BCL::FOLD - DE NOVO PROTEIN STRUCTURE PREDICTION BY ASSEMBLY OF SECONDARY STRUCTURE ELEMENTS

By

Mert Karakaş

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

In partial fulfillment of the requirements for

the degree of

DOCTOR OF PHILOSOPHY

in

Chemical and Physical Biology

December, 2011

Nashville, Tennessee

Approved:

Professor Jens Meiler

Professor Albert Beth

Professor Phoebe Stewart

Professor Charles Sanders

Professor Brandt Eichman

To Gülfem, my parents and my brother

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor Dr. Jens Meiler for his support throughout my graduate career. Being one of the first members of his laboratory, I had the great chance to work on a very challenging and large scale project from the ground up. He has provided very valuable input as well as showing great confidence in me which allowed me to grow as an independent researcher.

Last six years has been a great learning experience for me. I would like to thank my colleagues, specifically Nils Woetzel and Nathan Alexander who have been invaluable in every part of my thesis project. I would also like to thank Dr. Rene Staritzbichler and Dr. Brian Weiner for their scientific contributions to BCL::Fold project. I would also like to acknowledge the members of my thesis committee Dr. Al Beth, Dr. Phoebe Stewart, Dr. Chuck Sanders and Dr. Brandt Eichman.

It has not always been easy in graduate school. Fortunately I was lucky to have great friends who provided great companionship in this treacherous journey. I would like to especially thank Andrew Morin, Yoana Dimitrova, Kazım Tuncay Tekle, Can Envarlı and Cem Albayrak.

I would like to show my greatest gratitude to my loving family; my mother Ufuk Karakaş, my father Mehmet Karakaş and my brother Murat Karakaş. I feel extremely fortunate to have such a great family. They have been the inspiration for me every day in my life to pursue my dreams and to become the person I am today. They have provided me with constant love and support all my life.

Finally, I would like to thank the love of my life, my better half and my best friend Gülfem Güler. She has always been there for me with her never-ending love, understanding and encouragement. None of this would have been possible without her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artifical Neural Network
AUC	Area Under Curve
BCL	BioChemistry Library
CASP	Critical Assesment of Protein Structure Prediction
CO	Contact order
CRYO-EM	Cryo-electron microscopy
EPR	Electron Paramagnetic Resonance
FN	False negative
FP	False positive
GA	Global Agreement
GDT	Global Distance Test
GDT_TS	Global Distance Test
HMM	Hidden Markov Model
MCM	Monte Carlo Metropolis
MP	Membrane protein
NMR	Nuclear Magnetic Resonance
PDB	Protein Data Bank
SSE	Secondary structure element
SVM	Support Vector Machine
RCO	Relative Contact order
RMSD	Root mean square deviation
ROC	Receiver Operating Characteristics
TN	True negative
TP	True positive
VDW	Van Der Waals

SUMMARY

The focus of this work was to develop a method to predict residue-residue contact in proteins and a *de novo* protein structure prediction method. The developed methods, BCL::Contact and BCL::Fold, were benchmarked on large sets of proteins and participated in Critical Assessment of Techniques for Protein Structure Prediction (CASP) experiments for validation and comparison with other methods in the field. All described work is implemented as part of BioChemistry Library, an object-oriented C++ library developed in the Meiler Lab for computational biology and cheminformatics research.

Chapter I provides an introduction which includes a brief overview of protein structure and experimental methods for protein structure determination, followed by computational protein structure and residue-residue contact prediction, biennial assessment of these methods by CASP experiments, and methods for protein-protein structure comparison. Chapter II describes BCL::Contact, the residue-residue contact prediction method utilizing Artificial Neural Networks (ANNs) and how contact prediction can be used to improve computational protein structure prediction. Chapter III focuses on knowledge-based energy potentials which are developed for scoring secondary structure elements in proteins and are used in conjunction with BCL::Fold, a novel *de novo* protein structure prediction algorithm. Chapter IV focuses on the minimization framework, as well as the moves utilized in BCL::Fold and provides an extensive benchmark of the method.

Chapter II is a reproduction of a first author paper "BCL::Contact – low confidence fold recognition hits boost protein contact prediction and *de novo* structure determination" published in 2010 in Journal of Computational Biology[1]. Chapter III and Chapter IV are reproductions of co-first authored manuscripts titled "BCL::Score – Knowledge-based energy potentials for proteins with idealized secondary structure representation for *de novo* protein structure prediction" and "*De novo* prediction of complex and large protein topologies by assembly of secondary structure elements" respectively. Both of these manuscripts are submitted to PLoS Computational Biology and are result of collaborative work with Nils Woetzel.

The protein structure prediction framework described in Chapter IV serves as the basis for BCL::EM-Fold[2], a method for utilizing cryo-EM density maps for protein structure prediction, as well as several other methods for which publications are currently under progress. These other methods include but are not limited to protein structure prediction for membrane proteins, multimeric proteins, integration of NMR, EPR restraints and loop building.

CHAPTER I

INTRODUCTION

Protein Structure

Proteins are macromolecules responsible for diverse functions in biological systems. Distinct three dimensional structures that proteins adopt play crucial roles in their biological functions. Therefore, knowing the structure of a protein can reveal significant functional information.

Proteins are composed of one or more polypeptide chains, where each chain is made up of a sequence of amino acids. The length of each polypeptide chain can vary between twenty to thousands of amino acids. The sequence or "primary structure" of a protein implies a concatenation of one letter abbreviations for amino acids found in the chain. Amino acids contain an amine group, a carboxyl group and a side-chain group which varies for each type of amino acid. Twenty genetically encoded amino acid types are the building blocks of proteins.

Hydrogen bonds formed between the backbone amide hydrogen and the backbone carbonyl group lead to formation of secondary structure elements (SSEs) called α-helices and β-strands. Secondary structure of a protein refers to these stretches of secondary structure elements, whereas, tertiary structure is defined as the three-dimensional organization adopted by the packing of these SSEs. It is defined by primarily side chain interactions such as disulfide bridges, hydrogen bonds, ionic interactions, and van der

Walls interactions. Two or more polypeptide chains from the same protein or different proteins can also form interactions with each other, forming quaternary structure.

Protein Structure Determination

Experimental methods such as X-ray crystallography and nuclear magnetic resonance (NMR) can be utilized to determine the atomic coordinates, thus the tertiary structure of a protein. The experimentally determined structure files are commonly deposited in the Protein Databank (PDB)[3]. As of July 2011, the PDB contains 68,915 protein structures. These structures correspond to 17,699 unique clusters when filtered by 30% sequence identity. A large majority of these structures were determined using X-ray crystallography (60,652 structures) and NMR (7,840 structures) with the rest determined by electron-microscopy (254 structures), hybrid methods (36 structures) and other methods (133 structures). Despite these large numbers, many proteins of interest evade crystallization which is required for X-ray crystallography or are too large/unsuitable for NMR studies. More importantly, while membrane proteins account for nearly one third of current drug targets, there are only 1441 membrane protein structures deposited in the PDB.

Protein Structure Prediction

In the absence of knowledge of the atomic structure of a protein, computational methods can be utilized to predict secondary, tertiary, or quaternary structures for proteins of interest. With the increased availability of computational resources and the development

of complex and robust algorithms, these computational methods currently provide researchers an alternative to gain structural insight to protein of interest in cases where experimental methods are not applicable.

Protein structure prediction methods can be divided into two broad categories; (1) template-based methods which rely on one or more template proteins with a determined structure and a high sequence similarity to the protein of interest, and (2) *de novo* methods which do not assume the existence of such template proteins.

All protein structure prediction methods work on a given primary sequence information to generate a structural model with atomic coordinates. Depending on the method utilized, this process can take as little as few minutes. However, a single structural model does not provide high confidence structural insight. Conventionally, thousands of models are generated followed by a refinement step where only a small portion of the models are retained. The set of refinement protocols utilized depends on the method used for predicting models and can include filtering by predicted energies and clustering using structural distances.

Computational methods for protein tertiary structure prediction are evaluated biennially in Critical Assessment for Techniques for Protein Structure Prediction (CASP) experiments [4, 5]. CASP provides a blind experiment setup to evaluate a large number of protein structure prediction methods with a consistent set of parameters and therefore provides assessment of the improvements in the field in addition to highlighting the strengths and weaknesses. The CASP committee works with structural biologists and structural genomics centers to acquire target sequences for which the tertiary structures have been experimentally determined or are about to be determined. During a three

month summer prediction season, the target sequences are relayed to participating methods and tertiary structural models predicted by these methods are collected. Typically, a 2-3 week prediction time is allowed for each target and ~100 targets are released in an overlapping fashion during this 3 month period. The target proteins are categorized into two groups; template-based modeling targets (TBM) and free modeling targets (FM) based on the availability and the sequence similarity of template proteins with known structure for the given target. Participating methods are also categorized into two groups as human predictors and servers. Methods participating in the server category have to provide a webserver where target submission and retrieval of the predicted models can be automated through a webserver interface with no human interference. For server groups, the prediction time allowed is typically much shorter than for human predictor groups, and is one to two days for most targets. In the 9th round of the CASP experiment (CASP9), which was held in the summer of 2010, 139 server groups and 109 human groups participated in the tertiary structure prediction category, while 129 targets for server groups and 60 targets for human groups were released. 102 out of the 129 targets for server groups, and 46 out of the 60 targets for human groups had at least one TBM domain.

In addition to tertiary structure prediction, CASP experiments also evaluate a variety of protein structure related categories. These include; residue-residue contact prediction (RR), identification of disordered regions (DR), function prediction (FN), quality assessment (QA) as well as refinement where the participating groups compete in a second prediction round to improve the accuracy of submitted models from the first round.

Protein Structure Comparison Methods

Assessment of the quality of protein structural models generated by computational protein structure prediction methods requires methods for comparison with the native structure. The most commonly used method is root mean square deviation (RMSD). RMSD measures the average distance between corresponding atoms in the two superimposed protein structures. For *de novo* predicted models, usually RMSD is calculated for only Cα atoms, while all backbone atoms or all atoms can be used for high accuracy evaluation of homology/template-based modeling.

When benchmarking protein structure prediction methods on a large set of proteins of variable lengths, just relying on RMSD values causes issues due to the fact that a 6Å RMSD structural model for a small protein is actually easier to achieve than the same RMSD structural model for a larger protein. In order to evaluate prediction accuracies among benchmark proteins with varying lengths, a normalized RMSD value named RMSD100 is used[6]. By convention, structural models with less than 8Å RMSD are considered to have native-like topologies.

In the early years of the CASP experiment RMSD was the only means for evaluation. However, since RMSD measures the best global superimposition, it is unable to recognize good local superimposed regions in structures, as in the case of multi-domain proteins. To overcome this issue, a variety of supplemental protein structure comparison methods have been developed. MaxSub[7] and Global Distance Test (GDT) [8] are both measures that put more importance on good local structural alignments rather than a good global structural alignment. GDT is calculated by the largest set of atoms that can be superimposed below a given distance cutoff and returned as the percentage of total

number of atoms. A variant of GDT measure, GDT_TS returns the average of GDT values for 1Å, 2Å, 4Å and 8Å distance cutoffs, while GDT_HA (for high accuracy comparison), is the average of GDT values for 0.5Å, 1Å, 2Å and 4Å distance cutoffs.

Template Based Protein Structure Prediction

Proteins with similar sequences are very likely to have similar tertiary structures. Template-based methods leverage this fact to predict tertiary models for given protein sequences. The process starts with identification of template proteins with determined structures and high sequence similarities (typically >30%) to the protein sequence of interest. The structural information of the template proteins serve as a starting point. In the absence of a single template protein with a very high sequence similarity, many methods combine structural information from multiple template proteins with lower sequence similarities.

The successes of the template-based methods rely on the existence of such templates. Cozzetto et al[9] provides a good overview of template based methods that participated in CASP8 along with a detailed analysis of accuracies of the predicted models. Zhang server[10] was the top ranking in the server category, while a large group of methods were considered to be top performing for the human category, including Rosetta, Zhang, Zhang-Server and TASSER[10-12].

De novo Protein Structure Prediction

De novo methods do not rely on the existence of template proteins for protein structure prediction. In theory, this makes *de novo* methods applicable to a larger set of proteins. Unlike template based methods where the template protein structure is used as the starting point, *de novo* methods have to assemble the structure from sequence information only, usually starting with an extended chain conformation. As expected, this tends to be a much more difficult task. Therefore, compared to template-based methods, the expected accuracies of *de novo* methods for proteins with high sequence similarity templates are lower. Another drawback arising from the increased complexity that comes with *de novo* methods is the significant increase in the structure conformations that need to be sampled. In order to overcome this increase in conformational search space, most *de novo* methods rely on a simplified energy function where side chain atoms are usually missing or represented with the "centroid" atoms that replace Cβ atom and represent the properties of the side chain. These kinds of simplifications in protein representation are also reflected in the energy functions used and are aimed to first make the energy landscape smoother and more importantly to allow faster calculation of energies. In addition to energy functions, most *de novo* methods also try to employ a more reductionist approach in their sampling strategies, by applying larger changes such as larger and more frequent phi/psi angle alterations in each step.

De novo protein structure prediction typically starts with predicting secondary structure [13-16] and other properties of a given sequence such as β-hairpins [17], disorder [18, 19], non-local contacts [20], domain boundaries [21-23], and domain interactions [24, 25]. System-learning approaches most commonly used in this field include artificial

neural networks (ANN), hidden Markov models (HMM), and support vector machines (SVM) [26, 27].

This preparatory step is followed by the actual folding simulation. Rosetta, one of the best performing *de novo* methods, follows a fragment assembly approach [28-30]. For all overlapping nine- and three- amino acid peptides of the sequence of interest conformations are selected from the PDB by agreement in sequence and predicted secondary structure. Rosetta is capable of correctly folding about 50% of all sequences with less than 150 amino acids [31].The size limitation is due to the increase in conformational search space that needs to be sampled as the protein gets larger and in part due to the fragment assembly strategy. Replacement of fragments favors formation local-contacts, as in an ordered β-sheet, and therefore causes problems as proteins gets larger and are more likely to have complex topologies where non-local contacts are more readily observed. The simulation of the whole sequence as a connected chain prevents sampling different conformations easily since different topologies might require untangling of the loops and can easily cause knots. Although Rosetta is a *de novo* method, the generation of fragment libraries requires matching the given sequence to large database of available PDBs. Even if the fragments are relatively short in length, three- and nine- amino acids, the method expectedly performs better for proteins for which the fragment database has one or more high sequence similarity proteins.

Protein Structure Prediction using Limited Experimental Restraints

For a large set of proteins, high resolution structural information is not attainable due to technical limitations of methods used. In such cases, a variety of experimental methods can be utilized to collect limited experimental restraints. These limited data sets, although insufficient for protein structure determination individually, can still serve as structural restraints or constraints. Such experimental restraints are becoming more readily available for challenging and interesting proteins and can be crucial in gaining structural and function insight as well as determination of further investigations.

Cryo-electron microscopy (cryo-EM) is applicable to large proteins and macromolecular complexes (eg. viral capsid). The density maps obtained using cryo-EM can provide topological information with identification of α-helices starting at 9-10Å resolution, while β-strands can be identified at 4-5Å resolution. Electron paramagnetic resonance (EPR) can provide distance restraints between tagged amino acids, while mass spectrometry can be used to identify di-sulfide bonds. In addition, NMR can provide distance restraints, angle restraints and orientation restraints (residual dipolar couplings). Mass spectrometry coupled with chemical cross-linking can also provide low-resolution structural information. Lastly, small angle X-ray scattering (SAXS) and small angle neutron scattering (SANS) provide information of the shapes and sizes of proteins.

These limited structural data sets are not equally distributed throughout the proteins but instead are more readily available for backbone and SSEs as observed in many examples [2, 32-34]. This preference can be due to dynamics and labeling strategies used, since SSEs tend to be more rigid compared to flexible loop regions. The structural information in these data sets may not be sufficient enough to build a high accuracy structural model

for a given protein. However, these restraints/constraints have a great impact when combined with computational protein structure prediction methods. The restraints can significantly limit the conformational search space that needs to be sampled for protein structure prediction. By doing so, the predicted models can achieve higher accuracy than present in the experimental data as the remaining conformational search space is sampled more densely.

Over the last decade, a variety of protein structure prediction methods both template-based and *de novo* have been developed/updated to integrate these experimental restraints [35-38]. In template-based methods, these additional restraints can help to achieve a very high accuracy model. For targets where template-based modeling is not applicable due to lack of high sequence similarity template proteins, these experimental restraints can be used in conjunction with *de novo* protein structure prediction methods to sample native-like topologies more frequently. This integration can allow application of *de novo* methods to larger proteins to which *de novo* methods normally could not have been applied to with a reasonable amount of computational run time requirement and an acceptable level of expected accuracy in the predicted models.

BCL::Fold

BCL::Fold has been developed to address the current limitations of *de novo* protein structure methods in the field in applicability to larger proteins with complex topologies. The sequence assembly approaches employed by many *de novo* protein structure methods like Rosetta[28] have difficulty sampling conformations with an abundance of non-local

contacts, where parts of the sequence far away from each other in sequence order tend to be close to each other in three dimensional structure. This limitation is the direct result of simulating the folding of a protein by starting from an extended conformation. The size of the protein is another major bottleneck for *de novo* methods. Currently *de novo* methods perform well and are able to generate structural models with native-like topologies for proteins of lengths mostly below 150 residues.

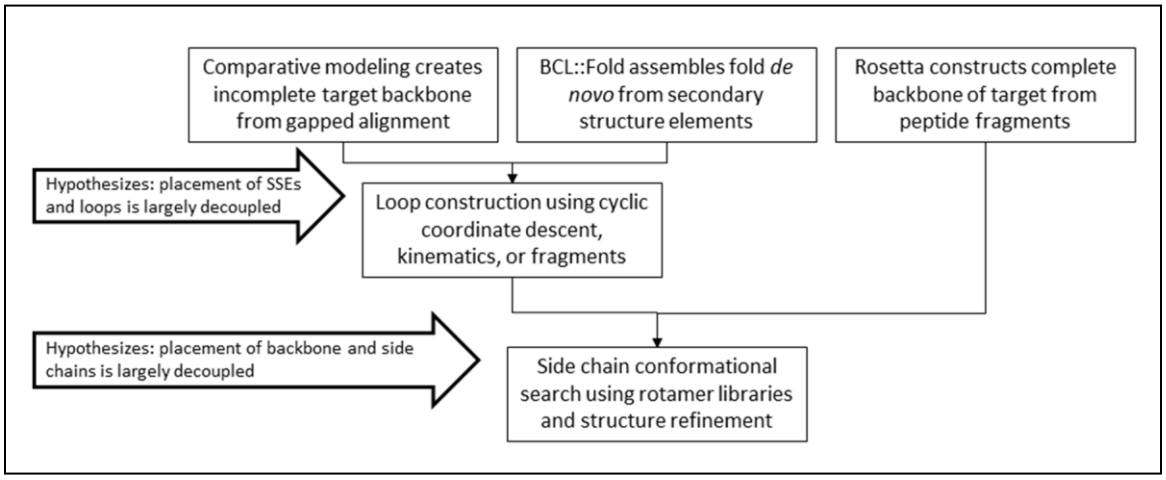

Figure 1 : Comparison of comparative modeling, BCL::Fold and Rosetta: In comparative modeling backbone is constructed partially from alignment, followed by loop construction and side chain building. On the other hand, *de novo* methods, such as Rosetta, only take advantage of the decoupling of backbone placement and the side chain building. BCL::Fold also decouples the construction of loops from assembly of secondary structure elements, similar to comparative modeling. Although these decouplings make computation more feasible by splitting the total search space into manageable portions, they are not absolute and in order to address these issues SSE placement has to be refined before loop building and backbone needs to be refined when side chains are constructed.

Traditionally construction of backbone of a protein has been separated from construction of side chain. This strategy is employed by both comparative modeling and *de novo* structure prediction methods (Figure 1). This decision is based on the assumption that overall placement of backbone is possible without explicit modeling of side chain atoms,

i.e. once the backbone is placed, the placement of side chains follow suite [39]. This approach can be further divided by decoupling of placement of SSEs and flexible loop regions as observed commonly in comparative modeling methods.

BCL::Fold builds upon this hypothesis as a novel approach where secondary structure elements (SSEs); namely α-helices and β-strands are assembled together while loops are not explicitly represented and modeled (Figure 1). The lack of loop connectivity allows a more robust sampling of different placements of SSEs and aims to overcome the size and complexity limitation. This strategy also leverages the fact that SSEs are sufficient in most cases to define the topology of a protein. Similar to other *de novo* methods, BCL::Fold also decouples the building of side chains from the placement of backbone atoms. Another positive outcome of this approach is that complex topologies with abundance of non-local contacts can be easily sampled since locations of SSEs can be readily swapped with each other as long as they are not too far from each other so that the loop can be closed after the minimization has ended. More importantly, BCL::Fold provides a simple and efficient tool to sample topologies which can be followed by any suitable choice of loop building and side chain building method in the field. This strategy also fits very well with the general protocol of *de novo* methods, where a very large number of models are generated only to be filtered down to a small percentage by score or clustering. By using BCL::Fold building of loops and side chains can be avoided until the filtering/clustering is completed and only a subset of models are left to continue with making the overall process rather efficient.

Contact Prediction

Two residues in a protein sequence are considered to be in "contact" if they are spatially close in the tertiary structure, conventionally with a C_β-C_β distance $\leq$8Å. In the absence of structural information, correct identification of all such contacting residue pairs in a given protein sequence can mediate structure determination using distance geometry methods [40, 41]. Contacts can be categorized into two groups based on the sequence separation (typically 12 amino acids) of residues that are in contact: (1) local contacts and (2) non-local contacts. Knowledge of even a small subset of non-local contacts can be extremely beneficial for protein structure prediction, since these contacts provide higher degree information than local contacts about the fold of a protein [42-44]. Therefore, contact prediction has been a topic of interest in the field of computational structural biology and can be utilized for inferring protein folding rates and pathways [45, 46] in addition to fold recognition [47, 48]. Similar to any experimental distance restraint/constraint, accurate prediction of contacts can improve the accuracy and the speed of *de novo* protein structure prediction by being used as an additional energy term and a filter for large number of generated models.

Contact prediction methods are classified into two groups [48]: (1) sequence-based and (2) structure-based. Sequence-based methods often use evolutionary correlated mutations [24, 49-55] and machine learning approaches [55-62] such as artificial neural networks (ANNs), Hidden Markov models (HMMs), or support vector machines (SVMs) to predict contacts. On the other hand, structure-based methods generally cluster best energy models generated by structure prediction techniques and pick the contacts that are observed most abundantly across the clusters [63-71]. As expected, structure-based

methods outperform sequence-based methods, especially if proteins of similar fold (templates) are available in the PDB and hence the predicted structural models are of high quality [64].

BCL::Contact

Although structure-based methods outperform sequence-based methods for contact prediction in terms of accuracy, in *de novo* protein structure prediction, applicability of structure-based methods is limited due to the absence of highly similar and complete structural templates in addition to being computationally expensive.

BCL::Contact is a bi-modal contact prediction method that employs both sequence-based and structure-based contact prediction. The sequence-based mode distinguishes itself by employing specialized ANNs for each distinct contact type and was developed for providing predictions rapidly in order to be used as input to *de novo* protein structure prediction methods. On the other hand, structure-based mode utilizes a single ANN in conjunction with models generated from fold recognition servers. This bi-modal approach allows assessment on the impact of contact prediction in protein structure prediction in cases where no sequence homologs are readily available.

BCL::Contact was designed to eventually be used in conjunction with BCL::Fold. With the integration of correlated mutations, BCL::Contact is planned to be added as an energy function and/or as an additional filter for discriminating native-like models in BCL::Fold.

BioChemistry Library

BioChemistry Library (BCL) is a scientific software library developed in the Meiler laboratory. As of 2011, BCL consists of more than 500,000+ lines of code and encompasses a variety of applications. Two main focuses of the library are computational biology tools for proteins and cheminformatics research. In addition to these, the BCL library includes methods for protein secondary structure prediction, protein sequence alignment, representation and storage of biological data, protein-protein structural comparison, energy functions for evaluation of protein structures, descriptor generation for protein sequences for development of further machine learning based methods, database access, machine learning via Artifical Neural Networks (ANN), Support Vector Machines (SVM) and a very flexible minimization/optimization framework that supports Monte Carlo-based minimizations on proteins and other targets.

Applications developed within BCL are released to the public via the BCL Commons website (http://bclcommons.vueinnovations.com/bclcommons) as well as via the Meiler lab website (http://www.meilerlab.org) as web servers. The releases for the applications are done concurrently with the publication of the corresponding methods. These applications are freely available to any academic entity, while a fee is charged for commercial uses.

CHAPTER II

BCL::CONTACT – LOW CONFIDENCE FOLD RECOGNITION HITS BOOST PROTEIN CONTACT PREDICTION AND DE NOVO STRUCTURE DETERMINATION

Introduction

The contact prediction problem is defined as the identification of all spatially close residue pairs in the tertiary structure of a given protein sequence (conventionally C_β-C_β distance ≤8Å). The motivation to solve this problem is that a complete list of all contacts defines the fold of the protein and allows structure determination using distance geometry methods [40, 41]. However, even very incomplete lists of long range contacts can facilitate protein fold prediction by reducing the number of possible topologies sometimes to a unique solution [72].

It is important to understand that not all contacts within a fold have the same value for protein structure prediction. While local contacts (contacts between amino acids nearby in sequence) are more readily predicted (e.g. within an α-helix or β-hairpin), their ability to constrain the fold space is limited. The challenge is predicting contacts between residues distant in sequence (sequence separations larger than 12 amino acids). Knowing only a few of these contacts frequently allows the fold of a protein to be defined completely [42-44].

Therefore, contact prediction methods have the potential to improve the speed and the accuracy of *de novo* protein structure prediction methods in two ways [44]: they can be used to enrich for good models in large ensembles of structural models or they can directly be used to guide *de novo* folding simulations. Furthermore, contact prediction is useful for fold recognition [47, 48] and inferring protein folding rates and pathways [45, 46].

Contact prediction methods can be classified into two groups [48]: (1) sequence-based and (2) structure-based. Sequence-based methods often use evolutionary correlated mutations [24, 49-55] and machine learning approaches [55-62] such as artificial neural networks (ANNs), Hidden Markov models (HMMs), or support vector machines (SVMs) to predict contacts.

A powerful concept in sequence-based contact prediction is use of evolutionary correlated mutations[49, 73, 74]. From multiple sequence alignments, residue pairs are identified that are mutated concurrently between sequences in the alignment throughout evolution. Often spatially close residues are mutated to complement the initial mutation and maintain the protein's structure and/or function [49]. Therefore, identification of such residue pairs yields potential residue-residue contacts. Halperin et. al [53] reviews use of correlated mutations for predicting inter-protein and intra-protein contacts and concludes correlated mutations by themselves can predict contacts with up to 20% accuracy [53]. In comparison, SAM_T06 by Shackelford and Karplus [68], implements a hybrid approach where information from correlated mutations along with various additional descriptors are used to train ANNs for predicting contacts with accuracies ranging up to ~60% for certain difficult targets while averaging ~25% for long distance contacts [44].

