, J. (2010). Quo vadis, virtual screening? a comprehensive survey of prospective applications. Journal of medicinal chemistry, 53(24):8461–8467.
- Rossmann, M. G. and Argos, P. (1976). Exploring structural homology of proteins. Journal of molecular biology, 105(1):75–95.
- Rost, B. (1999). Twilight zone of protein sequence alignments. Protein engineering, 12(2):85–94.
- Rost, B. (2001). Review: protein secondary structure prediction continues to rise. Journal of structural biology, 134(2):204–218.
- ŠAli, A. and Overington, J. P. (1994). Derivation of rules for comparative protein modeling from a database of protein structure alignments. Protein Science, 3(9):1582–1596.

- Samudrala, R. and Moult, J. (1998). An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. Journal of molecular biology, 275(5):895–916.
- Samudrala, R., Xia, Y., Huang, E., and Levitt, M. (1999). Ab initio protein structure prediction using a combined hierarchical approach. Proteins: Structure, Function, and Bioinformatics, 37(S3):194–198.
- Sánchez, I. E., Morillas, M., Zobeley, E., Kiefhaber, T., and Glockshuber, R. (2004). Fast folding of the two-domain semliki forest virus capsid protein explains co-translational proteolytic activity. Journal of molecular biology, 338(1):159–167.
- Sander, I. M., Chaney, J. L., and Clark, P. L. (2014). Expanding anfinen’s principle: contributions of synonymous codon selection to rational protein design. Journal of the American Chemical Society, 136(3):858–861.
- Saunders, R., Mann, M., and Deane, C. M. (2011). Signatures of co-translational folding. Biotechnology journal, 6(6):742–751.
- Scheraga, H., Anfinsen, C., and Schechter, A. (1973). Current topics in biochemistry.
- Schneider, M. and Brock, O. (2014). Combining physicochemical and evolutionary information for protein contact prediction. PloS one.
- Schulz, G. E., Barry, C. D., Friedman, J., Chou, P. Y., Fasman, G. D., Finkelstein, A. V., Lim, V. I., Ptitsyn, O. B., Kabat, E. A., Wu, T. T., Levitt, M., Robson, B., and Nagano, K. (1974). Comparison of predicted and experimentally determined secondary structure of adenylyl kinase. Nature, 250(5462):140–142.
- Schwede, T. (2013). Protein modeling: what happened to the “protein structure gap”? Structure, 21(9):1531–1540.
- Seckler, R., Fuchs, A., King, J., and Jaenicke, R. (1989). Reconstitution of the thermostable trimeric phage p22 tailspike protein from denatured chains in vitro. Journal of Biological Chemistry, 264(20):11750–11753.
- Seemayer, S., Gruber, M., and Söding, J. (2014). Ccnpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. Bioinformatics, 30(21):3128–3130.
- Shackelford, G. and Karplus, K. (2007). Contact prediction using mutual information and neural nets. Proteins: Structure, Function, and Bioinformatics, 69(S8):159–164.
- Shapovalov, M. V. and Dunbrack, R. L. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. Structure, 19(6):844–858.
- Shen, Y., Picord, G., Guyon, F., and Tuffery, P. (2013). Detecting protein candidate fragments using a structural alphabet profile comparison approach. PloS one, 8(11):e80493.
- Shi, J., Blundell, T. L., 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.

- Shindyalov, I., Kolchanov, N., and Sander, C. (1994). Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering*, 7(3):349–358.
- Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1):539.
- Simoncini, D. and Zhang, K. Y. (2013). Efficient sampling in fragment-based protein structure prediction using an estimation of distribution algorithm. *PLoS One*, 8(7).
- Skwark, M. J., Abdel-Rehim, A., and Elofsson, A. (2013). Pconsc: combination of direct information methods and alignments improves contact prediction. *Bioinformatics*, 29(14):1815–1816.
- Skwark, M. J., Raimondi, D., Michel, M., and Elofsson, A. (2014). Improved contact predictions using the recognition of protein like contact patterns. *PloS Computational Biology*.
- Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.
- Söding, J. (2005). Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7):951–960.
- Söding, J. and Remmert, M. (2011). Protein sequence comparison and fold recognition: progress and good-practice benchmarking. *Current opinion in structural biology*, 21(3):404–411.
- Sørensen, M. A. and Pedersen, S. (1991). Absolute in vivo translation rates of individual codons in escherichia coli: The two glutamic acid codons gaa and gag are translated with a threefold difference in rate. *Journal of molecular biology*, 222(2):265–280.
- Spencer, P. S., Siller, E., Anderson, J. F., and Barral, J. M. (2012). Silent substitutions predictably alter translation elongation rates and protein folding efficiencies. *Journal of molecular biology*, 422(3):328–335.
- Srinivasan, R. and Rose, G. D. (1995). Linus: a hierarchic procedure to predict the fold of a protein. *Proteins: Structure, Function, and Bioinformatics*, 22(2):81–99.
- Stevens, T. J. and Arkin, I. T. (1999). Are membrane proteins “inside-out” proteins? *Proteins: Structure, Function, and Bioinformatics*, 36(1):135–143.
- Sun, S. (1993). Reduced representation model of protein structure prediction: statistical potential and genetic algorithms. *Protein Science*, 2(5):762–785.
- Tanaka, S. and Scheraga, H. A. (1975). Model of protein folding: inclusion of short-, medium-, and long-range interactions. *Proceedings of the National Academy of Sciences*, 72(10):3802–3806.
- Tanaka, S. and Scheraga, H. A. (1976). Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, 9(6):945–950.