PROFCON [61], which ranked as one of the top groups in CASP6 also uses ANNs with descriptors including evolutionary profiles and secondary structure prediction. SVMCON uses similar descriptors with SVMs instead of ANNs, and is reported to achieve 27.7% accuracy for >= 12 residue sequence separation contacts [62]. A recent report by Wu and Zhang[64] introduces SVM-SEQ, a sequence-based contact predictor, and SVM-LOMETS, a structure template-based predictor based on previously reported LOMETS [69] meta-threading server which uses predictions from 9 different threading algorithms. In their analysis of predictions for an independent data set, accuracy of SVM-LOMETS is 39% and accuracy of SVM-SEQ is 23%. However when only new fold targets in CASP7 are considered, SVM-SEQ outperforms SVM_LOMETS and reaches an accuracy slightly better than of SAM_T06.

On the other hand, structure-based methods generally cluster best energy models generated by structure prediction techniques and pick the contacts that are observed most abundantly across the clusters [63-71]. PROSPECTOR_3.5 [70] implements a template-based approach where it collects the contacts found in the tertiary models produced by TASSER_2.0 [70] and picks the ones that are commonly observed across tertiary models. SVM-LOMETS [64], as described before, uses a similar approach but instead depends on LOMETS meta-server. As expected and as reported[64], structure-based methods outperform sequence-based methods, especially if proteins of similar fold (templates) are available in the PDB and hence the predicted structural models are of high quality [64]. However, in *de novo* protein structure prediction, applicability of structure-based methods is limited due to the absence of highly similar and complete structural templates.

Further, the computational intensity of protein structure prediction prior to contact prediction requires significant time and resources.

BCL::Contact introduces a novel hybrid approach where the sequence-based mode only relies on sequence information and utilizes individual ANNs for each distinct contact type. The structure-based mode combines results from various fold recognition servers using a single ANN. Here we present evaluations and comparisons of both modes of BCL::Contact on predicting contacts. In particular, the value of fold recognition for contact prediction in the hard fold recognition and new fold categories are evaluated. The object of this work is to evaluate if consensus fold recognition results improve contact prediction even if no sequence homologs were unambiguously detected by the underlying fold recognition methods. Further, the impact of contact prediction on *de novo* tertiary structure determination is measured by testing the ability of predicted contacts to a) enrich for native-like models in a set of decoys or b) directly guide protein folding simulations using the Rosetta *de novo* protein folding algorithm [30].

Methods

Contact Definitions and Contact Types

We use a C_β-C_β distance of 8Å or less as a threshold for defining two amino acids as being in contact. A minimum sequence separation of 12 residues is required to exclusively focus on non-local contacts. Furthermore, the sequence-based mode uses five distinct contact types between secondary structure elements in the order as they appear in the protein sequence: helix-helix, helix-strand, strand-helix, strand-strand and sheet-sheet.

This distinction was introduced to test the ability of the ANN to specialize for specific types of interactions between secondary structure elements. It is limited to the sequence-based mode due to the limited amount of training and test data available for the structure-based methods.

Protein Data Sets and Training Procedures

For the sequence-based mode, a non-redundant (20% sequence similarity) 1834 protein subset of the Protein Data Bank (PDB) was selected using the PISCES server [75] . 10% of the structures were selected as an independent dataset and removed prior to training the ANNs. With the remaining 90%, 10 ANNs for each of the five contact types were trained in a cross-validation setup using a different non-overlapping 10% of the data as a monitoring data set.

For the structure-based mode, 545 proteins which served as targets during LIVEBENCH7, LIVEBENCH8 and LIVEBENCH9 experiments [76] were used as the training dataset. 12% of these proteins (66) were withheld for independent testing. Independent ANNs were trained in a ten-fold cross-validation setup with non-overlapping monitoring data sets.

For both modes, sequence-based and structure-based, the average output from the ten ANNs is reported as the prediction result. All ANNs were trained in a "balanced" fashion with 50% contacts and 50% non-contacts by under-sampling the non-contacts. In sequence-based mode, the 50% non-contacts were a mixture of "true non-contacts" and "wrong-contacts" (contacts between other types of secondary structure elements). The

large ratio of non-contacts to contacts would otherwise bias the ANN towards predicting non-contacts.

Numerical Representation

In the sequence-based mode of BCL::Contact, for every residue pair (*i*,*j*), two sequence windows centered around these residues are used to generate input. The length of the window is chosen as five amino acids (two neighbors on each side of the amino acid of interest) for β-strands and nine amino acids (four neighbors on each side of the amino acid of interest) for α-helices. Both windows cover approximately 12 Å or two periods of the secondary structure element type.

Input to the ANNs (Figure 2) start with three position descriptors (1) number of residues N-terminal to *i*, (2) number of residues between *i* and *j*, (3) number of residues C-terminal to *j*. These global descriptors are followed by following descriptors for each amino acid in the two windows; JUFO three-state secondary structure prediction (www.meilerlab.org, three numbers per amino acid) [15], amino acid property profiles (seven numbers per amino acid; sterical parameter, polarizability, volume, hydrophobicity, isoelectric point, helix probability and strand probability) [77], as well as position specific scoring matrices from PSIBLAST 20 numbers per amino acid) [78]. Hence, the five ANNs had a variable number of inputs determined by the associated window lengths; helix-helix 543, helix-strand and strand-helix 423, strand-strand and sheet-sheet 303. All five ANNs had 16 hidden neurons, and one output neuron with an output range of [0, 1] with 0 being "non-contact" and 1 being "contact." A consensus

output is obtained from these five ANNs by weighing their prediction with the secondary structure predictions of both residues *i* and *j* as follows:

$$p(i\ and\ j\ in\ contact) =$$

$$\begin{aligned}&(H(i)\ x\ H(j)\ x\ HelixHelix(i,j\ ))\big(H(i)\ x\ S(j)\ x\ HelixStrand(i,j)\big)\\&+\ \big(S(i)\ x\ H(j)x\ StrandHelix(i,j)\big)\\&+\ (S(i)\ x\ S(j)\ x\ (StrandStrand(i,j)+\ SheetSheet(i,j)/2))\end{aligned} \quad \textbf{(1)}$$

Where *H(x)* is the secondary structure prediction α-helix probability of residue *x* and *S(x) is the* secondary structure prediction β-strand probability of residue *x*, *HelixHelix(x,y)*, *HelixStrand(x,y)*, *StrandHelix(x,y)*, *StrandStrand(x,y)* and *SheetSheet(x,y)* are predicted probabilities of contact for the residue pair (*x,y*) from each individualized ANN.

In the structure-based mode, the fold recognition results of 32 servers [79-98] that participated in the LIVEBENCH7, LIVEBENCH8 and LIVEBENCH9 experiments [76] were used as input (Table 1). The predictions were downloaded for 545 target proteins from the metaserver homepage (www.bioinfo.pl) [86]. The initial design of this method included only 24 servers, but no significant reduction in accuracy was observed. Nonetheless, reduction of number of servers used below a critical number or selective removal of the best fold-recognition servers is expected to have a negative effect on the accuracy of the method.

Table 1: List of tertiary structure prediction servers used by structure-based mode.

3ds3	fugsa	Pcomb	Robetta
3ds5	FUGU3	Pcons2	Sam-T99
bas_b	GenTHREADER	Pcons3	Sam-T02
bas_c	INBGU	Pcons4	SFST
Blast	Mbam	PDB-Blast	SHGU
FFAS03	mGenTHREADER	Pmodel3	SPARKS
FOLDFIT	orfBC	Pmodel4	Supfampp
FORTE1	ORFeus	PROSPECT2	Wurst

The input to ANN for the structure-based mode utilizes information from the models provided by these 32 servers in addition to similar sequence descriptors used by the sequence-based mode. A global agreement (*GA*) of the server predictions is calculated for each given target sequence as the fraction of contacts jointly predicted by all servers over the number of all predicted contacts. For every residue *i* and *j* , the input to the ANN consists of six global descriptors, (1) number of residues N-terminal to *i*, (2) number of residues between *i* and *j*, (3) number of residues C-terminal to *j*, (4) number of valid models from servers where coordinates for *i* and *j* were defined (*NS*), (5) number of such models in which *i* and *j* were found to be in contact (*NC*), and (6) the global agreement value *GA* for this given sequence. These global descriptors are followed by JUFO three-state secondary structure prediction (three values per amino acid) and amino acid property profile (seven values per amino acid) for *i* and *j*. For each of the 32 servers two values are input: (1) the inverse of the minimum distance observed between *i* and *j* in the ten models available for each server (*MD*), (2) the agreement of this server's predictions for *i* and *j* with all other servers (*AG* - if *i* and j predicted to be in contact by this server S_1, iterate over every other server S_2 that also predict *i* and *j* to be in contact and sum over the ratio of contacts S_1 and S_2 share). This process is illustrated in Figure 2. The ANN had

90 inputs, 32 hidden neurons, and one output neuron. The output range is [0, 1] with 0 being "non-contact" and 1 being "contact."

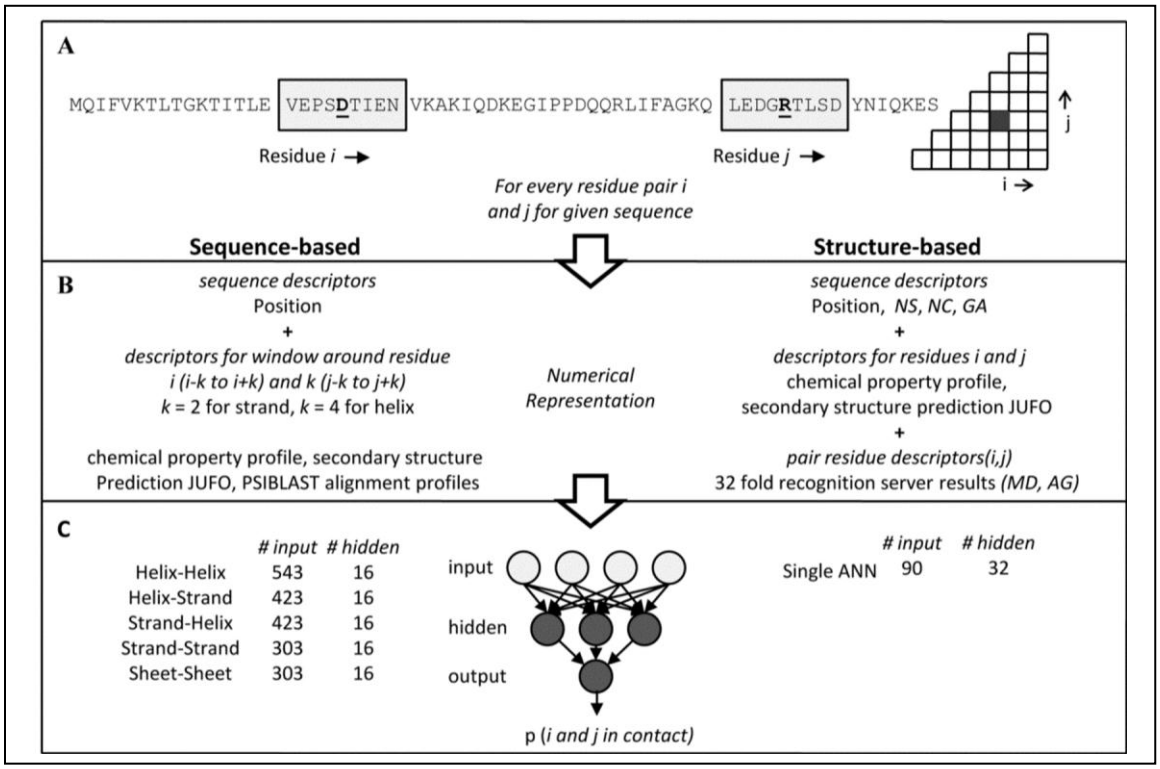

Figure 2: Scheme for sequence-based and structure-based ANN contact prediction: (A) For a given sequence, contact predictions are calculated for every residue pair i and j. Sequence windows around positions i and j are taken into account in the sequence-based mode. **(B)** The numerical representation for both methods consists of sequence descriptors, single residue, and pair residue descriptors. The sequence descriptors include number of residues N-terminal to i, number of residues between i and j, and number of residues C-terminal to j. The sequence-based mode uses sequence windows centered on residues i and j of length 5 residues (2 neighbors on each side) for β-strands or 9 residues (4 neighbors on each side) for α-helices. **(C)** The numerical representations are fed to ANNs. The structure-based mode reports the output of the single ANN while for sequence-based mode, the outputs from the five specialized ANNs for individual contact types is obtained using equation (1).

ANN Training and ROC Curve Analysis

The training algorithm was back-propagation of errors. The ANNs were trained until the root mean square deviation of the monitoring dataset was minimized (approximately 10,000 training periods). Training takes about 24h on a single typical PC processor.

The predictions from both methods were analyzed using receiver operating characteristics (ROC) curves. For all ROC curves, Area Under Curve (AUC) values are reported to quantify the improvement over a random predictor.

All methods for training, analysis, and contact prediction are implemented in the BioChemistry Library (BCL), an in-house developed C++ programming library.

Rosetta Model Building Guided by BCL::Contact

Improving accuracy of protein structure prediction is one the most important aims behind development of contact prediction methods. Thus, in order to further analyze the performance of BCL::Contact, contact predictions from BCL::Contact have been used as additional input to the protein structure prediction program Rosetta[30].

Rosetta was modified to include an additional contact prediction score. Disregarding predictions below a certain threshold, Rosetta assigns bonuses in the energy function during the folding process for structures in which residue pairs predicted to be in contact are found within 8Å (C_β-C_β distance). Variations on the threshold were systematically tested on the benchmark set of proteins and 0.2 was found to give optimum performance.

A subset of 17 structures was selected from all targets released in LIVEBENCH7, LIVEBENCH8, CASP5, and CASP6. The selection was based on having a size of less

than ~150 residues (limitations of Rosetta for *de novo* folding) [31] and being a hard fold recognition or *de novo* target without a known template (3D Jury J score lower than 50 http://bionfo.pl [86]). The rationale for choosing hard fold recognition targets was to realistically test the impact of such low confidence fold recognition results on *de novo* protein structure determination. The resultant subset was formed of the following structures; 1hjz, 1j1t, 1j26, 1l3p, 1lxj, 1mzb, 1nek, 1oh1, 1ojg, 1owx, 1oz9, 1p0z, 1p57, 1roc, 1sou, 1uan, 1v32. None of these structures was used in training any of the ANNs used by BCL::Contact.

For all 17 proteins, 10,000 structural models were generated using Rosetta's unaltered *de novo* folding protocol. The runs were then repeated for each protein with contact predictions from the sequence-based mode and with contact predictions from the structure-based mode as additional inputs.

Enrichment of native-like de-novo models

To test the ability of predicted contacts to select for native-like models and discriminate incorrect fold topologies, enrichment values were computed among the 10,000 models generated with Rosetta's unmodified *de novo* folding protocol. The enrichment values of low-RMSD *de novo* models are calculated as follows:

$$E = \frac{m}{0.01 * n} \quad \textbf{(2)}$$

where n is the total number of models (~10,000) and m is number of models in the top 10% by root mean square deviation (RMSD) that can also be found in the top 10% by

the newly implemented Rosetta sequence-based and structure based contact scores, respectively.

RMSD and MAXN% Distributions of de-novo Models

The Rosetta models generated with and without the use of contact prediction as input were compared by their distributions of RMSD and MAXN% (percentage of residues that can be superimposed to the native within 4Å) [99] for all models generated for 17 benchmark proteins. Both of these values are computed within Rosetta.

The top, 10^{th} percentile, and average values for RMSD and MAXN% are reported in Table 1 and Table 2 for all 17 proteins. For cases where improvements are observed, *p*-values are calculated from one-tailed t-tests to assess the statistical significance of improvements. In addition, the distributions are presented in histogram plots in Figure 5.

Results and Discussion

The Sequence-based mode correctly predicts 42% of native contacts with a 7% false positive rate while structure-based mode correctly predicts 45% of native contacts with a 2% false positive rate.

The sequence-based mode was tested with 183 proteins excluded from the training sets (10%). ROC curves for the average outputs for each contact type specific ANN and merged values (as described in Figure 2) are shown in Figure 3a along with the Area Under Curve (AUC) values. The helix-helix ANN achieves an AUC of 0.796. Helix-strand and strand-helix ANNs have AUC values of 0.834 and 0.831, respectively. Sheet-sheet contacts (0.784) and strand-strand contacts (0.789) are hardest for our method to predict correctly, because, in contrast to all other classes of contacts, distinguishing these contact types is not possible by predicted secondary structure. The consensus prediction method has an AUC value of 0.835. The significant deviations from the random predictor (the diagonal) for all ANNs indicate that the sequence-based mode is able to identify a substantial fraction of the non-local contacts correctly. With merged predictions and a threshold of 0.4, the sequence-based mode was able to correctly predict 42% of native contacts while identifying falsely 7% of non-contacts as contacts (Table 2).

The structure-based mode has been benchmarked with 66 LIVEBENCH [76] targets excluded from training. Figure 3b shows ROC curves displaying the average predictions for the independent dataset along with predictions from the sequence-based mode for the same dataset and corresponding AUC values. The structure-based mode (0.860) outperforms sequence-based mode (0.795) for these targets. The inset shows a clear

differentiation between sequence and structure-based modes in the region corresponding to higher predictions.

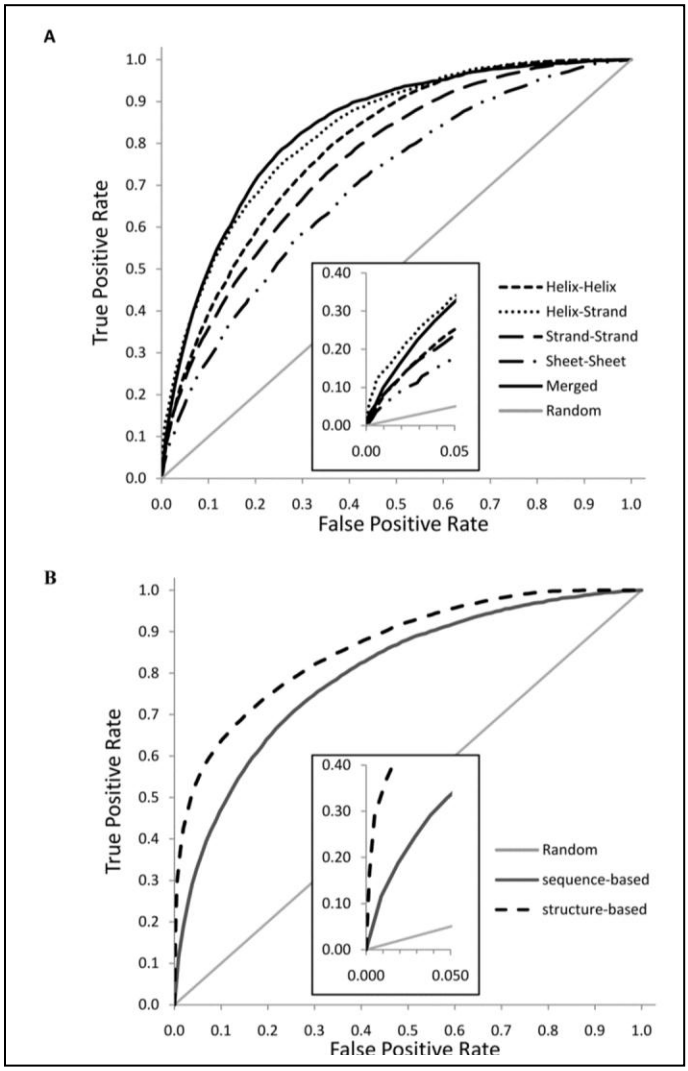

Figure 3: Receiver Operator Characteristics (ROC) curves for sequence-based and structure-based modes: (A) The ROC curves for sequence-based mode using the independent data set of 184 proteins are plotted. Individual curves are presented for all 5 ANNs specialized for individual contact types, the merged predictions, and the random predictor (diagonal). The helix-strand and strand-helix are represented with only one curve since they are virtually identical. The inset provides a magnification for the high confidence prediction region. AUC (Area under curve) values for these curves are 0.796 (helix-helix), 0.834 (helix-strand), 0.831 (strand-helix), 0.789 (strand-strand), 0.784 (sheet-sheet), and 0.835 (merged). **(B)** Plot shows ROC curve (same as (A)) for the structure-based mode benchmark on 66 LIVEBENCH targets excluded from the training and monitoring data sets. In addition, curves for the sequence-based mode for the same

66 targets and the random predictor are provided. The insert provides a magnification for the high confidence prediction region. The AUC values for these curves are 0.860 (structure-based) and 0.795 (sequence-based).

Table 2: BCL::Contact True Positive Rates (TPR) and False Positive Rates (FPR)

Threshold	Sequence-based		Structure-based	
	TPR	FPR	TPR	FPR
0.1	90	43	99	82
0.2	77	24	97	67
0.3	59	14	91	48
0.4	42	7	80	28
0.5	26	3	67	12
0.6	13	1	56	5
0.7	5	0	45	2
0.8	1	0	34	0
0.9	0	0	13	0

True Positive Rates (TPR) and False Positive Rates (FPR) for both sequence-based mode and structure-based modes of BCL::Contact for varying thresholds are reported. The standard deviation for contact predictions is smaller than 0.03 in both, sequence-based and structure-based mode.

When predictions above a threshold of 0.7 are identified as contacts, 45% of native contacts and 2% of non-contacts in the independent data set are predicted to be contacts (Table 2). Since only 5-8% of residue pairs in proteins are found to be in contact, absolute numbers for false and true positives are roughly equal at this cutoff.

In order to facilitate comparison of BCL::Contact with other methods, accuracy of the highest L, L/2 and L/5 predictions were calculated for each protein in the independent data set where L is the length of the protein of interest (Table 3). The sequence-based mode achieved accuracies of 12.2%, 15.4% and 20.9%, while the structure-based mode achieved accuracies of 67.4%, 72.7% and 77.0% when the highest L, L/2 and L/5 predictions are considered.

Table 3: Percentage of true positives in highest L, L/2 and L/5 predictions

pdb id	Sequence-based			Structure-based		
	L	L/2	L/5	L	L/2	L/5
1T5YA	7.3	6.3	0.0	89.6	91.7	100.0
1N6UA	11.2	15.0	19.1	86.5	87.9	88.4
1TC5A	18.3	28.6	52.6	94.4	100.0	100.0
1VJGA	28.1	40.0	72.7	93.2	93.6	97.7
1OE4A	10.4	13.6	16.0	95.2	100.0	100.0
1R75A	11.2	19.7	28.6	35.0	47.9	53.6
1O70A	13.4	17.1	18.8	40.7	49.4	61.5
1SF8A	3.1	0.0	0.0	24.2	25.0	20.0
1HL6B	11.9	13.3	26.7	69.5	81.3	96.7
1MW5A	9.5	8.4	0.0	17.9	21.1	21.1
1UX6A	9.0	7.9	0.0	37.8	42.9	40.9
1VJUA	9.0	11.5	0.0	14.1	14.1	9.7
1QYDA	14.5	25.3	38.7	93.4	94.9	100.0
1NU7D	7.0	5.6	7.1	13.3	17.5	28.1
1OMSA	19.5	26.2	33.3	58.5	68.9	70.8
1UFOA	18.3	33.3	45.8	96.3	94.2	95.8
1S1IF	2.4	2.4	0.0	98.2	98.8	97.0
1PG6A	13.0	11.4	12.5	15.0	11.4	2.0
1J1LA	12.2	16.3	48.3	90.8	91.8	91.4
1PSYA	12.6	15.9	33.3	81.1	90.5	96.0
1ODHA	13.6	15.9	11.8	35.0	46.6	60.0
1UOCA	9.9	8.2	6.9	93.2	95.2	100.0
1IZNA	17.2	24.8	48.3	54.8	60.7	65.5
1M1LA	5.0	8.4	26.1	42.7	47.9	44.7
1NNWA	16.4	14.1	0.0	97.3	100.0	100.0
1MZKA	12.8	14.3	14.3	97.9	100.0	100.0
1EI6A	8.3	16.0	39.0	88.6	95.6	97.6
1O5HA	5.5	5.6	9.5	25.8	29.6	34.9
1NW1A	11.5	15.7	32.6	87.6	92.6	98.9
1J03A	17.3	23.1	60.0	97.1	96.2	100.0
1S18A	8.9	10.7	30.3	49.1	58.3	83.6
1SGOA	24.1	40.0	64.3	51.8	58.6	75.0
1QV9A	13.9	19.6	50.0	85.0	98.6	100.0
1NQDA	16.1	19.4	41.7	50.8	56.5	62.5
1VKHA	26.0	42.0	66.7	78.0	79.7	76.4
1O7DE	14.1	15.6	16.7	76.6	84.4	96.0
1R71A	8.8	10.0	0.0	50.3	67.8	94.4
1MJDA	36.5	42.1	54.6	75.7	87.7	95.7
1RU8A	5.1	1.7	0.0	83.0	86.3	85.1
1P97A	8.6	6.9	0.0	94.8	94.8	91.3
1RQPA	17.2	29.1	33.3	61.1	77.5	91.7
1UMHA	4.3	4.3	11.1	51.3	64.5	73.0
1N05A	13.9	12.1	0.0	33.7	41.0	51.5
1UW7A	12.4	13.9	0.0	38.6	52.8	62.1
1NLXA	5.3	0.0	0.0	50.4	53.6	59.1
1JOPA	15.3	20.5	26.7	63.7	68.0	67.7
1L9KA	11.7	13.0	13.3	92.2	97.4	100.0
1UETA	9.0	13.6	13.6	82.8	81.0	80.7
1IQ6A	7.4	5.9	0.0	99.3	100.0	100.0
1Q8RA	8.2	9.8	16.7	65.6	80.3	95.8
1MZBA	15.9	14.5	0.0	93.5	98.6	100.0
1P42A	11.0	11.7	22.2	44.2	58.4	66.7
1NYCA	3.5	0.0	0.0	51.3	60.7	68.2
1O0PA	11.3	18.9	40.0	99.1	98.1	100.0
1USUB	0.0	0.0	0.0	56.7	68.6	76.5

pdb id	Sequence-based			Structure-based		
	L	L/2	L/5	L	L/2	L/5
1TE5A	6.9	9.2	7.7	74.0	65.4	61.5
1RI6A	15.5	13.8	20.6	99.1	100.0	100.0
1OW1A	7.1	12.1	10.5	27.8	32.3	33.3
1R61A	9.5	9.5	0.0	71.4	79.1	83.3
1Q5QA	24.3	36.6	46.2	98.1	97.7	100.0
1N8NA	17.7	18.7	9.5	89.8	94.4	95.4
1OEDB	0.0	0.0	0.0	16.9	20.5	24.0
1V5PA	9.4	9.4	0.0	99.2	100.0	100.0
1GVNB	18.6	22.1	31.0	68.4	81.4	89.7
1NKVA	19.2	27.7	26.9	91.5	93.9	92.3
average	*12.2*	*15.4*	*20.9*	*67.4*	*72.7*	*77.0*

Percentage of true positives in highest L (the length of the protein of interest), L/2 and L/5 predictions for 66 independent proteins excluded from training. The averages are reported in the last row.

The Structure-based mode has been ranked as one of the three best methods in CASP6.

The Structure-based mode has participated in CASP6. The analysis done by Graña et al. [20], has placed the method as one the top three groups (out of 16 groups) in terms of accuracy and coverage. In analysis of 11 new fold targets (sequences with no structural homologues), the method achieved a mean accuracy of 16% and a mean coverage of 8%. BCL::Contact was also one of the few methods that predicted the non-local sheet topology for target 273 (PDB code 1WDJ) correctly (shown in Fig. 1f in Graña et al. [20]). Figure 4b shows the tertiary structure and the contact map with the predictions from the structure-based mode for protein 1V5P. The contact map indicates a significant overlay of native and predicted contacts in particular for within β-sheet topology while the non-local contacts within the β-sheet are correctly identified.

The Sequence-based mode predicted long distance contacts in CASP7 with up to 40% accuracy.

The sequence-based mode of BCL::Contact has participated in CASP7 in the contact prediction category along with 16 other methods. The results were analyzed in detail by Izarzaguza et al. [44] based on predictions for 19 selected targets composed of 15 free modeling targets and 4 template based modeling targets. Predictions were submitted for BCL::Contact for 18 targets out of these 19. In 14 of them, our predictions met the criteria of having at least L/5 (length of given sequence divided by 5) number of predictions for long distance contacts (>24 residue sequence separation). Due to the lack of a large subset of common targets for which most groups have submitted predictions, no clear ranking of all groups was obtained [44].

The sequence-based method achieved an average accuracy of 4.6% and an average coverage of 2.4% for long distance contacts over 14 targets included in this analysis. However, in 50% of these targets none of the L/5 long-range contact predictions were correct. Our method achieved its best ranking (4^{th} out of 10) for target T0356_3 (PDB code 2IDB chain C) out of this set of targets, with an accuracy of 20.8% and coverage of 14.8%. When L/10 instead of L/5 highest confidence predictions are considered for this target, the accuracy reaches 34%. When all targets (including the template-based modeling targets) are considered, our method predicted most accurately for target T0345_1 (PDB code 2HE3) which is not a free modeling target. For this target, our method achieved 40.5% accuracy and 5.5% coverage for L/5 highest predictions, while these values rise up to 61.1% and 26.7% respectively when L/10 highest predictions are considered. Figure 4a illustrates the structure of 2HE3 with residues corresponding to L/5

highest predictions highlighted in purple and the contact map that shows predictions submitted for this target. The accurate prediction of non-local contacts within the β-sheet is remarkable.

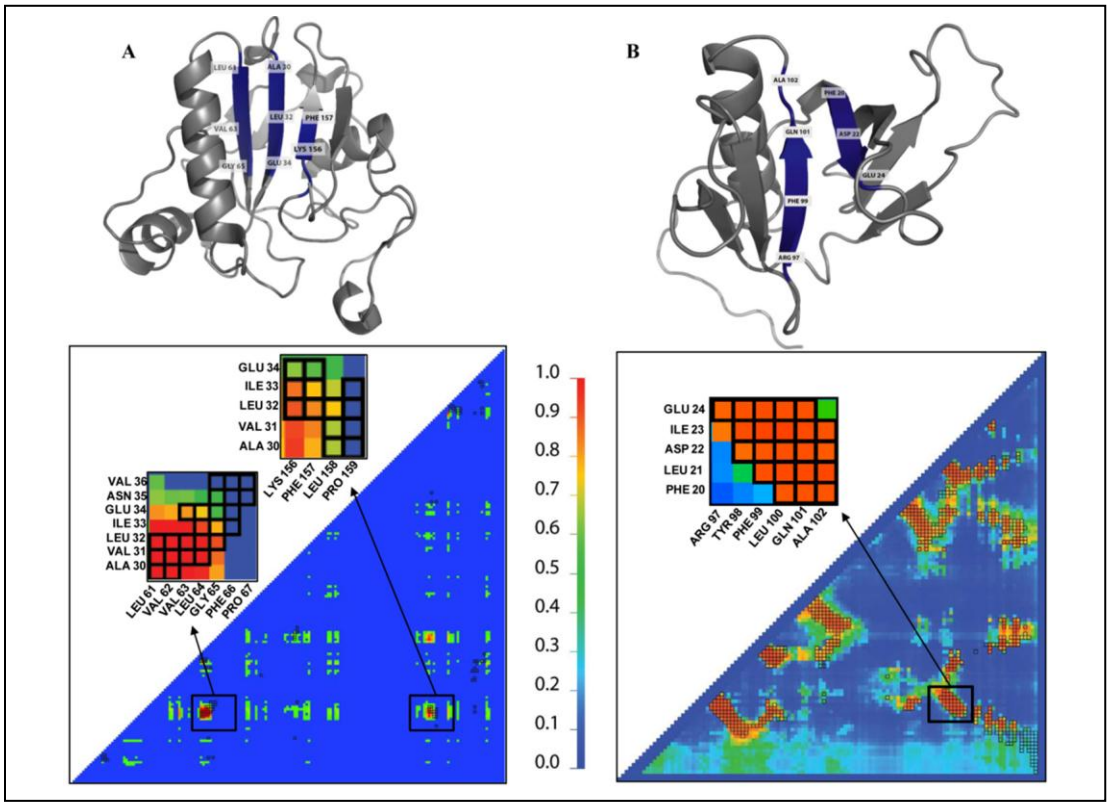

Figure 4: BCL::Contact predictions mapped on tertiary structures and shown as contact maps: The contact maps are colored according to the scale shown from blue (contact probability of 0.0) to red (contact probability of 1.0) **(A)** Tertiary structure for CASP7 target T0345_1 (pdb code 2HE3) with residues corresponding to L/5 highest confidence BCL::Contact predictions in sequence-based mode highlighted in dark blue and the contact map corresponding to the predictions submitted in CASP7 for the same protein**.** The highlighted residues in the structure correspond to strand pairings between *LEU61-PRO67* to *ALA30-GLU34* and *ALA30-GLUE34* to *LYS156-ILE159*. The magnified insets on the contact map correspond to these strand pairings. **(B)** The tertiary structure and the contact map with the predictions from the structure-based mode for target with LIVEBENCH id of 25864 (PDB CODE 1V5P). The high confidence predictions (red color) overlay with most of the native contacts (black boxes). The predictions lead to a true positive rate of 87% and false positive rate of 6%. The highlighted residues in the structure correspond to the strand pairing between *PHE20-GLU24* and *ARG97-ALA102*. The magnified inset on the contact map corresponds to this strand pairing and indicates a nearly perfect identification of these crucial non-local contacts.

Table 4: RMSD (Å) distributions for Rosetta folding runs for all 17 benchmark targets

pdb id	RMSD (Å)										
	no-contact			Sequence-based				Structure-based			
	top	*10%*	*avg*	*top*	*10%*	*avg*	*p-value*	*top*	*10%*	*avg*	*p-value*
1hjz	10.0	15.6	18.0	9.9	15.4	18.0	5.24E-02	9.5	13.9	16.7	1.13E-139
1j1t	14.9	19.4	21.4	14.0	19.3	21.3	1.18E-06	15.3	19.2	21.0	1.22E-32
1j26	9.6	15.7	18.1	9.1	15.3	17.9	2.74E-19	8.4	13.2	16.7	3.10E-121
1l3p	4.3	9.8	12.7	3.8	9.9	12.9	N/I	3.6	5.1	7.6	0.00E+00
1lxj	7.5	11.7	14.3	8.1	11.6	14.2	1.70E-02	8.2	12.8	15.2	N/I
1mzb	6.5	11.8	14.3	5.8	11.4	14.0	5.54E-29	5.4	9.0	12.5	1.39E-182
1nek	6.9	10.2	13.4	6.6	10.5	13.6	N/I	6.5	9.0	11.3	0.00E+00
1oh1	7.9	12.3	14.4	7.9	12.2	14.2	2.19E-10	5.6	10.1	13.2	4.06E-117
1ojg	6.2	11.6	14.5	6.5	11.7	14.4	1.02E-07	5.7	10.8	13.8	2.17E-35
1owx	12.7	16.5	18.2	12.0	16.3	18.0	1.01E-17	9.2	14.6	16.9	2.74E-190
1oz9	7.8	13.8	16.5	7.5	13.8	16.4	4.76E-01	6.3	11.8	15.0	1.11E-136
1p0z	4.5	10.7	13.9	5.3	10.2	13.7	7.66E-15	4.3	7.0	12.1	4.13E-104
1p57	10.8	14.2	15.7	11.0	14.0	15.7	9.69E-05	10.5	13.3	15.1	1.72E-70
1roc	13.3	16.6	19.1	10.0	16.5	19.0	5.34E-05	12.0	16.0	18.6	1.02E-30
1sou	11.2	16.6	19.1	10.4	16.7	19.0	1.30E-02	10.5	15.6	18.4	7.62E-45
1uan	15.3	19.3	21.4	15.6	19.1	21.3	1.45E-06	14.6	18.2	20.5	4.80E-93
1v32	7.5	11.5	13.5	7.6	11.3	13.4	2.81E-06	6.5	9.2	11.6	5.34E-284
avg	*9.2*	*14.0*	*16.4*	*8.9*	*13.8*	*16.3*	-	*8.4*	*12.3*	*15.1*	-

RMSD (Å) distributions for Rosetta folding runs for all 17 benchmark targets with no additional input and with input from sequence-based and structure-based modes of BCL::Contact. The top model, 10^{th} percentile, and average RMSD values are reported. For improved cases p-values from a one-tailed t-test are also reported. Runs for which the p-value did not improve are labeled as N/I.

Table 5: MAXN% distributions for Rosetta folding runs for all 17 benchmark targets

pdb id	MAXN%										
	No-contact			Sequence-based				Structure-based			
	top	*10%*	*avg*	*top*	*10%*	*avg*	*p-value*	*top*	*10%*	*avg*	*p-value*
1hjz	58.3	30.7	24.5	57.8	30.7	24.5	not improved	59.4	39.1	28.7	1.22E-133
1j1t	27.5	16.7	13.9	28.8	17.2	14.0	6.37E-03	24.9	18.0	14.6	1.03E-33
1j26	74.1	38.4	30.8	69.6	38.4	31.5	2.39E-15	73.2	45.5	35.7	1.04E-160
1l3p	92.2	50.0	38.5	92.2	49.0	37.4	N/I	100.0	85.3	64.9	0.00E+00
1lxj	80.8	50.0	41.7	80.8	48.1	40.4	N/I	66.4	46.2	41.8	2.07E-01
1mzb	73.5	45.6	34.3	80.9	47.1	35.0	4.42E-10	82.4	64.0	49.5	0.00E+00
1nek	64.6	43.4	33.0	69.9	42.5	32.3	N/I	69.0	48.7	36.9	2.04E-86
1oh1	59.6	43.1	33.5	64.2	41.3	32.0	N/I	83.5	50.5	40.5	1.21E-264
1ojg	71.3	47.8	38.5	73.5	47.8	38.2	1.30E-03	83.1	50.7	41.4	1.43E-54
1owx	64.5	37.2	30.2	57.9	37.2	30.2	N/I	77.7	46.3	36.2	1.74E-209
1oz9	68.0	42.0	32.8	72.0	41.3	32.4	N/I	76.7	48.0	38.1	1.12E-171
1p0z	93.1	53.4	45.4	86.3	54.2	44.9	N/I	94.7	65.7	52.3	1.62E-167
1p57	48.3	29.8	24.9	56.1	30.7	25.2	1.70E-05	44.7	32.5	26.5	1.52E-54
1roc	36.1	22.6	18.9	38.1	23.2	19.2	6.67E-11	38.7	24.5	19.9	9.78E-34
1sou	44.3	29.4	23.2	48.5	28.4	22.6	N/I	43.8	31.4	24.3	2.24E-21
1uan	34.4	21.6	17.8	41.0	22.0	18.0	1.33E-10	40.5	25.6	20.3	4.35E-148
1v32	72.3	45.5	35.9	69.3	45.5	36.1	1.58E-02	84.2	67.3	50.1	0.00E+00
avg	*62.5*	*38.1*	*30.5*	*63.9*	*37.9*	*30.2*	-	*67.2*	*46.4*	*36.6*	-

MAXN% (the percentage of residues that can be superimposed to the native within 4Å) distributions for Rosetta folding runs for all 17 benchmark targets with no additional input and with inputs from sequence-based and structure-based modes of BCL::Contact. The top model, 10^{th} percentile and average MAXN% values are reported. For improved cases p-values from a one-tailed t-test are also reported. Runs for which the p-value did not improve are labeled as N/I.

BCL::Contact induces up to 5 Å shift in average RMSD distributions and up to 26% shift in average MAXN% distributions when guiding de-novo folding

For both modes, sequence-based and structure-based, shifts in RMSD and MAXN% are reported in Table 4, Table 5 and Figure 3. In RMSD plots (Figure 5a), any improvement on the accuracy of models generated would be signified as a decrease in the RMSD values of models. These shifts are observed clearly for all four targets when using the structure-based contact predictions. The sequence-based mode also leads to a decrease in

the RMSD values for 1v32, 1uan and 1j26, although not as pronounced as in the structure-based mode.

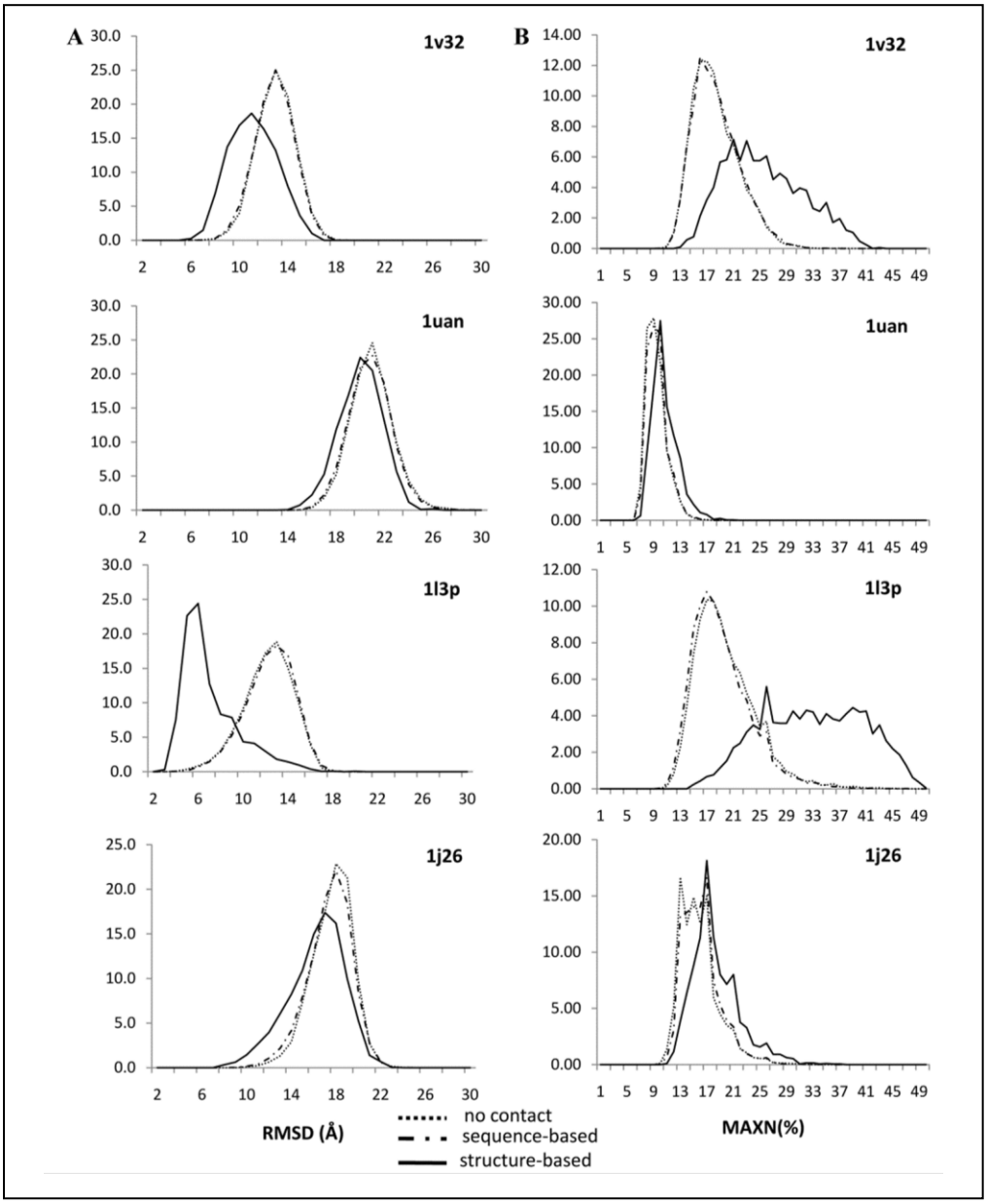

Figure 5: RMSD and MAXN% histograms for Rosetta folding runs with and without BCL::Contact prediction as input: For proteins 1v32, 1uan, 1l3p, and 1j26 the **(A)** RMSD distributions with a bin size of 1 Å and **(B)** MAXN% percentage distributions with a bin size of 4% are provided as histograms. For each protein, distributions are reported for folding with no contact prediction input, with input from sequence-based contact prediction mode, and with input from structure-based contact prediction mode.

In MAXN% plots (Figure 5b), in contrast to RMSD plots, improvement would be represented by a shift to the right when inputs from BCL::Contact are supplied to Rosetta. Similar to RMSD plots, usage of the structure-based contact predictions results in distinct shifts whereas the sequence-based mode improves Rosetta only slightly for targets 1uan and 1j26.

The sequence-based mode slightly improves the RMSD for the best model for 10 proteins, 10^{th} percentile for 13 proteins and average for 15 proteins, while structure-based mode improves the RMSD for the best model for 15 proteins, 10^{th} percentile and average for 16 proteins. A similar improvement of MAXN% values is observed for a similar number of proteins.

The structure-based mode provides an improvement of 1.3 Å in average RMSD values of all models produced, while also providing a 5.8% increase in the MAXN% distributions of the models generated. The sequence-based mode does not lead to any significant shift in the averages of both distributions. The structure-based mode performs exceptionally well for target 1l3p where it improves the RMSD of models on average by 5.1 Å (from 12.7 to 7.6) while improving the MAXN% of models by 26.4% (from 38.5% to 64.9%). With predictions from the structure-based mode Rosetta is able to produce the best model with RMSD of 3.6 Å to the native structure and MAXN% value of 100%.

In order to visualize the improvements provided by contact predictions in tertiary structure prediction, the best models by RMSD for 1l3p and 1oh1 are presented in Figure 6. For 1l3p, contacts from both sequence-based mode and structure-based mode result in a more compact packing for the helices, indicated also by the improvements in RMSD from 4.3Å to 3.8Å and 3.6Å, respectively. In particular contacts predicted between amino

acids *ALA168-PHE184*, *ALA168-PHE188*, *ALA171-PHE184*, as well as *ILE158-SER244*, and *ILE161-ILE240* help bring helices closer. In the case of protein 1oh1, sequence-based contact prediction does not result in an improvement of model accuracy. However, structure-based contact prediction results in an RMSD improvement of 2.3Å. The resultant model is the only model that has a well-defined sheet formation triggered by predicted contacts. The three highest predictions for the whole sheet region (residues 61 to 92) correspond to native contacts between amino acid pairs *LEU67-ILE77*, *GLU78-LEU89*, and *ILE78-LEU89*.

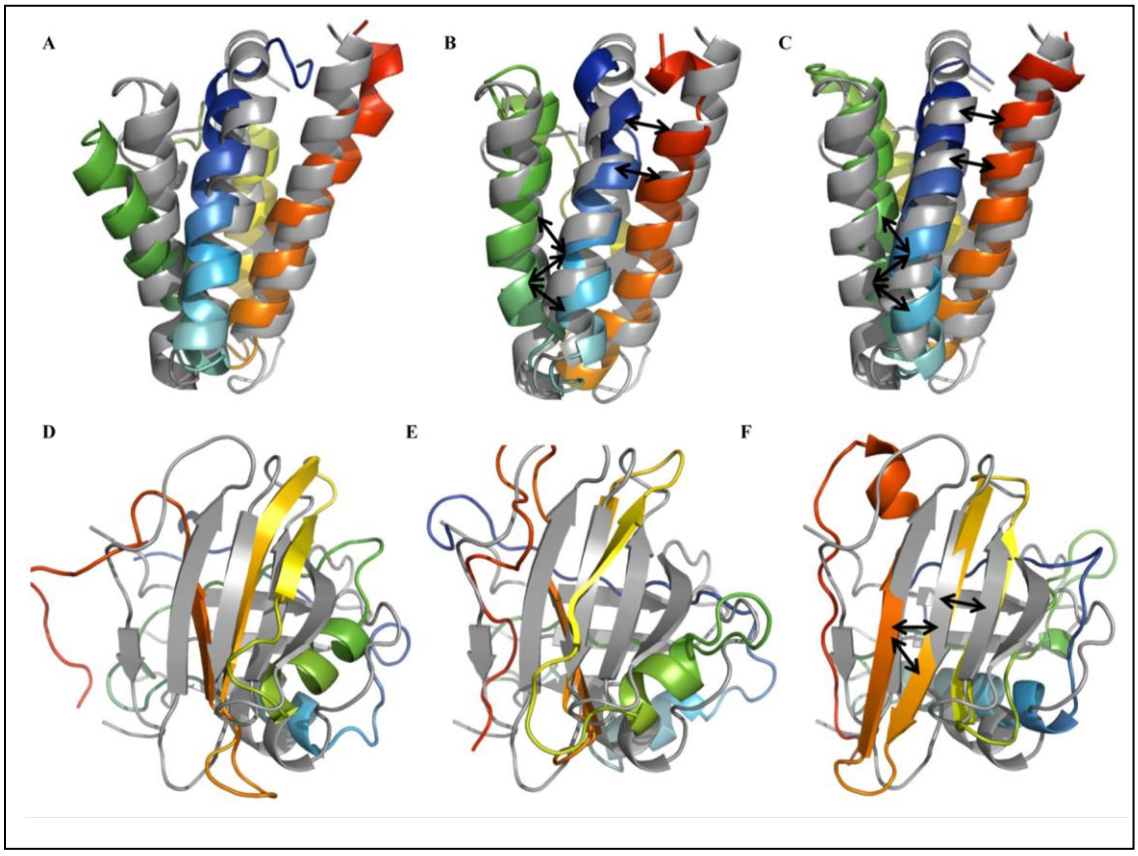

Figure 6: Comparison of best Rosetta models by RMSD in folding runs: (A) The lowest RMSD Rosetta model for protein 1l3p (rainbow coloring scheme) is shown superimposed with the native structure (gray). Panels **(B)** and **(C)** display the best models by RMSD when contacts predict by sequence-based and structure-based modes are used as score. The RMSDs of the models are 4.3Å, 3.8Å, and 3.6Å, respectively. The black

arrows in panels (B) and (C) indicate strongly predicted contacts between amino acids ALA168-PHE184, ALA168-PHE188, ALA171-PHE184 as well as ILE158-SER244, ILE161-ILE240 facilitate improved helix packing. Panels **(D-F)** show lowest RMSD models for folding protein 1oh1. The RMSDs for the models are 7.9 Å, 7.9 Å and 5.6 Å, respectively. The black arrows in panel (F) indicate strongly predicted contacts between amino acids LEU67-ILE77, GLU78-LEU89 and ILE78-LEU89 that are responsible for improved in sheet formation. The highest 50 predictions for the same region also correspond to native contacts.

BCL::Contact enriches for native-like models by factors of up to five

Another possible use of contact prediction is the discrimination of native-like models from the pool of thousands of models produced in *de-novo* protein structure prediction runs. The discriminative power of contact predictions can be measured by enrichment values (Table 6). The sequence-based mode performs poorly for targets 1l3p and 1nek, while providing slight enrichments for the rest of the cases with an average enrichment of 1.3. The structure-based mode achieves an average enrichment of 2.5, performing well for all targets except 1lxj. For example, the enrichment of 5.5 for target 1l3p maintains 548 of the best 1,000 models by RMSD when selecting the top 10% of 10,000 models by contact score, where a random scoring scheming would yield only 100 of the best 1,000 models by RMSD.

Table 6: Enrichment values for Rosetta runs for all 17 benchmark targets

pdb id	Sequence-based	Structure-based
1hjz	1.7	3.0
1j1t	1.3	1.6
1j26	1.6	2.3
1l3p	0.8	5.5
1lxj	1.1	0.2
1mzb	1.2	3.5
1nek	0.6	4.3
1oh1	1.7	2.0
1ojg	1.3	2.4
1owx	1.0	1.3
1oz9	1.0	2.3
1p0z	2.2	2.7
1p57	1.3	2.5
1roc	1.4	2.4
1sou	1.4	2.2
1uan	1.0	2.1
1v32	1.1	2.9
average	1.3	2.5

Enrichment values for Rosetta runs for all 17 benchmark targets with inputs from sequence-based mode and structure-based modes of BCL::Contact

Structure-based Contact Prediction Outperforms Sequence-based Contact Prediction even for hard Fold-Recognition Targets

In all comparisons, the structure-based mode outperforms the sequence-based mode, which is expected since it utilizes tertiary structure prediction results. This holds even for hard fold recognition targets and new folds, demonstrating that even though no template can be confidently identified, some structures found by fold recognition servers have at least partial similarity with the target structure. However, usage of the structure-based mode requires the submission of the sequence to tertiary structure prediction servers. This leads to a long processing time whereas the sequence-based mode provides contact

predictions within short processing times. However, the accuracy of sequence-based mode is limited by the lack of descriptors for evolutionary correlated mutations which has been demonstrated to be one of the most successful approaches in contact prediction methods [24, 49-55, 62]. Further, it generates many long range predictions for residue pairs that reside in different registers of interfaces in a pair of secondary structure elements. These false positives could be eliminated by a subsequent filter that limits the number of high probability predictions for each pair of secondary structure elements.

Conclusion

In this chapter, we have presented BCL::Contact, a novel contact prediction method based on ANNs. BCL::Contact competed in both CASP6 and CASP7 experiments. The structure-based mode was ranked as one of the top three groups in CASP6. The sequence-based mode was able to identify crucial long range contacts in CASP7 for some of the new fold targets. While achieving up to ~40% accuracy for such contacts, performance was not evaluated for several other targets due to the selection criteria applied prior to evaluation.

In addition to CASP experiments, both modes have been benchmarked for independent data sets. The sequence-based mode, when used with a threshold value of 0.4, was able to predict 42% of contacts correctly while identifying 7% of non-contacts falsely as contacts. The structure-based mode, when used with a threshold value of 0.7, achieved 45% accuracy in predicting contacts while falsely predicting 2% of non-contacts as contacts.