- Terwilliger, T. C., Stuart, D., and Yokoyama, S. (2009). Lessons from structural genomics. Annual review of biophysics, 38:371.
- Thomas, P. J., Qu, B.-H., and Pedersen, P. L. (1995). Defective protein folding as a basis of human disease. Trends in biochemical sciences, 20(11):456–459.
- Tosatto, S. C. (2005). The victor/frst function for model quality estimation. Journal of Computational Biology, 12(10):1316–1327.
- Tsai, C.-J., Sauna, Z. E., Kimchi-Sarfaty, C., Ambudkar, S. V., Gottesman, M. M., and Nussinov, R. (2008). Synonymous mutations and ribosome stalling can lead to altered folding pathways and distinct minima. Journal of molecular biology, 383(2):281–291.
- Ulmschneider, M. B. and Sansom, M. S. (2001). Amino acid distributions in integral membrane protein structures. Biochimica et Biophysica Acta (BBA)-Biomembranes, 1512(1):1–14.
- Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Information Theory, IEEE Transactions on, 13(2):260–269.
- Wang, G. and Dunbrack, R. L. (2003). Pisces: a protein sequence culling server. Bioinformatics, 19(12):1589–1591.
- Wang, K., Fain, B., Levitt, M., and Samudrala, R. (2004). Improved protein structure selection using decoy-dependent discriminatory functions. BMC Structural Biology, 4(1):8.
- Wang, Z., Eickholt, J., and Cheng, J. (2010). Multicom: a multi-level combination approach to protein structure prediction and its assessments in casp8. Bioinformatics, 26(7):882–888.
- Xu, D. and Zhang, Y. (2012a). Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. Proteins: Structure, Function, and Bioinformatics, 80(7):1715–1735.
- Xu, D. and Zhang, Y. (2012b). Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. Proteins: Structure, Function, and Bioinformatics, 80(7):1715–1735.
- Xu, J. and Zhang, Y. (2010). How significant is a protein structure similarity with tm-score= 0.5? Bioinformatics, 26(7):889–895.
- Yang, J.-R., Chen, X., and Zhang, J. (2014). Codon-by-codon modulation of translational speed and accuracy via mrna folding. PloS one.
- Yaseen, A. and Li, Y. (2014). Context-based features enhance protein secondary structure prediction accuracy. Journal of chemical information and modeling, 54(3):992–1002.
- Young, T. S. and Schultz, P. G. (2010). Beyond the canonical 20 amino acids: expanding the genetic lexicon. Journal of Biological Chemistry, 285(15):11039–11044.
- Zemla, A. (2003a). Lga: a method for finding 3d similarities in protein structures. Nucleic acids research, 31(13):3370–3374.

- Zemla, A. (2003b). Lga: a method for finding 3d similarities in protein structures. Nucleic acids research, 31(13):3370–3374.
- Zhang, C.-T. and Zhang, R. (1999). Skewed distribution of protein secondary structure contents over the conformational triangle. Protein engineering, 12(10):807–810.
- Zhang, G., Hubalewska, M., and Ignatova, Z. (2009). Transient ribosomal attenuation coordinates protein synthesis and co-translational folding. Nature structural & molecular biology, 16(3):274–280.
- Zhang, Y. (2008). I-tasser server for protein 3d structure prediction. BMC bioinformatics, 9(1):40.
- Zhang, Y., Arakaki, A. K., and Skolnick, J. (2005). Tasser: an automated method for the prediction of protein tertiary structures in casp6. Proteins: Structure, Function, and Bioinformatics, 61(S7):91–98.
- Zhang, Y., Ma, C., Yuan, Y., Zhu, J., Li, N., Chen, C., Wu, S., Yu, L., Lei, J., and Gao, N. (2014). Structural basis for interaction of a cotranslational chaperone with the eukaryotic ribosome. Nature structural & molecular biology, 21(12):1042–1046.
- Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. Proteins: Structure, Function, and Bioinformatics, 57(4):702–710.
- 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.
- Zou, T., Williams, N., Ozkan, S. B., and Ghosh, K. (2014). Proteome folding kinetics is limited by protein half-life. PloS One.
- Zwanzig, R., Szabo, A., and Bagchi, B. (1992). Levinthal’s paradox. Proceedings of the National Academy of Sciences, 89(1):20–22.