When used in protein folding simulations, the sequence-based mode provided only slight improvements in RMSD distributions of models, while with structure-based mode resulted in a significant reduction of RMSD values observed. It is expected that, with the inclusion of additional descriptors, such as correlated mutations, the sequence-based mode will also be able to provide clear improvements for tertiary structure prediction. Both methods are capable of enriching for native-like folds in a set of protein models created with the Rosetta *de n*ovo folding protocol, although the structure-based mode achieves approximately twice as high enrichment factors.

Despite the improvements in the experimental protein structure elucidation field, many proteins of interest still exist with little or no structural information available. Contact prediction methods that rely only on sequence information can be beneficial for structure prediction in such cases. Alternatively, with the emergence of new and better *de novo* tertiary structure predictions, contact prediction methods can increase their accuracy significantly by integration of models produced by such methods. BCL::Contact with both sequence-based and structure-based modes can be utilized in both of these situations. BCL::Contact is available to the scientific community at http://www.meilerlab.org/ .

CHAPTER III

BCL::SCORE - KNOWLEDGE BASED ENERGY POTENTIALS FOR PROTEINS WITH IDEALIZED SECONDARY STRUCTURE REPRESENTATION FOR DE NOVO PROTEIN STRUCTURE PREDICTION[1]

Introduction

Many protein structures have been determined using experimental techniques like x-ray crystallography [100] and NMR spectroscopy [101]. There are ~69,000 protein structures deposited in the Protein Data Bank (PDB) [102]. X-ray crystallography has the largest contribution with ~61,000 protein structures followed by ~8,000 protein structures determined by nuclear magnetic resonance (NMR) as of August 2011. Although the number of experimentally elucidated protein structures grows, challenges still exist. Membrane proteins are hard to express, crystallize and are usually too large to be studied by NMR [103]. Proteins that are only structured in the context of their biomolecular assembly like viruses or macromolecular machines can often be imaged to medium resolutions by cryo-EM [104] but crystal structures might not be obtained at high resolution in isolation for all participating proteins [105].

Protein structures which have close homologs within the scope of the PDB can be modeled to atomic resolution using comparative modeling [106]. If there are no homologs for the protein of interest in the PDB (or they cannot be detected), one has to

[1] This chapter was

apply *de novo* protein structure prediction algorithms. Rosetta is one of the most successful algorithms among the available tools for structure prediction from the amino acid sequence [30, 107]. Successes in fold determination – i.e. placement of the protein backbone to an accuracy of RMSD = 3-6Å – have been reported for proteins up to 160 amino acids [108]. More recently high-resolution structure determination to an accuracy within the experimental error of the crystal structure (all atom RMSD < 1.5Å) has been reported for proteins of up to 80 amino acids [109]. Both limits can be further extended by the inclusion of sparse experimental NMR [110] or EPR restraints [38]. Rosetta folds the continuous amino acid chain – an approach that mimics the protein folding process but is in part responsible for the size limits stated earlier as non-local sequence contacts are difficult for the algorithm to explore. A strong linear correlation between Rosetta failure and contact order – a measure of non-local sequence contacts – has been reported [42].

The Rosetta energy function is a knowledge-based potential that contains an amino acid environment term defined by burial of hydrophobic residues, an amino acid pair interaction potential defined by all amino acid pair distances and a secondary structure packing potential which uses multiple vectors to represent the conformation of a secondary structure element defining an additional dot product for β-strand-β-strand pairing to capture the hydrogen bonding between them. This potential also uses the loop length connecting two SSEs as an additional variable [107]. Rosetta represents the side chains with a "centroid" atom to approximate the average position of the side chain for an amino acid. The potential also includes a volume exclusion or van der Waals potential.

Alternative approaches to the computational protein folding problem include different flavors of molecular dynamics simulations. They are not as successful in predicting protein structures, but for small peptides e.g. a three β-stranded protein, they can provide insights into folding pathways [111] or they can help to close the gap between predicted inaccurate low resolution protein structures and high resolution crystal structures with their first principle full atom force fields [112]. The molecular mechanics energy potentials in these simulations are typically derived from first principle physical interactions (bond-length, torsion-angles, vdW interactions, coulomb-interactions, etc.). Prominent examples of such energy functions include CHARMM [113] and AMBER [114]. These force fields work with full-atom models of the protein in question, an approach that requires intensive computations for energy evaluations. The long computational time that results from the high accuracy of these potentials hinders the applicability of these methods to simulations where large conformational spaces need to be sampled rapidly, as would be required for larger proteins.

Here we introduce a comprehensive knowledge-based energy potential based on a simplified protein representation using only SSEs, i.e. α-helices and β-strands. These SSEs are sufficient to define the fold of a protein in the absence of loop regions. Although the presented energy potential is specialized for models without loop regions, it can also be used to evaluate full-chain protein models. The energy potential includes individual terms for; (1) amino acid pair distances based on C_β atom coordinates (HA2 atom for Glycine), (2) amino acid solvation, (3) a secondary structure element packing, (4) β-strand pairing, (5) loop length, (6) radius of gyration, (7) contact order, (8) backbone phi/psi angles, (9) amino acid clash, (10) SSE clash and (11) loop closure.

The rationale for such an energy function is to push the size limit of *de novo* protein structure prediction by limiting the conformational space that needs to be searched in the folding simulation to the relative arrangement of SSEs. This approach is based on the hypothesis that interactions between SSEs define the core of the protein and are major contributors to the stability of the fold and should therefore be considered first in an energy evaluation of the protein fold. Loop or coil regions, on the other hand, add large conformational spaces to the search problem due to their internal flexibility but contribute little to the stability of the fold. Therefore these regions can be omitted for the energy evaluation in the initial stage of *de novo* protein fold prediction.

Results

A database of 4379 chains and 3409 protein structures covers the space of topologies seen in the PDB

All knowledge based potentials described below have been derived from a databank that contained 4379 high resolution X-ray crystallography protein structures (R-factor 0.3 and better) with resolutions better than 2.0 Å only. A non-redundant set of proteins (<25% sequence identity) was culled using the PISCES server [115]. All membrane proteins, Cα-only entries, and structures from sources other than X-ray crystallography were excluded. The minimum chain length was determined to be 40 residues to exclude peptides that do not form structured protein domains with a well-defined core. For each of the energy potentials statistical representations of the respective geometrical features have been collected over the entire database.

The inverse Boltzman relation converts statistics into free energy functions

The collected statistical representations are converted into a free energy using:

$$E(X) = -RT \times \ln\left(\frac{p_{observed}(X)}{p_{background}(X)}\right)$$

Where $E(X)$ is the energy function for X – being the geometrical feature observed, R – the gas constant, T – temperature, $p_{observed}(X)$ – the probability with which that feature was observed and $p_{background}(X)$ the probability to observe that feature by chance. The propensity is reflected in the $p(X)/p_{background}(X)$ term. The function $E(X)$ was converted into cubic- or bicubic-splines where appropriate [116] to ensure continuous differentiability for possible use with gradient minimization.

Amino acid pair distance potential

In order to describe amino acid pair interactions, statistics for the C_β-atom distance between pairs of amino acids (X_i, X_j) have been collected. For Glycine, the HA2 hydrogen position was used (Figure 7A). Distances have been collected between 0 and 20 Å in bins of size 1 Å. Only amino acid pairs with sequence separation of at least 10 residues were considered in order to reduce the sequence bias. The background probability is derived by summing up all distance distributions of amino acid X_i's type with any other amino acid type and the distributions of X_j's amino acid type with any

other amino acid type. The raw counts and the sum of counts used for the background distribution were normalized to a sum of 1. These two distributions were divided by each other for each distance and splines were derived to yield the energy potentials (Figure 7).

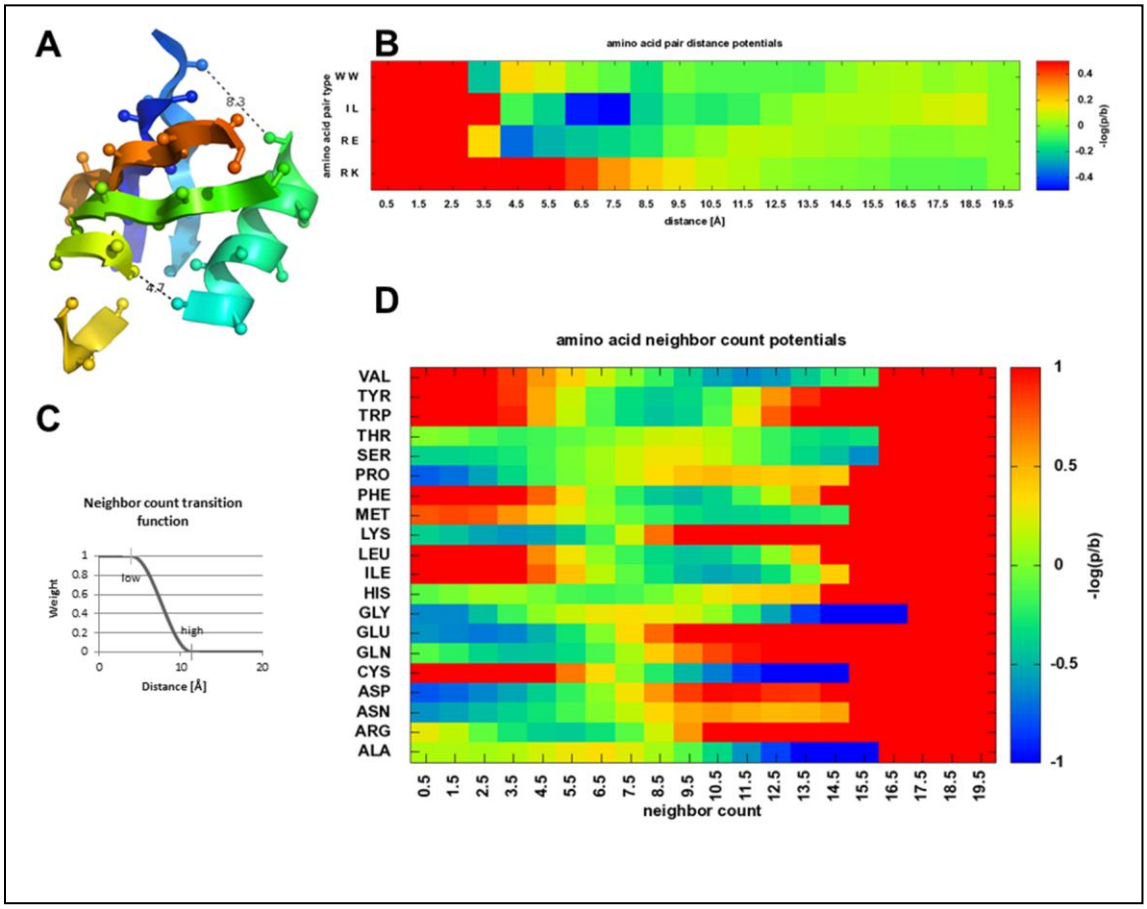

Figure 7: Amino acid pair distance and environment potential: (A) the idealized structure of 1ubi with Cβ and Hα2 atoms is shown with the distances between ILE 32 and LEU 56 (4.7 Å) and between LYS 11 and GLU 34 (8.3 Å). **(B)** selected amino acid pair distance potentials. **(C)** the transition function that is used between the lower and upper threshold in which the weight for the neighbor of considerations drops from 1 to 0 using half of a cosine function on the left. **(D)** the neighbor count energy potential for all 20 amino acids with their three letter code.

The potentials obtained follow the expected trends (Figure 7B). For example, Alanine and Isoleucine as well as Leucine are expected to interact favorably with itself due to van der Waals (vdW) attraction, which is reflected in the negative energies for short distances. Lysine and Arginine with positively charged side chains are expected to experience Coulomb repulsion when approaching each other which is reflected in the positive energy for short distances. Phenylalanine and Tryptophan may engage in π-stacking interactions, which are reflected in a preferred distance of 10 Å. Cysteine and Methionine do not have a preferred physical interaction resulting in a broad and not too low energy valley.

Rarely observed distances (bin count < 5) are considered to be errors in the experimental assignment. These energy bins are assigned an energy value of 4.

Amino acid environment potential

In order to describe the preference of an amino acid to be either exposed to solvent or buried in the protein core, a function that counts the neighbors of an amino acid was used (Figure 7C):

$$NC(aa_i) = \sum_j w\left(distance(aa_i, aa_j)\right)$$

Where $w(x)$ is:

$$w(x) = \begin{cases} x \leq threshold_{low}, 1 \\ x \epsilon \left(threshold_{low}, threshold_{high}\right), \frac{1}{2}\left(cos\left(\frac{x - threshold_{low}}{threshold_{high} - threshold_{low}} * \pi\right) + 1\right) \\ x \geq threshold_{high}, 0 \end{cases}$$

Weighing the actual neighbor count between $threshold_{low}$ and $threshold_{high}$ smoothens the potential and enables gradient based minimizations. The thresholds have been optimized for a high inverse correlation of the neighbor count value with the MSMS solvent accessible surface area (SASA) approximation implemented in the molecular visualization package VMD [117]. The lower threshold is set to 4.0 Å, the upper threshold to 11.4 Å [118]. The background probability is the sum of all normalized distributions for any amino acid type. Counts of 1 are considered errors, and set to 0 before normalizing the distributions. The resulting energy potential comprises interactions with the solvent but also with other residues in the core of the protein and encompasses Coulomb as well as van der Waals interactions (Figure 7D). Expected behavior can be observed for the potentials, e.g. Glycine with a small side chain and being nonpolar usually exhibits a high number of amino acid neighbors. Glutamate with a charged side chain has its minimum for a low neighbor count.

Loop length potential

In models without explicit representation of loop residues, it is important to guarantee that all consecutive SSEs can still be physically linked by the loop. Beyond the ability to physically link two SSEs with a fully extended loop, there are preferences for loops of a certain length to bridge a certain Euclidean distance (Figure 8A). To explore these

preferences, statistical data of the Euclidean distance between the ends of two SSEs with respect to the number of amino acids in the loop connecting those two SSEs have been collected. Using a constant background probability, the potential shown in Figure 8B was derived. For short sequence distances it is favorable that the Euclidean distance is short. Long Euclidean distances are forbidden by a positive energy. Euclidean distances below 4 Å are impossible, because they would indicate that two SSE ends would clash which cannot be observed in nature. There is a nearly linear dependency between the number of residues and the Euclidean distance represented by the long valley in the energy profile. However, as loops get longer, the range of Euclidean distance they bridge become wider and less consistent.

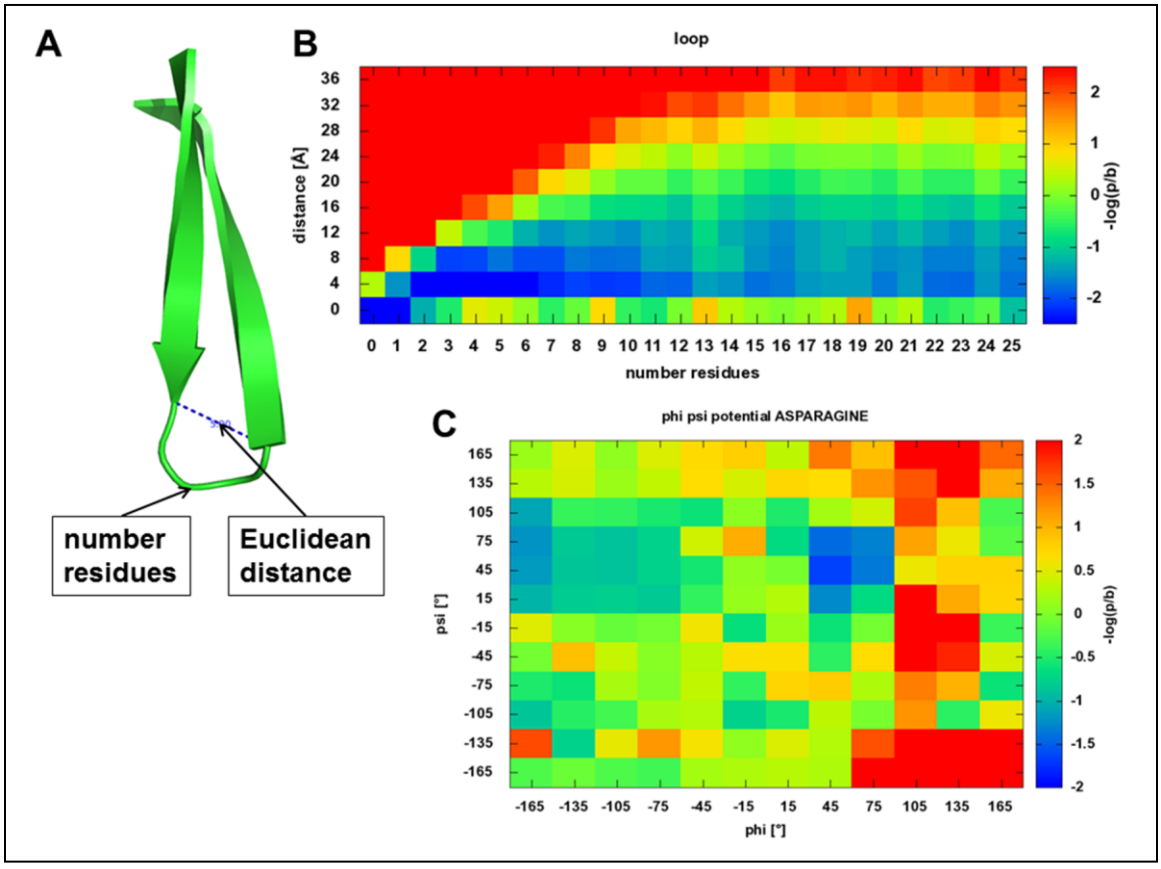

Figure 8: Loop closure potential: (A) two β-strands connected by a loop characterized

by the Euclidean distance between the two ends and the number of residues in the loop connecting those two ends. **(B)** the derived energy potential is shown, where the energy is a function of the number of residues in the loop and the Euclidean distance between the ends of the main axes. **(C)** representative phi-psi potential for Asparagine with clearly accessible and forbidden regions for the phi-psi angle combinations.

β-Strand pairing potential

This potential evaluates the pairing of β-strands to form β-sheets. It represents the likelihood of observing a twist of two β-strand fragments (Figure 9A) with respect to each other together with a distance between two β-strands. However, this potential does not check if actual hydrogen bonds can be built based on atom positions as it is purely based on the ideal fragment representation of the SSEs. The required refinement of the register shift is left for later optimizations at higher resolution. The distance between the β-strands is normalized by $p_{background}(X) \sim X$ since the chance to find a second object around an axis grows linearly with the distance of the object, similar to the girth of a circle. The resulting energy distribution depends on twist angle (0-360° with bin size 15°) and distance (0-12 Å with bin size 0.25 Å). It is interesting to see that although the resolution of the distance is only 0.25 Å, it is still possible to identify an optimal β-strand distance between 4.25 and 5 Å. Also the twist angle is represented with the deepest points around -15° and 165°, where 165° is more pronounced in its minimum showing that anti-parallel β-sheets are slightly more often represented in the database than parallel β-sheets.

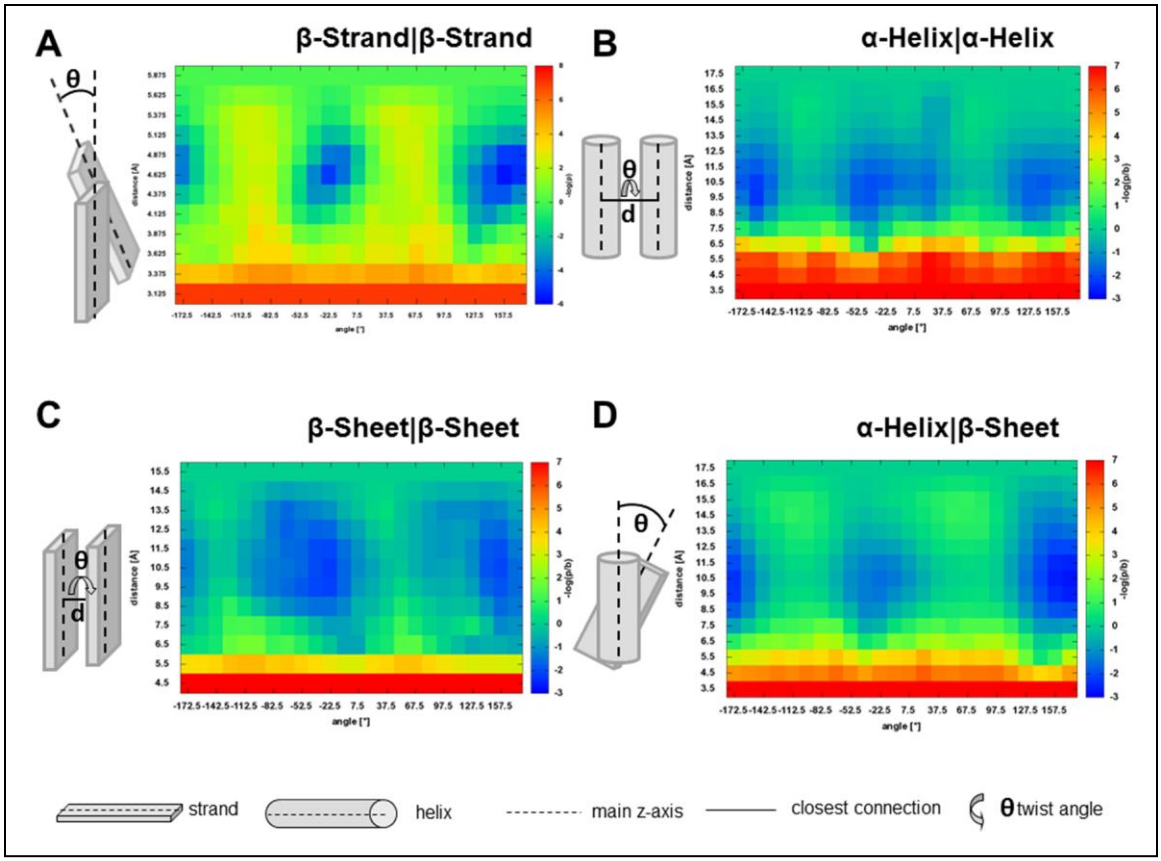

Figure 9: Strand pairing and SSE packing potential: Shown are all secondary structure element packing potentials with their schematic shortest connections, twist angle and their derived potentials. **(A)** β-Strand|β-Strand pairing potential with prominent distance of 4.75Å and angles of -15° and 165°. **(B)** α-Helix|α-Helix packing with preferred packing distance of 10Å and the preferred parallel angle of -45° and the anti-parallel packing of 135°. **(C)** β-Sheet|β-Sheet packing potential with a preferred distance 10Å and angles of -30° and 150 °. **(D)** α-Helix|β-Sheet packing with its packing distance around 10Å and an anti-parallel angle of 150°-180°.

Secondary structure element packing potential

Similar to the β-strand pairing potential, additional secondary structure element packing potentials have been derived. For α-helix-α-helix packing, the distance and the twist angle between the main axes have been analyzed (Figure 10A, C). For α-helix-β-sheet packing, the distance and the twist angle was only considered if the α-helix was contacting the side chains of the β-sheet, i.e. packs on top of the β-sheet (Figure 10D

left). For β-sheet-β-sheet packing – which differs from β-strand-β-strand pairing by relying on side chain interactions rather than backbone hydrogen bonds – contacts were considered only if the side chains of the β-sheets were facing each other (Figure 10D right). The resulting potentials are based on the distance and twist angle and reflect what is known about super secondary structure organization. Two α-helices pack in a preferred angle of -45°. The anti-parallel packing is slightly less common at around 135°. Further, weak minima around 15° and -165° are observed. Both cases of packing have a preferred distance of 10 Å (Figure 9B). For α-helix-β-sheet packing, the anti-parallel case with angles between 150° and 180° is most common as observed in the TIM-barrel fold or other "Rossman-Folds" [119] (Figure 9D). As in the α-helix-α-helix packing, the optimal distance is around 10 Å. The last case to consider is the β-sheet-β-sheet packing as seen in β-sandwiches, which is represented by two β-sheets with their side chains pointing towards each other. Two β-sheets pack in a distance around 10 Å with an equally preferred twist angle around -30° or 150° (Figure 9C). Twist angles lead in general to improved packing as the interacting side chains can reach into gaps left by the side chains of the opposite SSE [120].

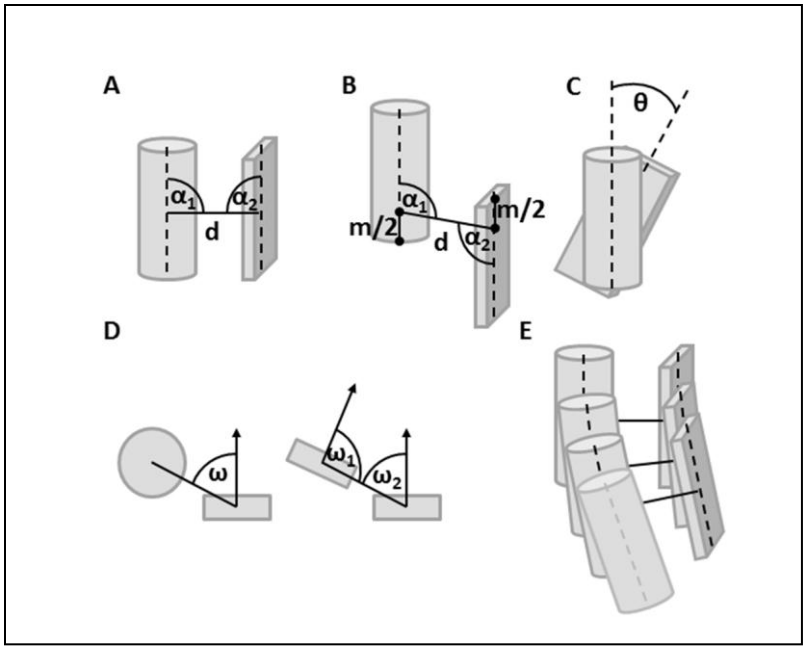

Figure 10: SSE fragments are shown with their geometric packing descriptors: (A) α1 and α2 are orthogonal, if the shortest connection between the main axes is orthogonal. **(B)** Connection is not orthogonal, since the minimal interface length m cannot be achieved. **(C)** θ is the twist angle around the shortest connection – which is equivalent to the dihedral angle between main axis 1 – shortest connection – main axis 2. **(D)** ω is the offset from the optimal expected position for a helix-strand interaction, if it is 0°, the helix is on top of the strand, if it is 90°, the helix would interact with the backbone of the strand. ω1 and ω2 are the offsets for a strand-strand packing – for omegas close to 90°, it is a strand backbone pairing interaction dominated by hydrogen bond interaction within a sheet, if they are close to 0°, it is dominated by side chain interactions like seen in sheet-sandwiches. **(E)** every SSE is represented as multiple fragments and the SSE interaction is described by the list of all fragment interactions, leaving out additional fragments of the longer SSE with suboptimal packing (bottom grey helix fragment).

Contact order score

Using the assembly of secondary structure elements to describe the topology of a protein enables the optimization protocol to explore a wide conformational space. However, especially if loop distances are not too constraining, global topologies due to overabundance of non-local contacts can be sampled. One measure for the complexity of the topology is the contact order. Contact order is defined as the average sequence

separation of all amino acids in contact, conventionally identified by a C_β -C_β distance <= 8Å. A larger contact order constitutes a more complex topology. For native proteins, the range of contact orders observed are found to be limited, meaning that not every possible complexity is explored for native proteins. This represents a natural border that generated protein models should not cross. It was found that there is a linear correlation between the contact order and the sequence length (Figure 11). A potential to evaluate the protein models contact order – number amino acid ratio was derived (Figure 12A).

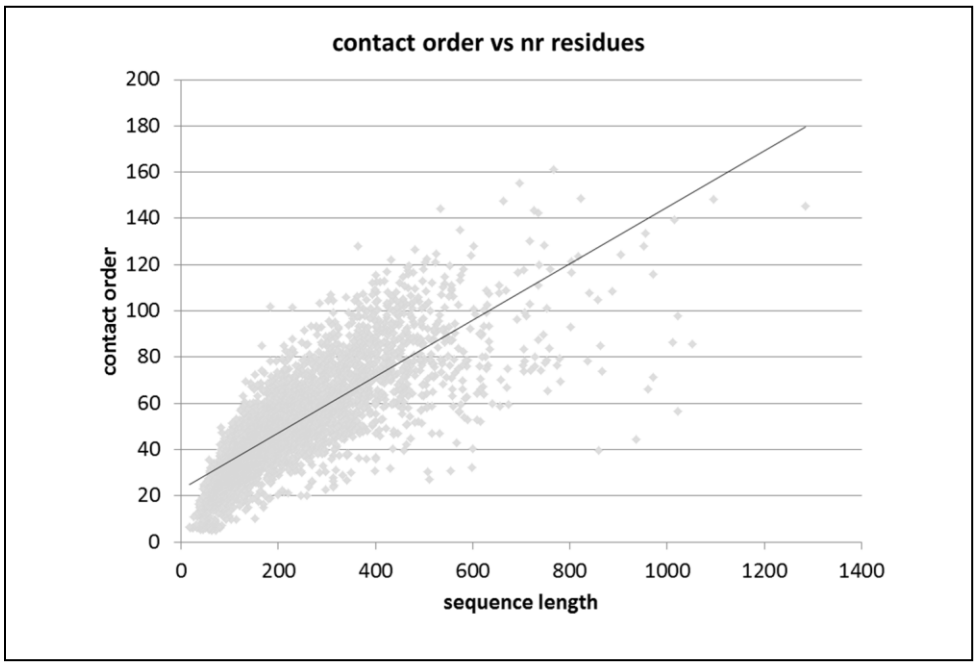

Figure 11: Contact order vs sequence length plot:

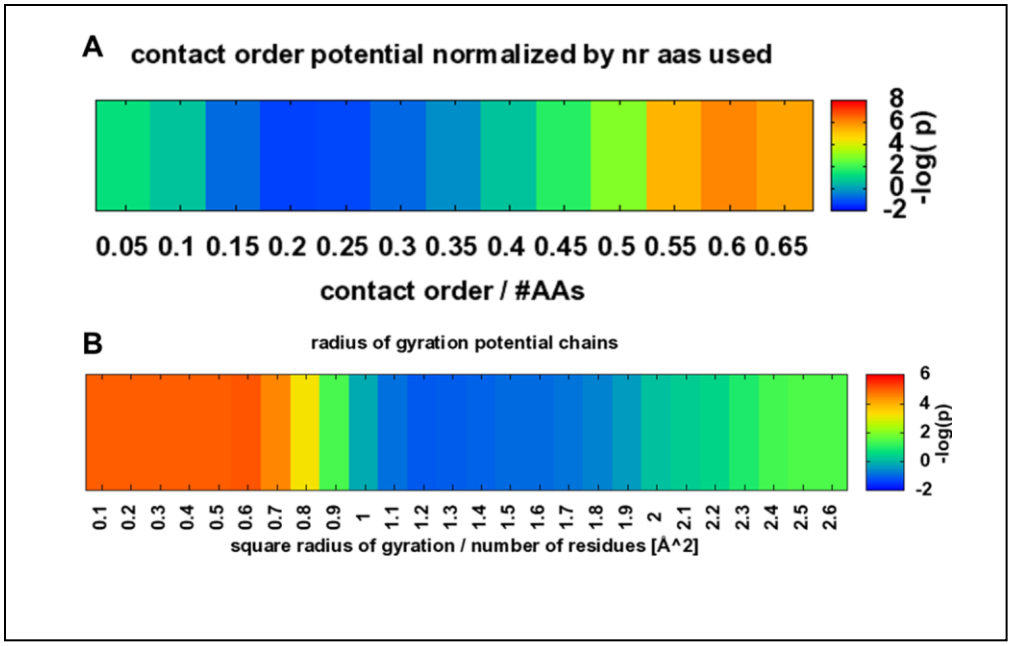

Figure 12: Contact order and Square radius of gyration potential: (A) Potential for the fold complexity is shown that is implemented by the contact order potential as the likelihood to observe a contact order to number of residues ratio in the model. **(B)** Statistics for the square radius of gyration over the number of residues was directly collected in a histogram and converted into a potential.

Radius of gyration potential

The square of the radius of gyration is proportional to en energy term that describes the compactness of the fold [30]. It is computed as the mean square distance of all C_β atom coordinates (HA2 for Glycine) to their mean position:

$$R_{gyr}^2 = \frac{1}{n}\sum_{i=1}^{n}(r_i - r_{mean})$$

When assembling protein structures from SSEs, this computation is not optimal since the protein model grows in size throughout one trajectory, leading to an increase in energy and thus a penalty when an SSE is added to the model. The new potential was derived with the assumption the square radius of gyration grow with the number of residues in the

protein model ($R_{gyr}^2 \propto n$). A potential was derived based on this correlation (Figure 12B).

$R_{gyr,expected}^2$ was determined by fitting a function to data acquired from the databank of high resolution protein structures (Figure 13). Unlike the previous term, this energy term would not penalize when assembling proteins by adding SSEs successively while still being applicable to evaluation of full sequence protein models.

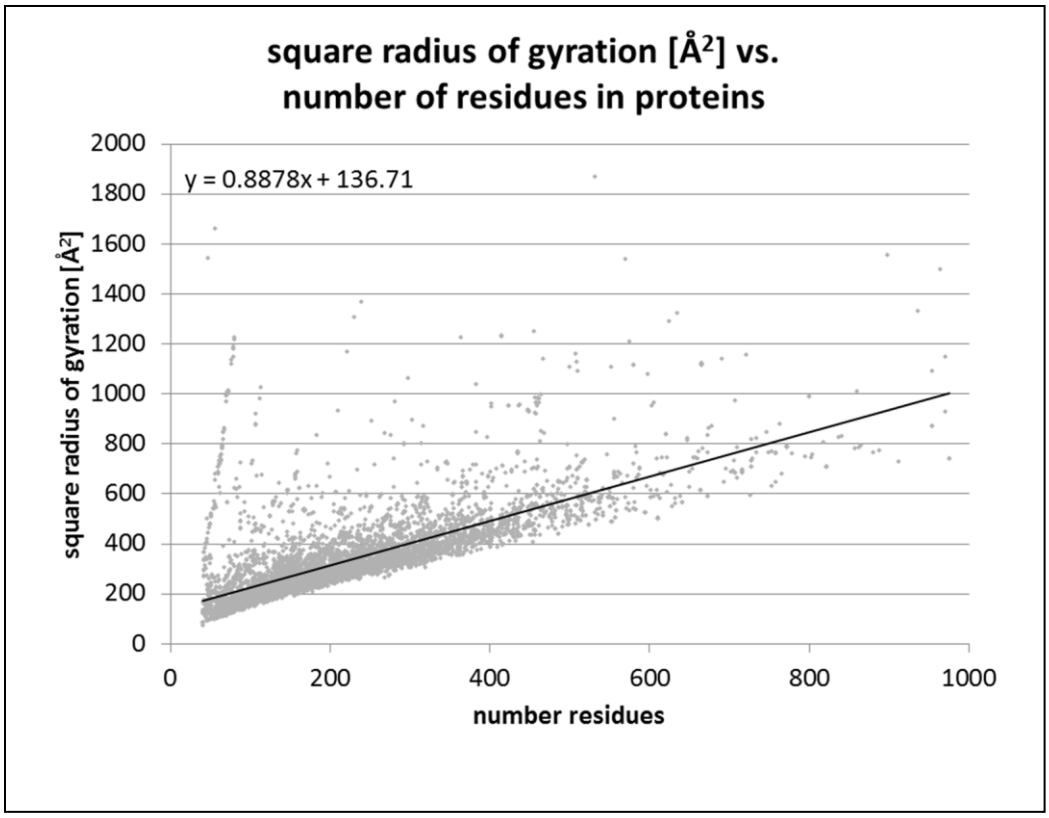

Figure 13: Square radius of gyration vs sequence length plot:

Phi Psi backbone potential

The backbone conformation of each amino acid in proteins can access certain phi and psi backbone angles. This dependency is commonly described in the Ramachandran plot. The phi and psi angles are calculated as the dihedral angle of the backbone atoms:

$$\varphi = dihedral(C_{i-1}, N_i, C_{\alpha,i}, C_{\beta,i})$$

$$\psi = dihedral(C_{\beta,i}, C_{\alpha,i}, C_i, N_{i+1},)$$

This dependency is normally amino acid and secondary structure dependent. However, since the actual secondary structure assignment is not always known, the statistics dependent only on the amino acid were derived leading to 20 individual potentials. The background distribution used is the sum over all normalized amino acid phi psi distributions. A representative phi psi backbone potential is depicted for Asparagine in Figure 8C. The most prominent accessible phi psi angle combinations can be found at 60°,60° and -150°,60°.

Amino acid clash, SSE clash and Loop closure potentials

A difficulty with Boltzman relation derived potentials is that the actual probability and background probability for events that are never observed need to be treated separately. This can be overcome by introducing a pseudo count for every event, which becomes more and more unlikely as the sample size becomes larger. This turns out to be not feasible for deriving knowledge based potentials from protein structures (data not shown). Additionally, the border conditions for “forbidden states” might require a higher

resolution than the actual potentials. Therefore, three additional terms were introduced to define constraints but do not possess discriminative abilities.

Amino acid pair clash

For the amino acid pair distance potentials, all occurring amino acid pair distances within protein structures have been calculated. They were binned with a resolution of 0.05Å for each amino acid type pair. The first bin with any counts, when iterating from shorter distances to larger distances, was determined to be the minimum permitted distance. Using this threshold, a "penalty" function is defined:

$$E(X_i X_j) = \begin{cases} d(aa_i, aa_j) \leq m_{AA} - w, 1 \\ d(aa_i, aa_j) \epsilon (m_{AA} - w, m_{AA}), \frac{1}{2}\left(cos\left(\frac{w + d(aa_i, aa_j) - m_{AA}}{w} * \pi\right) + 1\right) \\ d(aa_i, aa_j) \geq m_{AA}, 0 \end{cases}$$

With: m_{AA} Shortest observed distance for amino acid type pair

w Width of transition region

$d(aa_i, aa_j)$- Distance between amino acid pair

SSE clash potential

SSE clash potential was introduced to overcome the shortcomings observed for amino acid clash term. Since only C_β atom distances are used, certain conformations could lead to clashes but go undetected by amino acid pair clash term. An example for these kinds of conformations is when one β-strand is overlaid directly over another β-strand rotated by 180° around the main axis. To detect such clashes, a clash term that is based on the

packing of SSE fragments was derived. For this purpose, minimal distances between two SSE fragments have been defined as: helix-helix 4Å, helix-strand 4Å, strand-strand 3Å. The penalty function is similar to the amino acid pair clash function:

$$E(F_{i,}F_j) = \begin{cases} d(F_{i,}F_j) \leq m_{SS} - w, 1 \\ d(F_{i,}F_j)\epsilon(m_{SS} - w, m_{SS}), \frac{1}{2}\left(cos\left(\frac{w + d(F_{i,}F_j) - m_{SS}}{w} * \pi\right) + 1\right) \\ d(F_{i,}F_j) \geq m_{SS}, 0 \end{cases}$$

With: m_{SS} Minimal distance for that pair of SSE types

w Width of transition region

$d(F_{i,}F_j)$ Distance between the two SSE fragments

Loop closure potential

A restrictive loop closure potential is necessary to guarantee the possibility of closing loops using the residues bridging the gap between two SSEs. In order to complete the model by closing the loops, the Euclidean distance between the ends of two SSEs adjacent in sequence needs to be bridged by amino acids in the connecting loop. If the Euclidean length of the elongated loop sequence is shorter than the distance it needs to bridge, it is penalized:

$$E(SSE_i, SSE_{i+1})$$

$$= \begin{cases} \Delta d(SSE_i, SSE_{i+1}) \leq 0, 1 \\ \Delta d(SSE_i, SSE_{i+1})\epsilon(0, w), \frac{1}{2}\left(cos\left(\frac{w - \Delta d(SSE_i, SSE_{i+1})}{w} * \pi\right) + 1\right) \\ \Delta d(SSE_i, SSE_{i+1}) \geq w, 0 \end{cases}$$

$$\Delta d(SSE_i, SSE_{i+1}) = d(aa_{SSEi,last} - aa_{SSEi+1,first}) - (2.1136 + 2.5609 * n_{loopi,i+1})$$

The width of transition region is defined by w and enables a smooth transition between no and full penalty. The Euclidean distance between the backbone carboxyl-carbon of last AA of SSE_i and backbone nitrogen of first AA of SSE_{i+1} is defined the formula $d(aa_{SSEi,last} - aa_{SSEi+1,first})$. The linear function $2.1136 + 2.5609 * n_{loopi,i+1}$ estimates the Euclidean distance that can be bridge by the $n_{loopi,i+1}$ number of loop residues between SSE_i and SSE_{i+1} and was derived using the 95th percentile of the longest Euclidean distance observed for all loops of length between one and twenty amino acids in the databank (Figure 14).

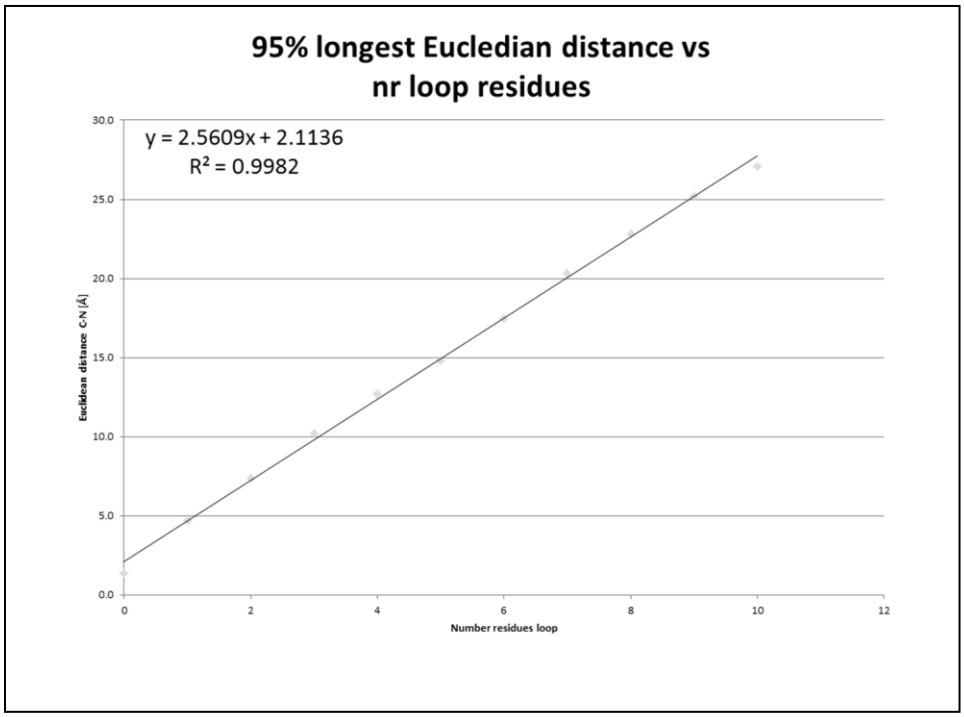

Figure 14: 95% longest Eucledian distance vs number residues in loop plot

53 protein model sets have been generated using Rosetta, BCL::Fold and perturbation

In order to benchmark the performance of the knowledge-based energy potentials, 53 diverse proteins have been selected and structural models were generated computationally using three methods: (1) Using Rosetta *de novo* protein structure prediction. (2) Removing loops from native structures and applying systematic perturbations to the structures. The sets of perturbations were chosen to generate models with preserved native-like topologies. (3) Using BCL::Fold *de novo* protein structure prediction algorithm by assembling the native secondary structure elements leading to protein models of various arrangements and topologies.

The reasoning behind using three separate methods was to obtain a diverse set of models which is more likely when a variety of sampling and scoring methods are used instead of a single one (with the exception of the perturbation method which did not have a scoring function). The identification of native-like structures was based on two measures: (1) GDT_TS < 25% and (2) RMSD100 < 8Å [6]. The percentage of such "good" models varied between 0 and 99.5% for benchmark proteins. Only model sets with percentage of good models between 1% and 99% were used for the analysis in a ten-fold cross validation calculation of enrichments. The cross validation subsets were generated by randomly removing models so that each subset contained 10% correctly folded models and 90% incorrect models.

Enrichment can evaluate the performance of an energy potential

A representative energy landscape of a set of protein models that was prepared to contain 10% of good models below an 8Å RMSD100 cutoff is depicted in Figure 15. The horizontal line denotes the best 10% of the models by the energy used. The resulting quadrants can be identified as in normal prediction experiments. Models that are below the RMSD100 cutoff are positives, and if they are also below the energy of the best 10%, they are considered as true positives (TP). If the model has a high energy despite being correct by the RMSD100, it is considered a false positive (FP). FN – false negative and TN – true negative are defined similarly. The optimal result would be to have empty FN and FP quadrants, because this would indicate that energy function would be completely accurate in identifying good models by RMSD100. The enrichment is now defined by the ratio of true positives within the 10% good models (TP+FN) divided by the initial ratio of good models by RMSD100 cutoff to the total number of models (TP+FN+FP+TN).

$$enrichment = \frac{TP}{TP + FN} * \frac{TP + FN + FP + TN}{TP + FN}$$

The maximal enrichment for a 10% cutoff will be 10, no enrichment will have the value 1, and everything that performs worse will have a value between 1 and 0.

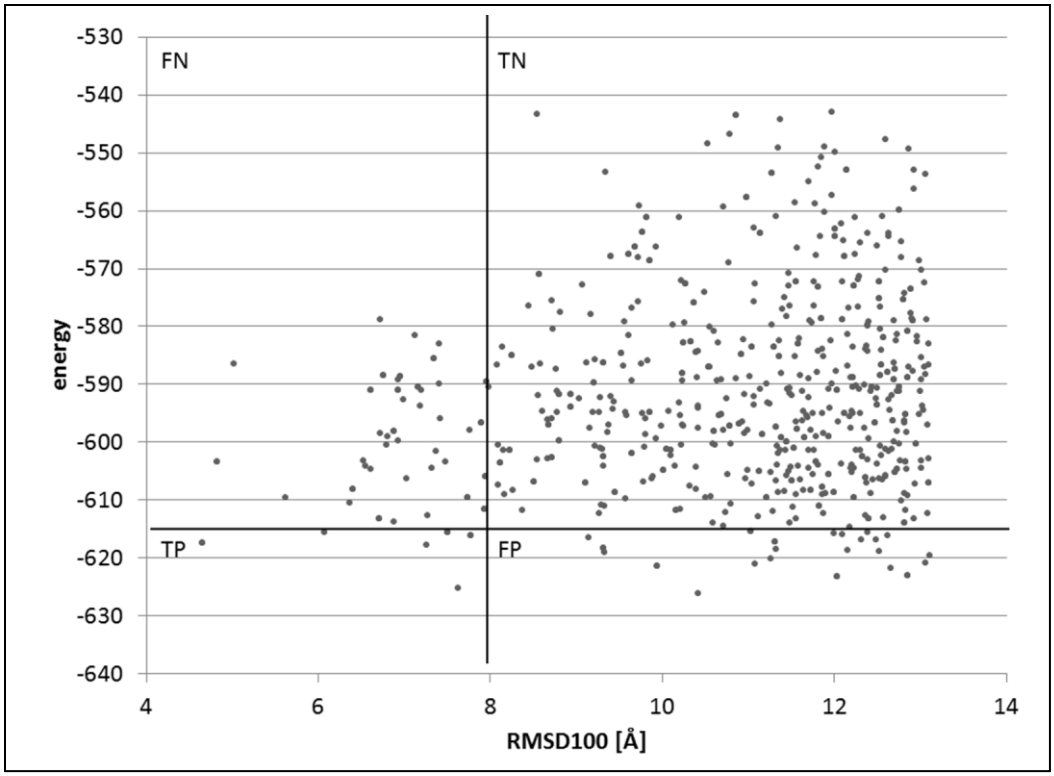

Figure 15: RMSD100 vs. energy plotted as representative energy landscape: Quadrant denoted by FN stand for false negative, TP for true positives, FP for false positives and TN for true negatives. The horizontal line divides the plot at best 10% models by energy, the vertical line at 10% of good models with RMSD100 cutoff around 8Å.

Benchmark enrichment of native like structures through potentials

Table 7 contains enrichments for various scores when evaluated on the models generated by three different methods for the 53 protein sets. The percent of model sets that could be enriched by a statistical significant factor (mean –sd > 1.0) are reported. Cells with enrichments above statistical significance (mean-1*sd ~ 85%) adding up to more than 50% of the model sets are highlighted. The loop closure and clash scores (data not shown) have a significant ability to discriminate against random models (for the BCL folded and perturbed structures) but do not perform well for Rosetta folded models. The

amino acid pair distance, amino acid neighbor count and the SSE packing potentials achieve enrichments for nearly all the protein sets.

Table 7: Percentage of models enrichment for benchmark proteins

RMSD100 < 8Å		total	amino acid distance	amino acid neighbor count	contact order	loop length	phi psi	radius of gyration	SSE packing	strand pairing	sum
	rosetta	18	**72%**	**61%**	22%	22%	**67%**	**61%**	**100%**	33%	**56%**
	perturb	53	**98%**	**100%**	**77%**	**94%**	**57%**	**55%**	**91%**	**57%**	**89%**
all	fold	14	50%	43%	43%	**57%**	36%	21%	29%	0%	50%
	rosetta	12	**83%**	**75%**	17%	25%	**67%**	**58%**	**100%**	17%	50%
	perturb	24	**96%**	**100%**	**71%**	**92%**	50%	**63%**	**79%**	4%	**75%**
alpha	fold	10	**60%**	40%	30%	50%	40%	20%	40%	0%	**60%**
	rosetta	3	**67%**	0%	33%	0%	33%	33%	**100%**	**67%**	33%
	perturb	8	**100%**	**100%**	**63%**	**100%**	38%	**63%**	**100%**	**100%**	**100%**
beta	fold	3	33%	**67%**	**67%**	**67%**	33%	33%	0%	0%	33%
	rosetta	3	33%	**67%**	33%	33%	**100%**	**100%**	**100%**	**67%**	**100%**
	perturb	21	**100%**	**100%**	**90%**	**95%**	**71%**	43%	**100%**	**100%**	**100%**
ab	fold	1	0%	0%	**100%**	**100%**	0%	0%	0%	0%	0%
	rosetta	12	**92%**	**58%**	25%	17%	**67%**	**58%**	**100%**	25%	50%
	perturb	17	**94%**	**100%**	**76%**	**100%**	41%	**82%**	**88%**	47%	**88%**
0-150	fold	9	33%	**56%**	**56%**	**78%**	33%	22%	11%	0%	44%
	rosetta	6	33%	**67%**	17%	33%	**67%**	**67%**	**100%**	50%	**67%**
	perturb	36	**100%**	**100%**	**78%**	**92%**	**64%**	42%	**92%**	**61%**	**89%**
151-300	fold	5	**80%**	20%	20%	20%	40%	20%	**60%**	0%	**60%**
GDT_TS > 25%											
	rosetta	30	50%	**67%**	30%	33%	**73%**	**67%**	**83%**	47%	**67%**
	perturb	52	**73%**	**87%**	0%	**87%**	**88%**	40%	**98%**	**60%**	**98%**
all	fold	18	**56%**	**56%**	39%	**61%**	33%	33%	**56%**	11%	**72%**
	rosetta	12	**83%**	**75%**	17%	25%	**67%**	**58%**	**100%**	17%	50%
	perturb	24	**96%**	**100%**	**71%**	**92%**	50%	**63%**	**79%**	4%	**75%**
alpha	fold	12	**100%**	**100%**	**75%**	**100%**	50%	**83%**	**100%**	0%	**100%**
	rosetta	3	**67%**	0%	33%	0%	33%	33%	**100%**	**67%**	33%
	perturb	8	**100%**	**100%**	**63%**	**100%**	38%	**63%**	**100%**	**100%**	**100%**
beta	fold	5	**100%**	**100%**	**60%**	**100%**	20%	**100%**	**100%**	**100%**	**100%**
ab	rosetta	3	33%	**67%**	33%	33%	**100%**	**100%**	**100%**	**67%**	**100%**

	perturb	20	**100%**	**100%**	**90%**	**100%**	**75%**	45%	**100%**	**100%**	**100%**
	fold	1	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
0-150	rosetta	12	**92%**	**58%**	25%	17%	**67%**	**58%**	**100%**	25%	50%
	perturb	17	**94%**	**100%**	**76%**	**100%**	41%	**82%**	**88%**	47%	**88%**
	fold	11	**100%**	**100%**	**73%**	**100%**	27%	**91%**	**100%**	45%	**100%**
151-300	rosetta	6	33%	**67%**	17%	33%	**67%**	**67%**	**100%**	50%	**67%**
	perturb	35	**100%**	**100%**	**77%**	**94%**	**66%**	43%	**91%**	**60%**	**89%**
	fold	7	**100%**	**100%**	**71%**	**100%**	**71%**	**86%**	**100%**	14%	**100%**

For each score and benchmark set, the percentage of protein model sets that had an (enrichment – 1 * standard deviation) > 1.0 are displayed. Two classifications for "good" models were used (RMSD and GDT_TS), and protein model sets have been classified as α with #helices >= 2, as β with #strands >= 2, and αβ if both conditions are fulfilled. Also a sequence length classification with <=150 and all above was performed. Cells with bold percentages and grey background highlight the cases where for more than 50% of the protein sets a significant good enrichment could have been achieved.

Discussion

Knowledge based potentials resemble first principles of physics and chemistry

Knowledge based potentials are not derived from first principles of physics and chemistry, but still resemble their consequences. The amino acid pair distance potential indicates a preference for particular side chain interactions that can be explained by Coulomb forces, van der Waals interactions or other polar or nonpolar interactions like π-stacking. The neighbor count correlates with the expectation that amino acids with polar side chain like to be exposed to the solvent as indicated by fewer neighbors in low energy neighbor counts. But it also shows that nonpolar amino acids prefer to be buried in the core, where they can escape unfavorable interaction with the solvent in addition to forming nonpolar interactions with other side chains within the core of the protein.

Secondary structure packing resembles possible geometric arrangements

The secondary structure pairing potentials cannot directly be explained by physical interactions, but they resemble the mechanical possibilities of arrangements of simplified SSE representations. The α-helix, as a cylinder with screw ridges has optimal packing for slightly tilted angles. Sheets have an internal bend of -15° and sandwiches obey to an orthogonal packing rule.

Size dependent radius of gyration measure discriminates for compact structures

A square radius of gyration measure linearly depends of the number of amino acids in the protein and can be used to evaluate the compactness of proteins and is comparable between proteins of different sizes.

Idealization does not eliminate details of interactions

The energy potentials presented are specifically designed for the problem of assembling SSEs in their ideal geometry and without explicit representation for loop residues. Although they are optimized for low resolution ideal SSEs, they can also enrich and help distinguish models that have been generated with a flexible back bone in the Rosetta program and are far away from being ideal in their SSE geometries.

Enrichments are never close to the maximum

There are two major explanations as to why maximum enrichment for any of the score for any set is never above five. Firstly, enrichment is not a linear measurement. Secondly, Rosetta energy function is already sophisticated and BCL::Fold uses the potentials for assembling the secondary structure elements, so that resulting models are optimized towards those energy functions. Every improvement would have to add to the already very successful discriminative ability to distinguish native-like protein structures from random models.

C_β atom is sufficient to approximate side chain position

The amino acid pair potential and the amino acid environment potential are both successful in discriminating for native-like protein structures which implies that a C_β atom side chain representation (HA2 for Glycine) is sufficient not only for describing possible interactions with other amino acids as a pair potential but also as an environment potential.

Enrichment can be achieved regardless of the sampling algorithm

Although Rosetta generates low resolution models, they have complete chain and defined backbone conformations. All scores except for the loop length and contact order score can enrich for native like models. Since Rosetta models are of uninterrupted sequence, the loops are already almost optimal, and the potential cannot differentiate any more. The loop length potential can enrich perturbed and BCL folded structures. Due to the

unrestrained sampling of the secondary structure elements, loops are violated and the potential is capturing this. The contact order score prevents low and highly complex folds if several SSEs are swapped or not in close proximity. This is the case for BCL folded and perturbed structures, where the potential helps regardless of size and SSE composition but when RMSD100 is used for classification. With the GDT_TS, it is possible to reach the 25% criteria by having only a partially optimal arrangement of SSEs. This yields not only to a good GDT_TS measure, but also to a better contact order score.

As expected, the strand pairing score performs well only for β-strand containing proteins. The loop length and the contact order score do not help for Rosetta folded benchmark sets, while they are important for BCL folded and perturbed structures. The best discrimination for native like models is observed for perturbed protein structures. The radius of gyration score performs well for proteins < 150 residues, but seems to degrade for larger proteins. It can be observed that the percentages of GDT_TS and RMSD100 classification drop under 50% for the perturbed structures. The perturbation protocol is designed to preserve the topology and hence, the radius of gyration of the model. This effect relative to the change in the quality measure is more relevant for larger proteins. The weighted sum of individual terms performs consistently over the benchmark set, showing that an optimal linear combination can overcome the weaknesses of the individual terms.

Methods and Materials

Divergent databank of high resolution crustal structures

Statistics have been derived from a divergent high resolution protein database which was generated using the Protein Sequence Culling Server [115]. X-ray structures with sequence identity < 25%, resolution < 2.0 Å, R-value < 0.3, sequence length > 40 residues were culled from the PDB. This guarantees that similar sequences are not over represented introducing a bias to proteins that are easier to experiment on or are of higher interest in the scientific fields.

Neighbor count

The neighbors were counted in a novel way by defining two thresholds. A neighbor with the lower radius was counted as a full neighbor, a neighbor above the higher threshold was ignored, and for neighbors within the both boundaries, the position was converted using a sine function to calculate a weight 0 and 1 which is added to the total neighbor count.

Secondary structure element packing

SSEs as defined by the secondary structure assignments in PDB files were first filtered by their lengths and α-helices with a length <7 residues and β-strands <5 residues were ignored. The remaining α-helices and/or β-strands were described as overlapping sets of fragments with lengths of 5 residues for helices and 3 residues for strands (Figure 10E). An ideal SSE fragment was superimposed with the coordinates of the backbone

coordinates of the SSE fragment from the PDB to determine the orientation (translation and rotation in Euclidean space) of this fragment. The main axes were considered to be line segments; a minimal interface length between the two SSE fragments of 4 Å was achieved by subtracting 2 Å from each end of each SSE's main axis (Figure 10B). The packing between two fragments was described by the analytical shortest connection between those two line segments. If this connection was orthogonal, it was considered to be a full contact. If the connection was not orthogonal, a contact weight was defined as a function of the angle between the main axes and the shortest connection. This angle between 90° and 0° was then used to determine a weight between 0 and 1 using half of a cosine function and for both angles those weights are multiplied.

$$w_I = \frac{\cos 2\alpha_1 + 1}{2} \frac{\cos 2\alpha_2 + 1}{2}$$

The twist between the SSE fragments is defined by the dihedral angle θ between the SSE main axes (Figure 10C). The relative offset, which is important when strand backbone hydrogen interactions could play a role, are defined by the offset angle ω between 0° and 90° (Figure 10D). For a strand-helix packing, only one offset angle can be defined, where an ω close to 90° is not favorable, a packing on to with an offset of 0° is desired, since it is dominated by amino acids side chain interactions. A weight is defined:

$$w_O = \frac{\cos 2\omega + 1}{2}$$

If two strands are involved in the interaction, it is necessary to distinguish a strand-strand backbone hydrogen bond mediated packing and a sheet-sheet (sandwich-like) amino acid side chain mediated interaction. For omega's around 90° it has a strand-strand interaction

character, if the omegas are close to 0°, it is considered to be a sheet-sandwich interaction. Two weights can be defined:

$$w_{sandwhich} = \frac{\cos 2\omega_1 + 1}{2}\frac{\cos 2\omega_2 + 1}{2}$$

$$w_{pairing} = \left(1 - \frac{\cos 2\omega_1 + 1}{2}\right)\left(1 - \frac{\cos 2\omega_2 + 1}{2}\right)$$

The actual packing between two SSEs is a list of fragment interactions (Figure 10E). This list is determined by identifying the packing of each fragment of the shorter SSE with the fragments of the longer SSE (for identical sizes, the SSE that comes first in sequence is the "shorter" one) and adding the packing with the highest interaction weight w_I to the list. These packing objects were used in the statistics for counts with the product of the weights, and later in the scoring the overall energy of the interaction by scoring each packing object scaled with their weights.

Generation of benchmark sets

The benchmark sets of protein models were generated using three different methods. 53 sequences of length between ~70 up to ~300 residues have been selected to represent diversity in respect to: helical and strand content as well as sequence length : 1AAJA, 1BGCA, 1BJ7A, 1BZ4A, 1CHDA, 1DUSA, 1EYHA, 1G8AA, 1GAKA, 1GCUA, 1GS9A, 1HYPA, 1IAPA, 1ICXA, 1IFBA, 1J27A, 1JL1A, 1K6KA, 1LKFA, 1LKIA, 1LWBA, 1M5IA, 1NFNA, 1OA9A, 1OZ9A, 1PRZA, 1ROAA, 1TZVA, 1UBIA, 1UEKA, 1VGJA, 1VK4A, 1WBAA, 1WNHA, 1WR2A, 1WVHA, 1X91A, 1XGWA, 1XKRA, 1XQOA, 2CWYA, 2E3SA, 2EJXA, 2FM9A, 2ILRA, 2IU1A, 2OF3A,

2OPWA, 2OSAA, 2YV8A, 2YVTA, 2ZCOA, 3B5OA. 10,000 models were folded *de novo* for each sequence using Rosetta [28]. Since Rosetta does not assign secondary structure, DSSP [121] was used to add definitions to the models. 10,000 models each were folded using the BCL::Fold program. Additionally, 12,000 perturbed structures were generated using the BCL::Fold program by starting with the native and applying randomly the following perturbations to the starting structure: SSE rotation and translation; SSE flip; swapping two SSEs and SSE removal. Using an RMSD100 cutoff of 8Å to the native structure was applied to identify "good" native like models as well as a GDT_TS>25% cutoff. The remaining models in each set were considered "bad" or non-native like. If there were less that 1% or more than 99% good models, that set was ignored for further analysis, since it indicates that the sampling algorithm is not suitable for that protein structure, either creating too many good models or not being able to generate enough models that could be classified as native like. The ratio good/bad are dependent on the performance of each protocol. To compensate for different ratios, 10 sets with 10% good models each were generated for each protocol and protein. Models were randomly selected from the set that is underrepresented in the good/bad ratio. These models were added to overrepresented classified models. Enrichments were calculated over all 10 sets and a mean and standard deviation is reported (data not shown). The sum was calculated as a linear combination of the potentials with a weight set (Table 8)

Table 8: Score weight set for the sum function

Aa dist	Aa neigh	loop	rgyr	sseclash	ssepack_fr	strand_fr	co_score
0.3	60	14.5	12.5	500	10	36	2.5

CHAPTER IV

BCL::FOLD – *DE NOVO* PREDICTION OF COMPLEX AND LARGE PROTEIN TOPOLOGIES BY ASSEMBLY OF SECONDARY STRUCTURE ELEMENTS

Introduction

Understanding of protein function and mechanics is facilitated by and often depends on the availability of structural information. The Protein Data Bank (PDB), as of April 2011, holds 66,726 protein structure entries, 87% determined by X-Ray crystallography and 12% determined by Nuclear Magnetic Resonance (NMR) spectroscopy, and the remaining 1% determined by Electron microscopy and hybrid methods [1,2]. The millions of protein sequences revealed by genome projects necessitate utilization of computational methods for construction of protein structural models. Comparative modeling utilizes structural information from one or more template proteins with high sequence similarity to the protein of interest to construct a model. As the PDB grows and the number of proteins with an existing suitable template of known structure increases, this method is unarguably most important [3].

However, despite impressive advancements in the combination of experimental protein structure determination techniques [4,5] with comparative modeling [6], entire classes of proteins remain underrepresented in the PDB as they evade crystallization or are unsuitable for NMR studies; e.g. membrane proteins [7] and proteins that only fold as part of a large macromolecular assembly [8,9]. Such proteins adopt more frequently

topologies not yet represented in the PDB so that the current structural knowledge fails to encapsulate necessary information to represent all protein families and folds expected to be found in the nature [10]. In such situations *de novo* methods for prediction of protein structure from the primary sequence alone can be applied.

De novo protein fold determination is possible for smaller proteins of simple topology

De novo protein structure prediction typically starts with predicting secondary structure [11,12,13,14] and other properties of a given sequence such as β-hairpins [15], disorder [16,17], non-local contacts [18], domain boundaries [19,20,21], and domain interactions [22,23]. System-learning approaches such as artificial neural networks (ANN), hidden Markov models (HMM), and support vector machines (SVM) are most commonly used in this field [24,25].

This preparatory step is followed by the actual folding simulation. Rosetta, one of the best performing *de novo* methods, follows a fragment assembly approach [26,27,28]. For all overlapping nine- and three- amino acid peptides of the sequence of interest, conformations are selected from the PDB by agreement in sequence and predicted secondary structure. Rosetta is capable of correctly folding about 50% of all sequences with less than 150 amino acids [29].

Chunk-Tasser is another fragment assembly method for de novo structure prediction that was one of the best performing methods in the CASP8 experiment [30]. This method generates chunks, three consecutive SSEs connected by two loops, using nine- and three-residue fragments. The final models are built by using these chunks as the starting point

coupled with a minimization process that also utilizes threading and distance restraint predictions [31].

For small proteins with less than 80 amino acids models can sometimes be refined to atomic-detail accuracy

During the folding simulation, most *de novo* methods use a reduced protein representation that excludes side chain degrees of freedom to simplify the conformational search space and potential. The fastest and most accurate algorithms to add side chains in order to build atomic detail models rely on sampling likely conformations of amino acid side chains, so-called rotamers [32,33,34]. At this stage, the backbone of flexible loop regions can be further refined, in Rosetta by a combination of fragment insertions, side chain repacking, and gradient minimization. In the CASP6 experiment, Rosetta was able predict *de novo* the structure of a small α-helical protein to a resolution of 1.59Å [26]. Following this success, Bradley and co-workers showed comprehensively that high resolution backbone structure prediction facilitates the correct placement of side chains and thus *de novo* high resolution structure elucidation for small proteins[35]. Note that the refinement of backbone conformations and construction of side chain coordinates aligns with most comparative modeling protocols [36,37] (Figure 1). These algorithms model gaps and insertions using loop closure algorithms that use analytical geometry [38], molecular mechanics [39], or loop libraries from the PDB [40] before entering the refinement process. Thereby both approaches – *de novo* structure prediction and comparative modeling – share the decoupling of the construction of backbone and side chain coordinates. This procedure relies on the hypothesis that accurately placed backbone coordinates define the side chain conformations [33].

Progress is stalled by inefficient sampling of large and complex topologies

De novo methods perform well only for small proteins, because the conformational search space increases rapidly as the protein gets larger. Despite simplified representation of proteins that omit side chain degrees of freedom, sampling the correct topology remains the major bottleneck for folding large proteins. Sampling is complicated for large proteins not only by size, but also by a larger number of non-local contacts, i.e. interactions between amino acids that are far apart in sequence. More of these interactions contribute to protein stability and are therefore important to sample in order to find the correct topology. At the same time, when folding a continuous protein chain each of these contacts complicates the search as conformational changes between the two amino require coordinated adjustment of multiple phi, psi angles to not disrupt the contact. To quantify the number of such non-local contacts the relative contact order (RCO) of a protein was defined which is the average sequence separation of residues "in contact", i.e. having their Cβ atoms (H2A for Glycine) within 8Å [41,42]. As the RCO increases above 0.25, the success rate of *de novo* prediction drops drastically [43]. Also, the geometry of non-local interactions and β-strand pairings in particular is often inaccurate as relative placement of the SSEs cannot be optimized independently form the connecting amino acid chain. This limitation must be overcome for *de novo* methods to be successfully applied to larger proteins. Interestingly, contact order correlates also with protein folding rates [44] suggesting that the sampling of non-local contacts is the rate-limiting step in protein folding.

De novo protein structure prediction optimally leverages limited experimental datasets for proteins of unknown topology

Interestingly, experimental structural data that become available for proteins of unknown topology are often limited, i.e. sparse or low in resolution. Typically, these limited data sets focus on and are more readily available for backbone atoms in ordered secondary structure elements. For example, cryo-Electron Microscopy and X-Ray crystallography yield medium resolution density maps of 5-10 Å where secondary structure can be identified but loop regions and amino acid side chains remain invisible [45,46,47,48,49]. NMR and EPR spectroscopy yield sparse datasets due to technological or resource limitations [49,50]. Chemical cross linking coupled with mass spectrometry has also been shown to be applicable for protein structure determination at these low resolutions [51,52,53].

While *de novo* protein structure prediction is typically insufficient in accuracy and confidence to be applied to determine the structure of a protein without the help of experimental data, a series of manuscripts was published that demonstrated the power of such technologies to predict protein structures accurately at atomic-detail when combined with limited experimental data sets of different origin. Qian et al. previously demonstrated use of *de novo* structure prediction to overcome crystallographic phase problem [54]. *De novo* methods have also been applied for rapid fold determination from unassigned NMR data[55] and structure determination for larger proteins from NMR restraints [56]. In addition, *de novo* structure prediction has also been coupled with EPR restraints [57,58,59] as well as cryo-EM [49]. Kahlkof et. al study of *de novo* structure prediction of laminin using distance restraints from natural cross-links revealed a

structural similarity to galactose binding proteins [52], which was later confirmed when the structure was experimentally determined by X-Ray [60]. Numerous other studies have also harnessed the power of *de novo* structure prediction with experimental restraints [61,62,63].

Objective of the present work is to introduce an algorithm for protein folding with a novel approach of assembly of secondary structure elements (SSEs) in three-dimensional space. This approach seeks to overcome size and complexity limits of previous approaches by discontinuing the amino acid chain in the folding simulation thereby facilitating the sampling of non-local contacts. Exclusion of loop regions focuses the sampling to the relative arrangement of rather rigid SSEs limiting the overall search space. The approach can be readily combined with limited datasets which tend to restrain the location of backbone atoms in SSEs. It leverages established protocols for construction of loop regions and side chains to yield complete protein models (Figure 1). The decoupling of the placement of SSEs from the construction of loop regions relies on the hypothesis that accurate placement of SSEs will allow for construction of loop regions and subsequent placement of side chain coordinates, a hypothesis tested excessively in comparative modeling. This approach assumes further that the majority of the thermodynamic stabilization achieved through formation of the core of the protein is defined by interactions between SSEs and can therefore be approximated with an energy function that relies exclusively on scoring SSEs. This hypothesis has been tested in the previous chapter.

Results and Discussion:

In fragment assembly based approaches to *de novo* protein structure prediction, local contacts are sampled more efficiently than the non-local ones due to inherent restrictions imposed by the connectivity of the amino acid sequence. This restriction leads to one of the major challenge in *de novo* protein structure prediction – the sampling of complex topologies as defined by the abundance of non-local contacts and thus higher relative contact order (RCO) values [43]. Further, fragment based approaches spend a large fraction of time sampling the conformational space of flexible loop regions that contribute little to the stability of the fold. Therefore the accuracies of the methods deteriorate as the conformational search space gets larger, typically for proteins with more than 150 residues. In particular β-strand interactions are often sampled insufficiently dense to arrive at the correct pairings with good geometries. In result, regular secondary structure cannot be detected in the models giving them the well-known "spaghetti"-look. The score deteriorates hampering detection of the correct topology in a large ensemble of models.

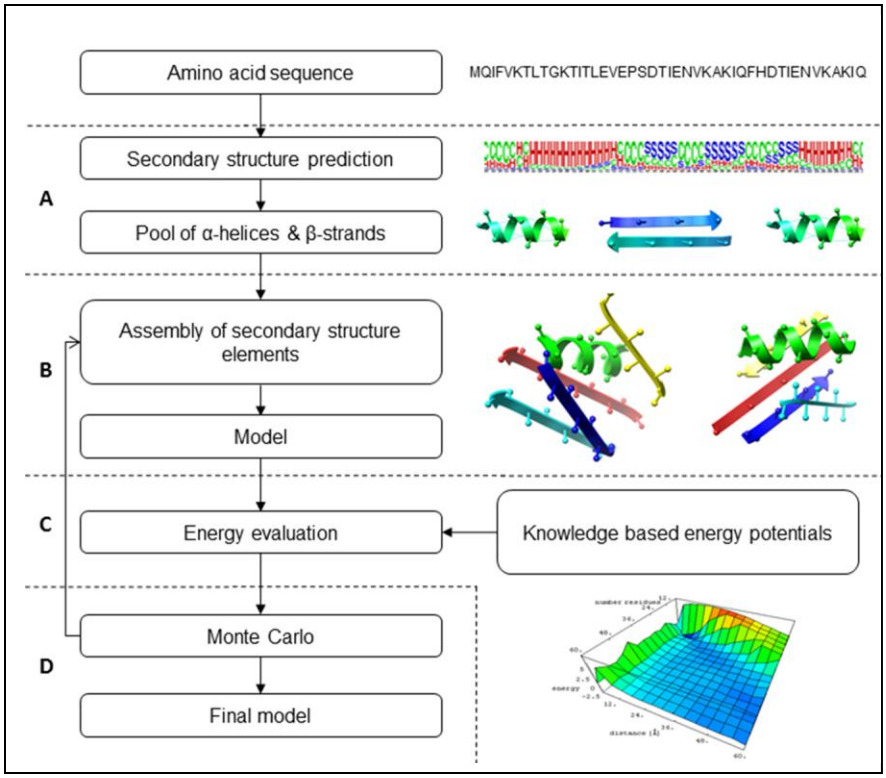

Figure 16: BCL::Fold protocol flowchart: (A) Generation of secondary structure element (SSE) pool. Three secondary structure prediction methods, PSIPRED, SAM and JUFO, have been equally weighted to achieve a consensus three state secondary structure prediction. For a given amino acid sequence, stretches of sequence with consecutive α-helix or β-strand predictions above a given threshold are identified as α-helical and β-strand SSEs and added to the pool of SSEs to be used in the assembly protocol. **(B)** Assembly of SSEs. The initial model only has a randomly picked SSE from the SSE pool. At each iteration, a move is picked randomly and applied to produce a new model. The details regarding utilized moves are given in the next panel. **(C)** Energy Evaluation using knowledge based potentials**.** After each change, the model is evaluated using knowledge based potentials. These include loop closure, amino acid environment, amino acid pair distance, amino acid clash, SSE packing,strand pairing, SSE clash and radius of gyration. **(D)** Monte Carlo Metropolis minimization. Based on the energy evaluation, models with lower energies than the previous model are accepted, while models with higher energy can be either accepted or rejected based on Metropolis criteria. The accepted models are further optimized, in case of rejected models, the minimization continues with the last accepted model. The minimization is terminated after either a specified total number of steps or a specified number of rejected steps in a row. The protocol consists of two such minimizations, one for assembly and one for refinement.

BCL::Fold is designed to overcome size and complexity limitations in de novo protein structure prediction.

BCL::Fold assembles secondary structure elements (SSEs), namely α-helices and β-strands while not explicitly modeling loop conformations (Figure 16). Individual residues are represented by their backbone and C_β atoms only, ($H_{\alpha 2}$ for Glycine). A pool of predicted SSEs is collected using a consensus of secondary structure prediction methods. A Monte Carlo Metropolis (MCM) minimization with simulated annealing is used where models are altered by SSE-based moves (Table 12) and evaluated by knowledge-based energy potentials (Table 14). The reduced representation of proteins in BCL::Fold decreases the conformational search space that has to be sampled. Moving discontinued SSEs independently of each other accelerates sampling of non-local contacts. The knowledge-based scoring function employed by BCL::Fold is described in a companion manuscript in the same issue of this journal.

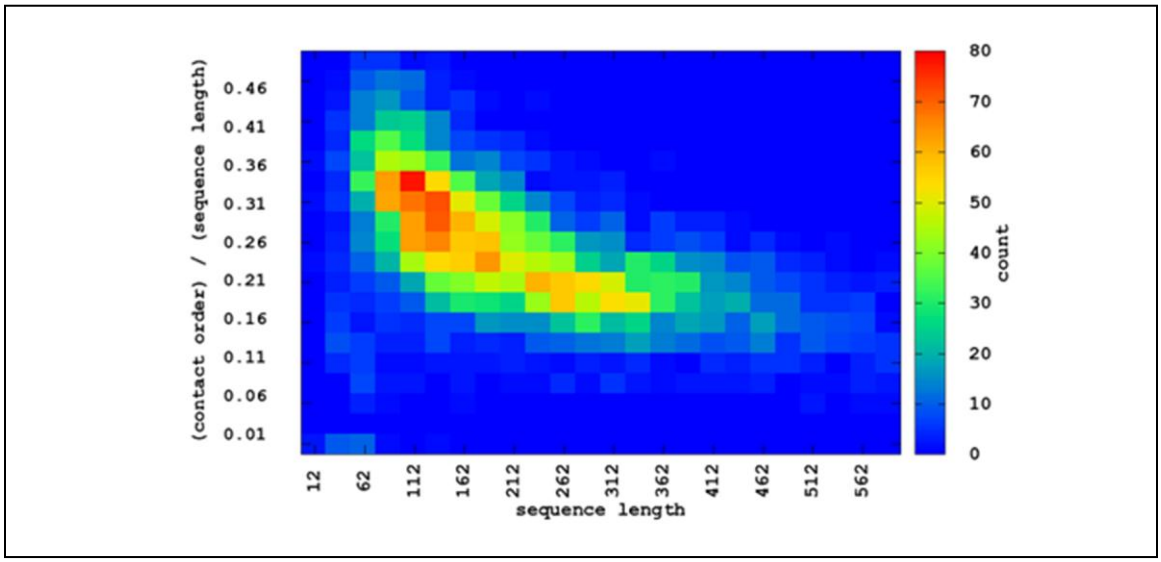

Figure 17: Contact order distribution for proteins: The heat map shows the distribution of contact order with respect to sequence lengths for ~4000 culled native proteins.

BCL::Fold was evaluated using a benchmark set of proteins collected using PISCES culling server. The set includes 66 proteins of lengths ranging from 83 to 293 residues with <30% sequence similarity. The set contains different topologies including 31 all α-helical, 16 all β-strand, and 19 mixed αβ folds (Table 9). The selected proteins have RCOs in the range of 0.12 to 0.47 with an average of 0.30 ± 0.07. It should be noted that as proteins get larger, RCO values start decreasing (Figure 17). Therefore we introduced a normalized contact order measure NCO which is defined as the square of the contact order divided by sequence length and largely independent of protein size.

Table 9: Benchmark set of proteins:

FULL SEQUENCE							FILTERED SEQUENCE					
PDB id	N_{aa}	N_{sse}	N_{α}	N_{β}	CO	RCO	N_{aa}	N_{sse}	N_{α}	N_{β}	CO	RCO
1BGCA	174	7	7	0	67.75	0.39	108	5	5	0	81.94	0.47
1EYHA	144	8	8	0	33.59	0.23	107	8	8	0	36.48	0.25
1FQIA	147	9	9	0	44.35	0.30	90	9	9	0	46.87	0.32
1GAKA	141	7	7	0	57.17	0.41	96	6	6	0	51.38	0.36
1GYUA	140	10	2	8	34.86	0.25	63	8	0	8	32.51	0.23
1IAPA	211	11	11	0	60.11	0.28	123	9	9	0	77.40	0.37
1ICXA	155	13	6	7	47.25	0.30	103	10	3	7	46.52	0.30
1J27A	102	6	2	4	44.41	0.44	76	6	2	4	46.89	0.46
1JL1A	155	10	5	5	52.69	0.34	97	10	5	5	50.41	0.33
1LKIA	180	8	6	2	73.33	0.41	113	5	5	0	76.37	0.42
1LMIA	131	10	1	9	40.95	0.31	63	9	0	9	41.77	0.32
1OXJA	173	11	11	0	35.54	0.21	108	8	8	0	30.49	0.18
1OZ9A	150	10	5	5	34.00	0.23	101	9	5	4	37.53	0.25
1PBVA	195	10	10	0	30.84	0.16	128	10	10	0	30.06	0.15
1PKOA	139	13	3	10	44.12	0.32	58	9	0	9	43.50	0.31
1Q5ZA	177	11	11	0	40.42	0.23	77	6	6	0	46.33	0.26
1RJ1A	151	8	8	0	45.07	0.30	113	7	7	0	41.83	0.28
1T3YA	141	12	6	6	30.33	0.22	83	9	4	5	25.99	0.18
1TP6A	128	9	3	6	32.97	0.26	94	9	3	6	31.72	0.25
1TQGA	105	4	4	0	36.73	0.35	88	4	4	0	38.04	0.36
1TZVA	142	9	9	0	32.42	0.23	97	7	7	0	35.14	0.25
1UAIA	224	18	2	16	57.10	0.25	114	15	0	15	55.64	0.25
1ULRA	88	7	2	5	40.11	0.46	55	7	2	5	36.68	0.42
1VINA	268	16	16	0	51.29	0.19	156	12	12	0	51.04	0.19
1X91A	153	6	6	0	48.33	0.32	113	5	5	0	46.98	0.31
1XAKA	83	7	0	7	30.22	0.36	38	6	0	6	33.08	0.40
1XKRA	206	14	6	8	65.80	0.32	147	14	6	8	66.11	0.32
1XQOA	256	14	14	0	60.32	0.24	162	14	14	0	67.52	0.26
1Z3XA	238	14	14	0	36.63	0.15	129	13	13	0	32.88	0.14
2AP3A	199	7	7	0	53.65	0.27	156	5	5	0	55.95	0.28
2BK8A	97	10	1	9	35.03	0.36	47	7	0	7	30.67	0.32
2CWRA	103	9	0	9	35.71	0.35	60	8	0	8	33.53	0.33

2EJXA	139	10	3	7	41.78	0.30	107	10	3	7	38.38	0.28
2F1SA	186	12	12	0	30.75	0.17	115	12	12	0	35.40	0.19
2FC3A	124	10	6	4	47.78	0.39	80	9	5	4	51.27	0.41
2FM9A	215	10	10	0	58.23	0.27	153	9	9	0	59.69	0.28
2FRGP	106	11	2	9	36.63	0.35	64	9	0	9	33.94	0.32
2GKGA	127	11	6	5	32.56	0.26	80	10	5	5	32.51	0.26
2HUJA	140	4	4	0	50.34	0.36	99	4	4	0	53.84	0.38
2IU1A	208	11	11	0	42.10	0.20	126	10	10	0	43.75	0.21
2JLIA	123	8	4	4	30.25	0.25	69	8	4	4	29.23	0.24
2LISA	136	6	6	0	55.90	0.41	91	5	5	0	53.23	0.39
2OF3A	266	16	16	0	34.76	0.13	202	16	16	0	31.79	0.12
2OSAA	202	11	11	0	49.60	0.25	124	9	9	0	50.70	0.25
2QZQA	152	13	3	10	46.24	0.30	63	7	0	7	52.92	0.35
2R0SA	285	16	16	0	58.40	0.20	165	13	13	0	57.84	0.20
2RB8A	104	8	0	8	33.84	0.33	46	7	0	7	29.12	0.28
2RCIA	204	13	7	6	63.82	0.31	126	10	4	6	63.77	0.31
2V75A	104	5	5	0	32.84	0.32	65	5	5	0	34.26	0.33
2VQ4A	106	10	1	9	33.71	0.32	54	8	0	8	32.07	0.30
2WJ5A	101	7	1	6	31.44	0.31	42	6	0	6	28.26	0.28
2WWEA	127	8	5	3	34.86	0.27	69	7	4	3	35.10	0.28
2YV8A	164	14	1	13	59.67	0.36	79	12	0	12	56.88	0.35
2YXFA	100	9	1	8	32.85	0.33	46	7	0	7	31.37	0.31
2YYOA	171	14	1	13	50.72	0.30	66	12	0	12	58.41	0.34
2ZCOA	293	16	16	0	51.60	0.18	205	15	15	0	56.53	0.19
3B5OA	244	11	11	0	83.49	0.34	169	9	9	0	85.09	0.35
3CTGA	129	11	7	4	33.78	0.26	68	9	5	4	32.00	0.25
3CX2A	108	10	2	8	39.67	0.37	53	7	0	7	37.05	0.34
3FH2A	146	9	9	0	43.06	0.29	100	9	9	0	42.92	0.29
3FHFA	214	13	13	0	51.79	0.24	147	12	12	0	58.19	0.27
3FRRA	191	9	9	0	54.64	0.29	141	9	9	0	55.61	0.29
3HVWA	176	14	7	7	48.29	0.27	109	11	5	6	51.62	0.29
3IV4A	112	11	6	5	35.13	0.31	77	9	4	5	32.98	0.29
3NE3B	130	11	6	5	42.02	0.32	81	9	4	5	48.43	0.37
3OIZA	99	7	3	4	26.73	0.27	63	7	3	4	25.52	0.26

For each of the 66 proteins in the benchmark set, following are displayed : 4 letter code PDB id and 1 letter code chain id, number of amino acids (N_{aa}), number of secondary structure elements(N_{sse}), number of α-helices (N_{α}), number of β-strands (N_{β}), contact order (CO), relative contact order (RCO). The left section of the table identified as "original sequence" displays statistics for the full sequence protein, while the "filtered sequence" statistics are calculated only on amino acids that are found in secondary structure elements that satistfy the length criteria; at least 5 residues for α-helices and 3 residues for β-strands.

Consensus prediction of SSEs from sequence to create comprehensive pool for assembly

The secondary structure prediction programs JUFO [64,65] and PSIPRED [66] were used to create a comprehensive pool of predicted SSEs. Two methods are used to avoid deterioration of BCL::Fold performance if one of the methods fails. To further avoid dependence on potentially incorrect predicted secondary structure we implement two strategies: a) the initial pool of SSEs contains multiple copies of one SSE having different length. In extreme cases of ambiguity this could be an α-helix predicted by one method

and a β-strand predicted by the other or one long α-helix that overlaps with two short α-helices that span the same region. b) The length of SSEs is dynamically adjusted during the folding simulation in order to allow simultaneous optimization of protein secondary and tertiary structure [13]. Both strategies require a scoring metric that analyzes the agreement of a given set of SSEs with the predicted secondary structure. Before the actual folding simulation is started, a pool of likely SSEs is created using a MCM simulation. The soring scheme and the pool generation are described in more detail in the methods section. SSEs predicted by this method are only added to the secondary structure pool if they satisfy the minimum length restrictions; five residues for α-helices and three residues for β-strands. Rationale for removal of very short SSEs is two-fold: a) the reduced accuracy of secondary structure prediction techniques for such short SSEs [67] and b) the limited contribution to fold stability expected from short SSEs (Chapter III).

Table 10 depicts Q3 [68] accuracies, as well as the percentage of native secondary structures correctly predicted and the average shifts for the SSE pools of the 66 benchmark proteins using PSIPRED and JUFO secondary structure prediction. For this set of benchmark proteins BCL::SSE generated SSE pools using PSIPRED compared to JUFO exhibit higher Q3 values (79.6% ± 10.6 vs 70.2% ± 11.9), higher native SSE recovery (96.1% ± 6.4 vs %90.3 ± 10.7). This trend is also observed for shift values (3.1 ± 2.2 vs ±4.3 ±2.8) which measure the sum of the deviations in first and last residues of the predicted SSEs when compared with native SSEs. Although PSIPRED has a better overall performance, a combined pool of PSIPRED and JUFO has the highest native SSE recovery (%96.6) and the lowest shift (2.7). Because the SSE pool is constructed in a pre-

processing step, secondary structure prediction methods can be changed or SSEs can be manually adjust if desired.

Table 10: Secondary structure pool statistics for the benchmark proteins:

pdb id	PSIPRED			JUFO			PSIPRED + JUFO	
	Q3	%found	shift	Q3	%found	shift	%found	shift
1BGCA	88.7	100.0	4.2	81.6	100.0	6.0	100.0	4.2
1EYHA	87.7	100.0	3.5	69.0	87.5	5.3	100.0	3.5
1FQIA	82.2	88.9	1.6	74.8	88.9	4.6	88.9	1.5
1GAKA	87.4	100.0	7.8	68.0	83.3	6.6	100.0	4.5
1GYUA	86.8	100.0	1.1	75.4	100.0	2.1	100.0	0.9
1IAPA	82.1	100.0	6.1	78.8	100.0	5.7	100.0	5.4
1ICXA	84.8	100.0	1.7	76.1	90.0	2.1	100.0	1.6
1J27A	96.2	100.0	0.5	71.3	83.3	2.4	100.0	0.5
1JL1A	75.5	100.0	4.1	66.7	100.0	5.4	100.0	4.0
1LKIA	75.9	80.0	10.3	44.1	80.0	16.8	80.0	10.3
1LMIA	53.7	66.7	2.8	42.5	66.7	3.8	66.7	2.5
1OXJA	83.5	100.0	6.3	76.2	100.0	4.6	100.0	2.3
1OZ9A	91.1	100.0	1.0	79.3	88.9	2.0	100.0	0.8
1PBVA	93.9	100.0	0.8	89.3	100.0	1.4	100.0	0.6
1PKOA	77.1	100.0	1.8	62.8	88.9	2.6	100.0	1.6
1Q5ZA	76.3	100.0	3.2	64.0	100.0	3.8	100.0	1.3
1RJ1A	90.2	100.0	6.0	86.6	100.0	7.4	100.0	5.3
1T3YA	73.0	100.0	2.7	77.8	100.0	2.2	100.0	2.0
1TP6A	75.5	88.9	2.6	58.8	88.9	5.0	88.9	2.6
1TQGA	96.6	100.0	0.8	82.2	100.0	4.0	100.0	0.5
1TZVA	84.8	100.0	6.4	80.0	100.0	7.0	100.0	6.1
1UAIA	68.3	93.3	1.9	64.8	86.7	2.0	100.0	1.5
1ULRA	90.2	100.0	0.9	76.5	100.0	2.3	100.0	0.7
1VINA	83.5	100.0	2.1	71.8	83.3	4.2	100.0	1.8
1X91A	88.6	100.0	2.6	79.2	80.0	4.3	100.0	2.6
1XAKA	51.2	83.3	2.8	27.3	50.0	2.3	83.3	2.8
1XKRA	85.0	92.9	1.5	80.4	85.7	1.2	92.9	1.2
1XQOA	71.8	92.9	4.5	62.5	85.7	4.3	92.9	3.3
1Z3XA	82.6	92.3	1.3	70.6	84.6	7.9	100.0	3.8
2AP3A	81.6	100.0	6.4	76.3	100.0	12.0	100.0	6.0
2BK8A	94.0	100.0	0.4	72.9	100.0	1.9	100.0	0.4
2CWRA	77.4	100.0	2.3	75.8	87.5	1.3	100.0	1.6
2EJXA	71.4	90.0	2.7	47.3	70.0	6.6	90.0	2.7
2F1SA	83.3	91.7	1.5	76.0	83.3	2.1	91.7	1.3
2FC3A	84.4	100.0	1.6	68.7	88.9	3.8	100.0	1.4
2FM9A	85.5	100.0	5.7	85.2	100.0	2.8	100.0	2.6
2FRGP	69.1	88.9	2.1	68.8	88.9	2.5	88.9	2.0
2GKGA	90.0	100.0	0.8	73.6	80.0	1.3	100.0	0.7
2HUJA	94.0	100.0	1.5	83.8	100.0	5.3	100.0	1.5
2IU1A	82.0	90.0	2.7	77.7	90.0	11.4	90.0	2.6
2JLIA	65.2	100.0	3.0	64.4	100.0	4.3	100.0	3.0
2LISA	88.2	100.0	4.6	73.1	100.0	7.4	100.0	4.6
2OF3A	85.4	100.0	7.9	78.2	87.5	5.3	100.0	5.5

2OSAA	79.4	88.9	2.4	70.3	88.9	4.0	88.9	2.1
2QZQA	48.2	85.7	3.8	43.3	85.7	4.5	85.7	3.5
2R0SA	68.6	84.6	2.2	61.3	69.2	3.1	84.6	2.2
2RB8A	80.8	100.0	1.4	82.4	100.0	1.3	100.0	1.0
2RCIA	60.7	90.0	5.3	52.8	90.0	6.3	90.0	4.4
2V75A	76.9	100.0	3.6	72.6	100.0	4.0	100.0	3.2
2VQ4A	74.2	100.0	2.1	71.0	75.0	1.3	100.0	2.0
2WJ5A	83.7	100.0	0.7	73.5	100.0	1.7	100.0	0.8
2WWEA	79.8	100.0	4.3	66.7	71.4	3.2	100.0	4.1
2YV8A	81.2	91.7	1.0	75.3	83.3	1.3	91.7	0.5
2YXFA	69.2	100.0	2.3	55.7	100.0	3.4	100.0	2.1
2YYOA	69.1	100.0	2.2	62.0	100.0	3.0	100.0	2.1
2ZCOA	83.8	93.3	5.7	81.0	100.0	8.3	100.0	5.1
3B5OA	72.3	100.0	9.1	58.6	88.9	8.1	100.0	7.4
3CTGA	83.1	100.0	1.4	67.5	77.8	2.4	100.0	1.4
3CX2A	75.4	100.0	1.3	67.2	100.0	2.4	100.0	1.4
3FH2A	96.0	100.0	0.4	89.4	100.0	3.4	100.0	0.3
3FHFA	68.9	91.7	5.4	63.3	91.7	8.4	100.0	6.7
3FRRA	93.0	88.9	5.4	86.2	88.9	6.5	88.9	5.3
3HVWA	60.0	90.9	3.9	54.6	72.7	4.0	90.9	2.5
3IV4A	83.1	100.0	1.7	81.0	100.0	2.0	100.0	1.3
3NE3B	80.9	100.0	1.3	76.7	100.0	2.3	100.0	1.1
3OIZA	68.0	100.0	3.6	64.5	100.0	3.9	100.0	3.3
avg	79.6	96.1	3.1	70.2	90.3	4.3	96.6	2.7
stdev	10.6	6.4	2.2	11.9	10.7	2.8	6.4	1.9

The table depicts Q3 score, percentage of native SSEs identified as well average shifts for the pools generated using secondary structure prediction methods PSIPRED, JUFO and combined PSIPRED+JUFO for all of the 66 proteins in the benchmark set. The last two rows contains the average and the standard deviations.

Two-stage assembly and refinement protocol separates moves by type and amplitude

BCL::Fold samples the conformational search space by a variety of SSE-based moves. These moves coupled with exclusion of loop residues, provide a significant advantage in fast sampling of different topologies. The minimization process is divided into two stages. The “assembly” stage consists of large amplitude translation or rotations and moves that add or remove SSEs. Other moves central to this phase shuffle β-strands within β-sheets or break large β-sheets to create β-sandwiches. The “refinement” stage focuses on small amplitude moves that maintain the current topology but optimize

interactions between SSEs and introduce bends into SSEs. Currently both stages utilize the same energy function (Chapter III).

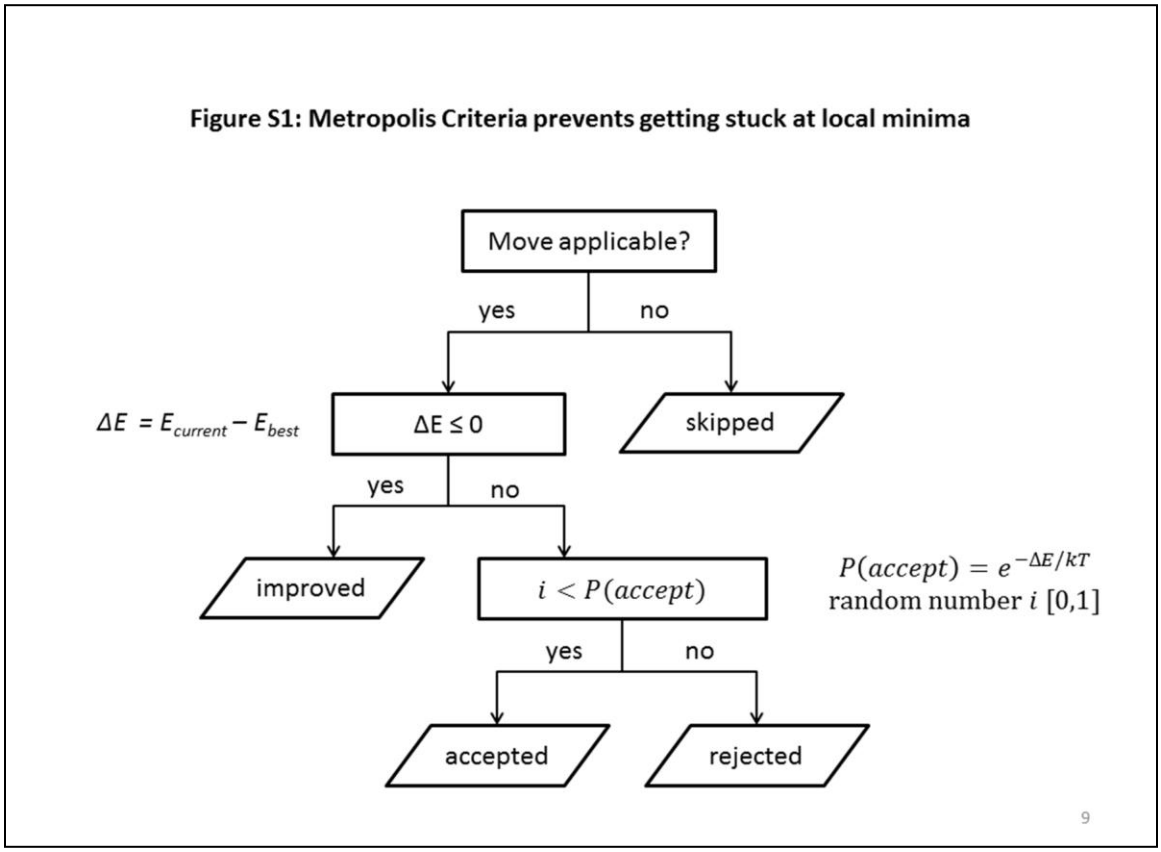

Figure 18: Metropolis Criteria: At each step, the result is evaluated and step is assigned one of four possible outcomes; skipped, improved, accepted, and rejected. The first check determines whether the move was applicable or not. In cases where the SSEs required for the move are not yet added to the model, the step is determined to be as skipped. If the move was successfully applied, then the energy difference to the last improved model is calculated, if the energy has improved (became lower), then the step is assigned as improved. If the energy increased, then a random number is used to determine whether the move should still be accepted. The acceptance ratio for this purpose is biased by the amount of the increase in the energy.

Once the SSE pool is input, the algorithm initializes the energy functions and move sets with corresponding weight sets for assembly and refinement stages. A starting model for the minimization is created by inserting a randomly selected SSE from the pool into an

empty model. The starting model is passed to the minimizer which executes assembly and refinement minimization. The assembly stage terminates after 5000 steps in total or after 1000 consecutive steps that did not improve the score. The refinement stage terminates after 2000 steps in total or 400 consecutive steps that did not improve the score. In general a move can result in one of four outcomes (Figure 18): "improved" in score, "accepted" through Metropolis criterion, "rejected" as score worsened, or "skipped" as SSE elements required for the move are not present in the model. The temperature is adjusted dynamically based on the ratio of accepted steps (see Methods).

A comprehensive list of all moves used in BCL::Fold is given in Table 12 along with brief descriptions. The moves are categorized into six main categories; (1) adding SSEs, (2) removing SSEs, (3) swapping SSEs, (4) single SSE moves, (5) SSE-pair moves, and (6) moving domains, i.e. larger sets of SSEs. Representations for a selection of moves used in BCL::Fold are illustrated in Figure 19. SSE, SSE-pair and domain moves are further categorized into specific versions for α-helices and β-strands or α-helix domains and β-sheets resulting in a total of nine individual categories. The relative probability or weight for each move category is initialized at the beginning of the minimization and depends on the SSE content of the pool. For example, β-sheet moves are excluded if the given pool contains only α-helices. This procedure limits the number of move trials that are unsuccessful or "skipped" because the needed SSEs are not in the model. As mentioned in the previous section, depending on the amplitude, moves are categorized to be used in either the assembly stage or the refinement stage. Out of 106 moves, 72 are used exclusively in assembly and 33 are used exclusively in refinement. Resizing SSEs ("sse_resize") is the only move used in both stages. Table 13 provides statistics of how

frequently each move leads to an improved, accepted, rejected, or skipped status as well as the average improvement in the score observed for all the improved steps based on statistics collected on the 64 benchmark proteins. Assembly moves have an average score improvement of -170 ± 101 BCLEU while the refinement moves have an average score change of -29 ± 21 BCLEU (Table 13).

The five individual moves with largest score improvements either add SSEs or manipulate β-strand, including "add_strand_next_to_sheet", "sheet_pair_strands", "add_sse_short_loop" and "add_sse_next_to_sse". At the same time, these moves also lead to improved models with a relatively high percentage, ranging from 10% to 30% of the cases where the move is not skipped. On the other hand, these moves, especially ones including adding SSEs, also lead to a high percentage of skipped steps. This is due to the fact that the weight for these moves is currently not dynamically adjusted depending on how many SSEs are already added to the model. On the contrary, moves with small average score improvements are less frequently skipped but also less frequently accepted. It is somewhat misleading to analyze the moves in isolation as rearranging or refining the topology often requires a series of different moves and success of one move relies on availability on suitable companion moves.

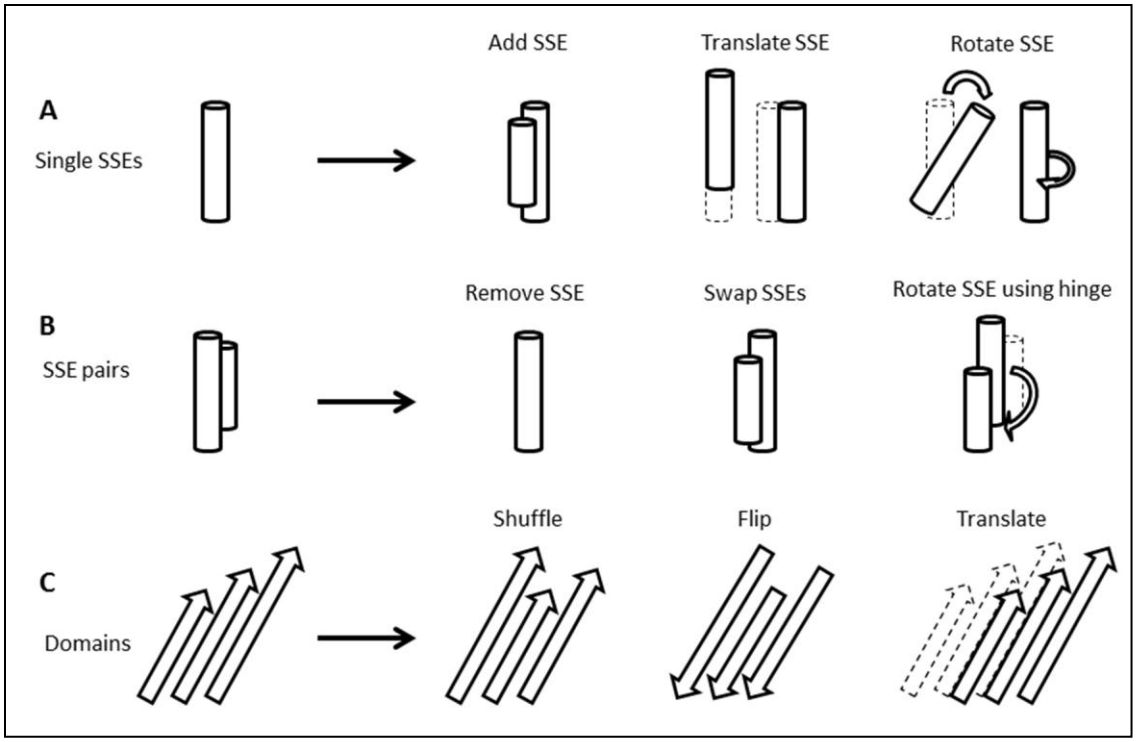

Figure 19: SSE-based moves allow rapid sampling in conformational search space: The type of moves used in BCL::Fold protocol are explained with a representative set. **(A)** Single SSE moves: These moves can including adding a new SSE to the model from the pool as well as translation/rotations/transformations. **(B)** SSE pair moves: One of the SSEs in the pair can be removed, the locations can be swapped and one can be rotated around the other SSE which is used as a hinge to define rotation axis. **(C)** Domain based moves: These moves act on a collection of SSEs such as helical domain or β-sheets. The examples show how the locations of strands can be shuffled within in a β-sheet or how a β-sheet can be flipped externally or translated together.

BCL::Fold samples native-like topologies for 92% of benchmark proteins

10,000 structural models were generated for each protein in the benchmark set using BCL::Fold. As described, two separate runs were performed with BCL::Fold, one with using a SSE pool composed of native SSE definitions as computed from the experimental structures using DSSP [69]. A second run was performed using a BCL::SSE predicted

pool. To facilitate analysis of models loops were constructed using a rapid CCD based method [38](see Methods). However, in the present analysis we focus on placement of SSEs to form the topology and evaluate models using two qualities measures; RMSD100 and Contact Recovery (CR). CR measures percentage of native contacts recovered, where a contact is defined as between two amino acids of at least 12 residues sequence separation and <8Å Cβ distance. The average and standard deviations of RMSD100 and CR values of the best models generated by these runs can be found in Table 11**.** Figure 22 and Figure 23 illustrate the best RMSD100 SSE-only and complete structural models generated by BCL using predicted SSE pools for a selection of benchmark proteins.

BCL::Fold using the correct secondary structure RMSD100-values of 5.5 ± 1.6Å (SSE only models) and 6.8 ± 1.7Å (complete models) were achieved. For simulations with predicted SSEs RMSD100 values of 6.0 ± 1.6Å (SSE only models) and 7.2 ± 1.7Å (complete models) were obtained. For comparison, ROSETTA [28] generated models with RMSD100-values of 6.4 ± 2.1Å. BCL::Fold improved the RMSD100 when compared with Rosetta in 24 cases (36%) with correct SSE definitions and in 19 cases (29%) using a predicted SSE pool. When CR values are considered, BCL::Fold using the correct secondary structure achieved 44.6 ± 15.1 (SSE only models) and 45.0 ± 15.0 (complete models). For simulation with predicted SSEs CR-values of 39.6 ± 15.3 (SSE only models) and 41.9 ± 15.0 (completed models) were obtained. For comparison, ROSETTA generated models with CR-values of 39.4 ± 17.5. BCL::Fold improved the recovery of native contacts when compared with Rosetta in 47 cases (71%) with correct SSE definitions and in 40 cases (60%) using a predicted SSE pool.

When best models by RMSD100 are considered, BCL::Fold was able to predict the correct topology in 61 cases (92%) independent of usage of correct or predicted SSE pools. After loop construction native-like models are obtained for 50 cases (75%) using correct SSE predictions and 41 cases (62%) using a predicted SSE pool. In comparison Rosetta constructed native-like models for 45 cases (68%). When a CR value of >20% is taken as cutoff success rates change to 64 cases (97%) and 62 cases (94%), respectively, for BCL::Fold and to 60 cases (91%) for Rosetta. We attribute the deterioration of BCL::Fold models after loop construction mostly to limited sampling performed at this stage of the protocol as the present work focuses on topology assembly.

For further analysis, the best-scoring 100 models (1%) for each protein and each method were kept. For these subsets the percentage of targets where the best model by RMSD100 was below 8.0Å were calculated. BCL::Fold using correct SSEs was able to generate a <8.0Å RMSD100 model in top 1% by score for 56% of targets (SSE only models) and 39% (complete models). These values for BCL::Fold simulations with predicted pools were 44% (SSE only models) and 22% (complete models). This was followed by a similar analysis where CR measure was used instead of RMSD100. BCL::Fold using correct SSEs was able to generate >20% CR models for 74% of targets (SSE only models) and 79% (complete models). For simulations with predicted pools CR values were 73% (SSE only models) and 76% (complete models).

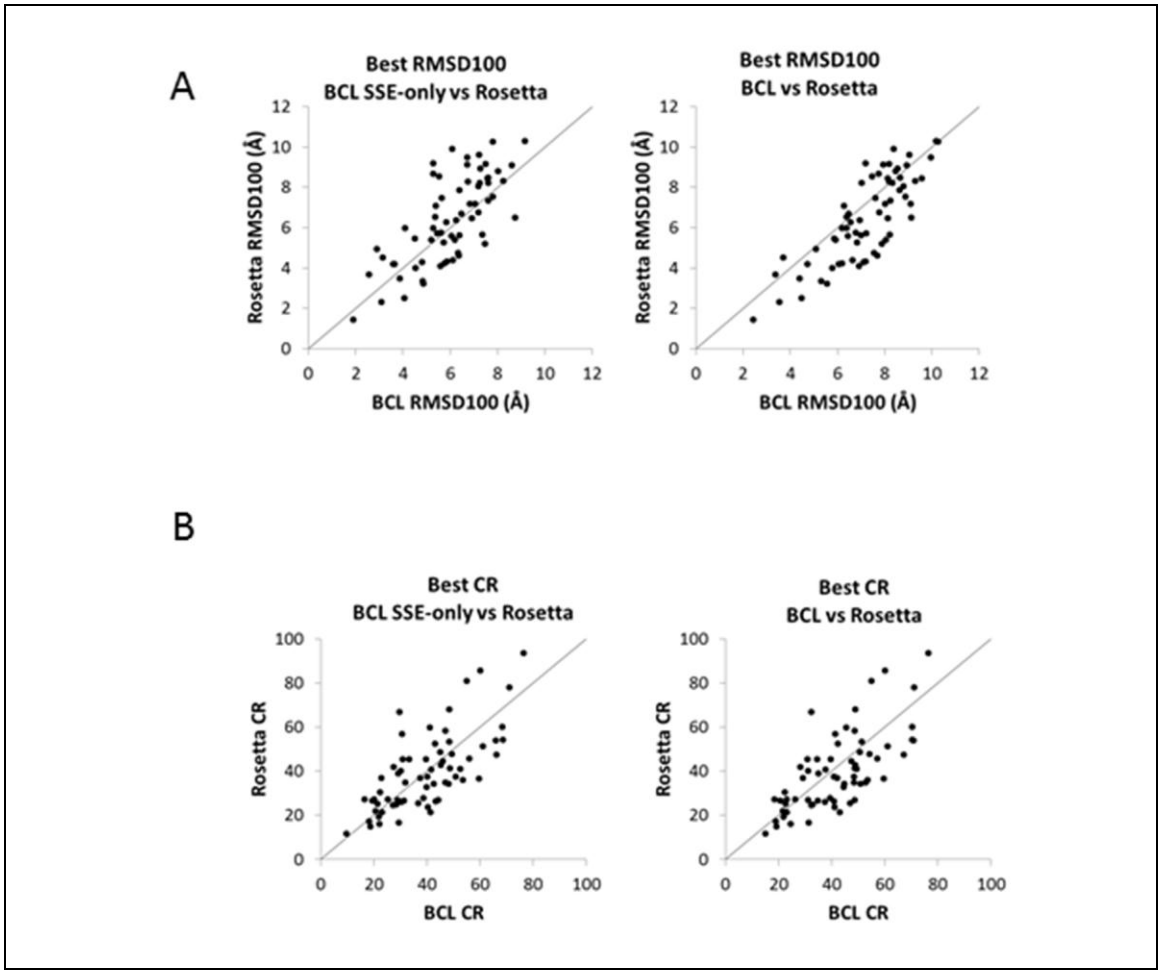

Figure 20: Comparison of best RMSD100 and CR values for BCL and Rosetta: Scatter plot comparing **(A)** best RMSD100 or **(B)** best CR SSE-only (left) and complete (right) BCL models vs Rosetta models. The BCL models considered are from BCL::Fold runs using predicted SSE pools. **(B)** Scatter plot comparing best CR SSE-only (left) and complete (right) BCL models vs Rosetta models. The BCL models considered are from BCL::Fold runs using predicted SSE pools.

Comparison of best RMSD100 and CR values achieved for all benchmark proteins between Rosetta and BCL are provided in Figure 20. When RMSD100 values are considered, SSE-only models for BCL runs with predicted SSE pools (Figure 20A left panel) provide a better performance than Rosetta. As explained earlier, SSE-only models are given an advantage due to the smaller number of atoms over which RMSD100 values

are calculated for these models in the lack of flexible loop regions. When complete models are compared with Rosetta (Figure 20A right panel); it is observed that Rosetta produces lower RMSD100 models for more targets although performance correlates very well. Figure 20B displays CR values giving a light advantage to the BCL in recovering native-like SSE contacts. These results are promising for BCL::Fold especially given the fact that BCL::Fold was designed with a focus on getting the SSE topology correct.

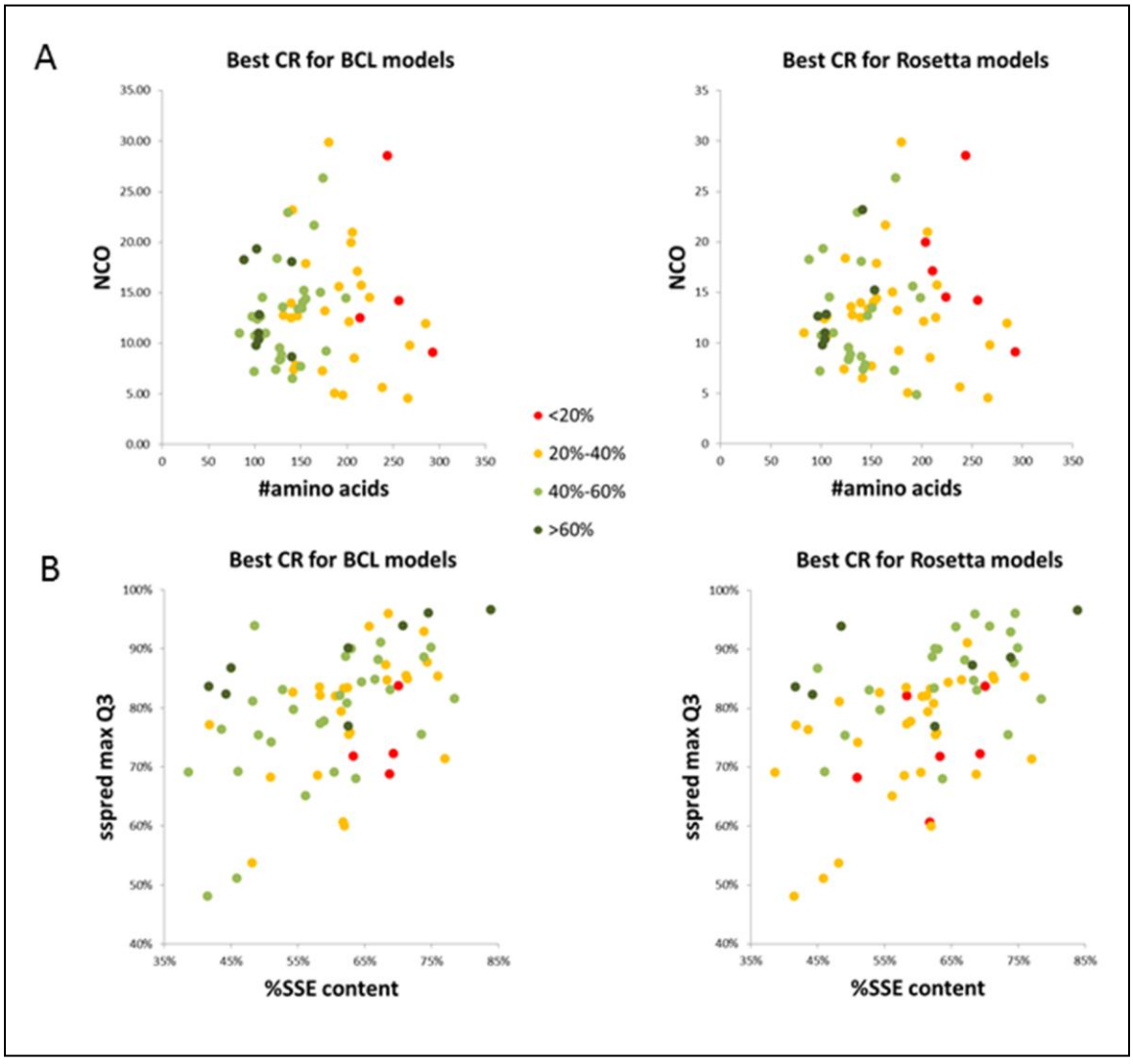

Figure 21: Determinants of high CR values in BCL and Rosetta models: (A) Plot of sequence length in number amino acids vs relative contact order (RCO) for all benchmark proteins. **(B)** Plot of percentage of amino acids found in SSEs vs maximum

Q3 value achieved from JUFO or PSIPRED pools for all benchmark proteins. Individual plots are presented for BCL models from BCL::Fold runs using predicted SSE pools (left panels) and Rosetta models (right panels). Points in both (A) and (B) are colored according to the best CR value achieved for that benchmark protein in BCL runs using predicted SSE pools and complete models; <20% (red), 20% to 40% (orange), 40% to 60% (green) and >60% (dark green).

BCL::Fold performance varies between different targets, as observed in the plots mentioned above. We wanted to investigate whether there is a correlation of performance with sequence length, fold complexity, secondary structure content, or accuracy of the secondary structure prediction. For this purpose, for each benchmark protein, the sequence length is plotted against NCO values (Figure 21A left panel) and each point is colored according to the highest CR value achieved for the complete models generated by BCL::Fold runs using predicted SSE pools for that protein. As seen in the plots the best performing proteins (>60%), are limited to <150 residue proteins. On the other hand, 40% to 60% CR values were achieved for proteins up to 200 residues, and 20% - 40% CR values were attainable for proteins up to 275 residues. Similar plots are also provided for Rosetta models in Figure 21A right panel for comparison.

Table 11: Best RMSD100 and CR values for models generated by BCL and Rosetta

	RMSD100					cr12				
pdbid	$BCL_{N\text{-}SSE}$	BCL_{N}	$BCL_{P\text{-}SSE}$	BCL_{P}	Rosetta	$BCL_{N\text{-}SSE}$	BCL_{N}	$BCL_{P\text{-}SSE}$	BCL_{P}	Rosetta
1BGCA	2.94	4.25	5.41	6.29	7.06	61.11	59.26	45.37	49.07	42.59
1EYHA	6.06	6.92	5.87	7.20	4.30	28.69	31.15	41.80	37.70	40.77
1FQIA	7.17	7.60	6.20	8.06	5.37	38.46	36.92	40.00	44.62	32.58
1GAKA	4.90	7.06	6.38	7.69	4.60	42.59	42.59	29.63	32.41	66.67
1GYUA	4.41	6.39	4.11	6.39	5.96	58.68	58.68	61.16	61.16	51.24
1IAPA	6.46	7.55	7.38	8.23	5.65	27.12	27.12	22.03	22.03	19.11
1ICXA	6.51	6.80	6.07	6.46	5.59	45.74	46.28	51.06	48.40	37.44
1J27A	3.20	3.62	3.15	3.72	4.49	71.20	70.40	68.80	70.40	54.07
1JL1A	6.04	8.01	6.75	8.19	8.26	39.05	43.33	31.43	34.76	26.52
1LKIA	2.90	5.40	7.07	9.10	7.18	59.29	59.29	23.01	29.20	36.70
1LMIA	5.82	8.60	6.72	9.97	9.49	49.65	48.94	29.08	31.21	26.95
1OXJA	6.42	7.67	7.21	7.79	6.75	44.68	45.74	30.85	30.85	45.45
1OZ9A	5.78	6.88	5.22	5.93	5.39	38.36	43.40	40.25	40.88	37.50
1PBVA	8.02	9.02	8.75	9.14	6.47	28.30	28.30	30.19	31.13	40.00

1PKOA	6.03	7.81	7.58	8.15	8.43	43.07	40.88	38.69	39.42	27.74
1Q5ZA	5.64	8.03	7.28	8.56	8.93	40.00	40.00	41.54	43.08	21.28
1RJ1A	4.66	5.52	4.86	5.33	3.33	48.43	48.43	55.97	57.23	45.56
1T3YA	5.71	5.92	5.86	6.56	6.27	49.61	51.94	42.64	44.96	34.06
1TP6A	4.91	5.65	5.74	6.83	5.25	49.65	53.15	46.15	47.55	44.59
1TQGA	1.91	2.19	1.92	2.44	1.41	76.40	76.40	76.40	76.40	93.55
1TZVA	4.55	6.01	4.89	5.58	3.20	46.28	46.28	39.67	39.67	45.45
1UAIA	6.13	7.95	7.24	9.05	9.62	38.58	37.01	29.53	31.50	16.54
1ULRA	3.17	4.82	3.61	4.73	4.18	62.14	63.11	66.02	70.87	53.98
1VINA	7.42	8.53	7.62	8.68	8.48	25.46	25.46	20.83	21.76	21.98
1X91A	2.40	3.30	4.08	4.49	2.49	77.64	77.64	48.45	49.07	68.02
1XAKA	8.17	9.53	5.28	7.77	8.67	53.03	53.03	31.82	48.48	34.85
1XKRA	6.14	7.29	8.04	8.47	8.79	28.47	30.29	27.37	32.48	24.51
1XQOA	8.05	8.71	7.50	8.20	9.16	18.82	18.28	18.82	19.35	14.73
1Z3XA	7.74	9.19	7.58	9.59	8.44	24.27	24.27	22.33	22.33	30.37
2AP3A	2.78	3.16	3.67	6.06	4.17	52.52	51.08	43.17	42.45	52.26
2BK8A	5.09	6.89	4.81	7.13	4.27	56.41	57.69	55.13	55.13	80.77
2CWRA	5.99	6.05	5.66	7.63	7.46	45.80	45.04	44.27	48.85	26.72
2EJXA	6.28	6.64	7.48	7.89	5.17	41.43	42.14	29.29	35.00	38.82
2F1SA	6.68	7.93	7.61	8.26	7.34	27.27	28.28	25.25	26.26	27.27
2FC3A	4.90	7.39	5.63	6.78	5.75	34.62	40.00	36.92	46.92	25.36
2FM9A	6.22	7.05	6.26	6.95	6.37	24.05	24.05	21.52	22.78	25.14
2FRGP	4.67	5.69	5.38	6.41	6.53	57.02	56.14	53.51	53.51	35.96
2GKGA	3.43	4.31	3.89	4.40	3.45	52.10	52.94	47.06	48.74	58.20
2HUJA	2.12	2.98	2.57	3.37	3.65	71.31	71.31	66.39	67.21	47.29
2IU1A	6.70	7.99	6.84	8.04	7.16	25.16	25.79	20.13	23.27	27.06
2JLIA	6.18	7.11	6.93	8.14	6.46	46.59	46.59	37.50	42.05	36.89
2LISA	4.77	6.03	5.47	7.22	5.71	43.33	43.33	41.11	45.56	59.79
2OF3A	8.92	9.26	8.25	9.30	8.30	20.24	20.65	19.43	20.65	26.58
2OSAA	6.55	7.78	7.21	8.82	8.05	27.14	27.14	28.57	32.86	24.86
2QZQA	5.48	8.73	6.10	8.40	9.89	63.24	61.03	47.06	52.94	34.81
2R0SA	6.95	9.53	7.80	10.29	10.27	24.49	24.49	23.13	23.13	21.30
2RB8A	3.27	5.07	2.91	5.09	4.91	62.11	61.05	68.42	70.53	60.00
2RCIA	5.32	6.99	9.17	10.20	10.29	47.50	48.75	22.08	24.58	16.06
2V75A	3.26	3.66	3.11	3.55	2.29	67.12	65.75	60.27	60.27	85.71
2VQ4A	4.18	6.65	5.28	7.20	9.18	57.94	57.01	59.81	59.81	36.45
2WJ5A	5.80	8.48	6.41	8.63	7.86	67.80	66.10	71.19	71.19	77.97
2WWEA	4.92	6.43	5.30	6.22	5.97	45.95	48.65	48.65	48.65	41.38
2YV8A	5.71	7.85	5.54	7.48	8.53	47.62	47.02	43.45	41.07	26.19
2YXFA	5.84	7.28	6.13	6.65	4.38	51.46	51.46	48.54	51.46	53.40
2YYOA	6.68	8.55	6.72	7.94	9.13	37.91	39.22	40.52	41.18	23.53
2ZCOA	7.75	8.42	7.63	8.33	8.20	18.28	19.03	18.28	19.03	17.18
3B5OA	6.46	7.28	8.62	8.96	9.10	23.11	23.11	9.78	15.11	11.43
3CTGA	5.52	6.89	5.61	6.93	4.07	52.11	53.52	45.07	50.70	48.72
3CX2A	4.96	7.88	7.27	7.04	8.20	54.37	54.37	49.51	54.37	47.57
3FH2A	6.37	7.34	6.35	7.55	4.73	33.33	33.97	27.56	28.21	41.85
3FHFA	8.60	9.17	7.81	8.88	7.54	19.42	21.36	16.50	18.45	27.07
3FRRA	6.50	7.50	4.53	5.87	5.46	30.82	30.82	33.33	34.59	45.25
3HVWA	6.10	8.02	6.48	6.49	6.69	39.61	40.26	30.52	37.66	25.95
3IV4A	3.34	4.68	4.54	5.80	3.98	60.61	61.62	52.53	49.49	40.95
3NE3B	5.01	5.65	6.41	7.01	5.60	45.45	48.25	48.25	51.05	34.10
3OIZA	4.48	5.00	5.76	6.21	4.20	50.00	50.96	30.77	41.35	56.76
avg	5.50	6.81	6.04	7.21	6.42	44.55	44.96	39.63	41.88	39.42
stdev	1.61	1.73	1.58	1.68	2.16	15.13	14.81	15.31	14.96	17.52

The table lists for all proteins, best RMSD100 and best CR observed for models generated by BCL and Rosetta. BCL results are presented in 4 columns: SSE only models using native SSE definitions ($BCL_{N\text{-}SSE}$), complete models using native SSE definitions (BCL_{N}), SSE only models using predicted SSE definitions ($BCL_{P\text{-}SSE}$), complete models using predicted SSE definitions (BCL_{P}). The 5th columns under RMSD100 and GDT_TS are for Rosetta models. The average values and standard deviations could be found in the last two columns

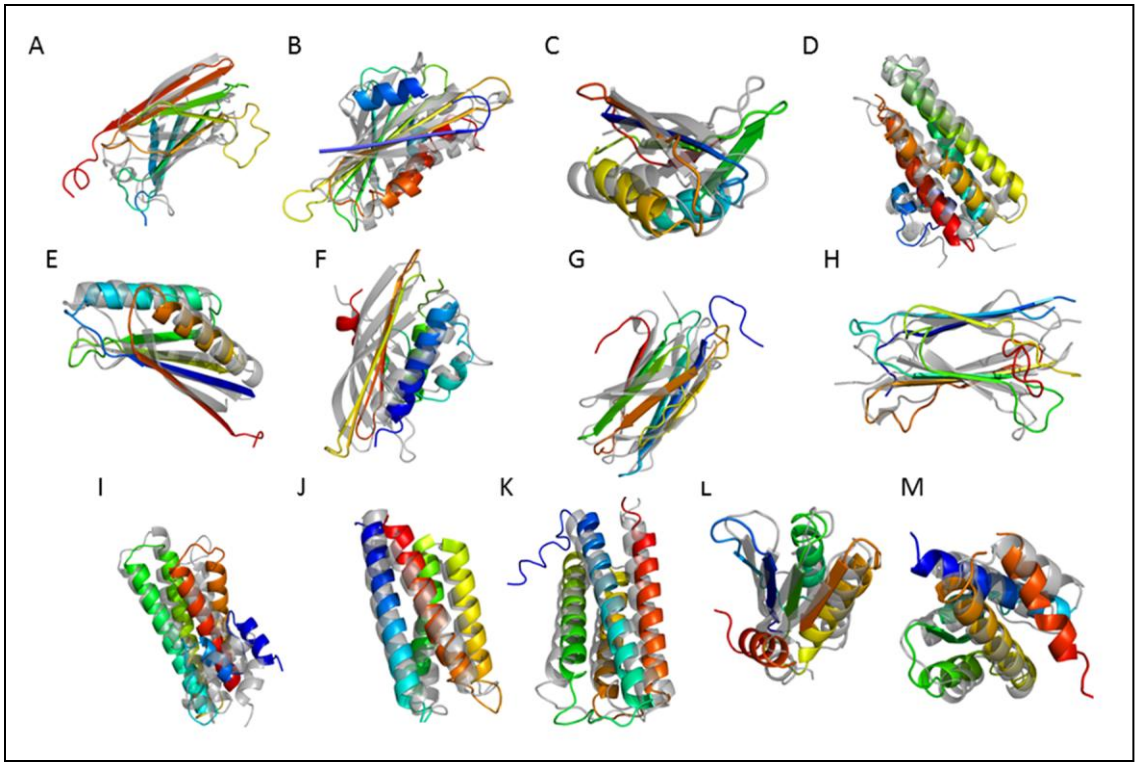

Figure 22: Structures for a selection of best RMSD100 SSE-only models generated by BCL::Fold: BCL::Fold generated best RMSD100 complete models using predicted SSE pool for a selection of proteins. The generated models are rainbow colored and superimposed with the native structure (gray) for following proteins along with the RMSD100 of the models: **(A)** 1GYUA – 6.39Å **(B)** 1ICXA – 6.46Å **(C)** 1ULRA – 4.73Å **(D)** 1X91A – 4.49Å **(E)** 1J27A – 3.72Å **(F)** 1TP6A – 6.83Å **(G)** 2CWRA - 7.61Å **(H)** 2RB8A – 5.09Å **(I)** 1RJ1A – 5.33Å **(J)** 1TQGA 2.44Å **(K)** 2HUJA – 3.37Å **(L)** 3OIZA – 6.21Å **(M)** 2V75A – 3.55Å

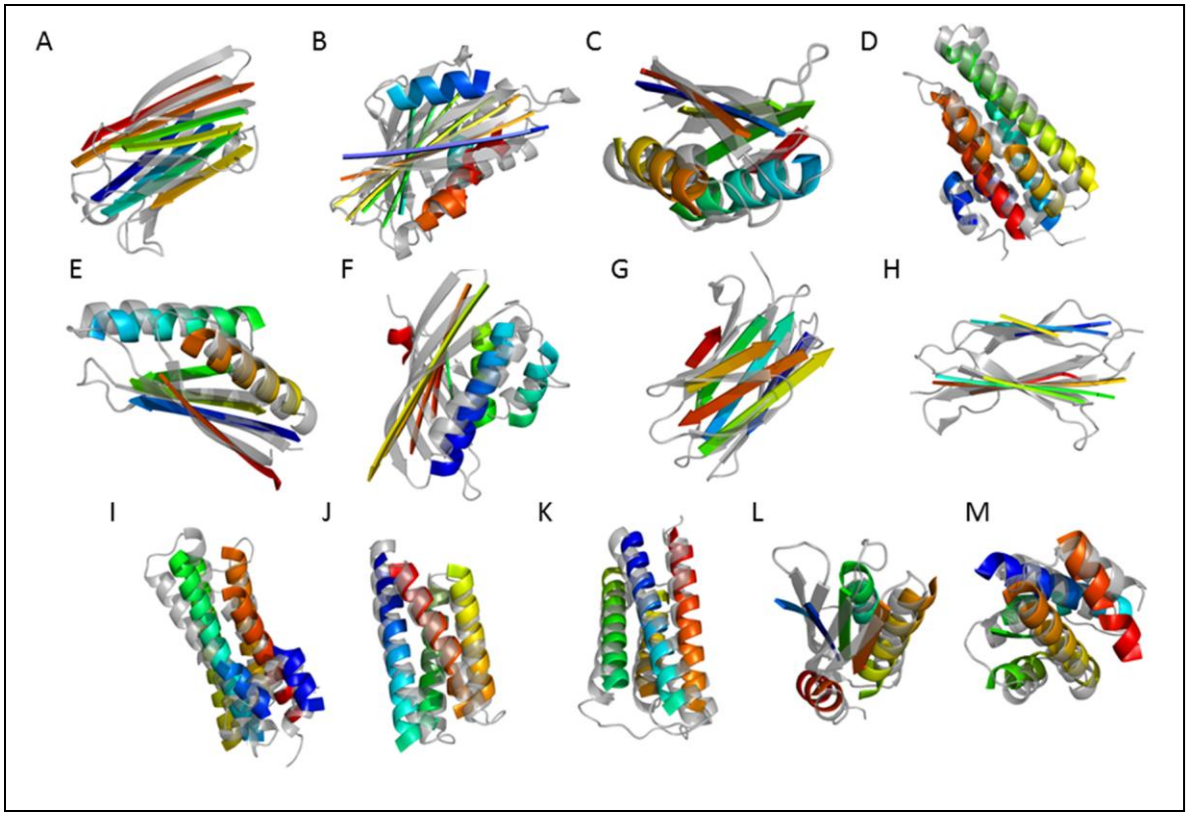

Figure 23: Structures for a selection of best RMSD100 complete models generated by BCL::Fold: BCL::Fold generated best RMSD100 SSE-only models using predicted SSE pool for a selection of proteins. The generated models are rainbow colored and superimposed with the native structure (gray) for following proteins along with the RMSD100 of the models: **(A)** 1GYUA – 4.11Å **(B)** 1ICXA – 6.07Å **(C)** 1ULRA – 3.61Å **(D)** 1X91A –4.08Å **(E)** 1J27A – 3.15Å **(F)** 1TP6A –5.74Å **(G)** 2CWRA -5.66 Å **(H)** 2RB8A –2.91Å **(I)** 1RJ1A –4.86Å **(J)** 1TQGA – 1.92Å **(K)** 2HUJA –2.57Å **(L)** 3OIZA – 5.76Å **(M)** 2V75A –3.11Å

Accurate secondary structure improves quality of BCL::Fold models only slightly

Comparison of BCL::Fold runs with predicted with correct SSEs (Table 11) reveals that using native SSE definitions provides an average improvement of 0.64Å in RMSD100 for SSE only models and 0.40 Å RMSD100 for complete models after loop construction. Although the effect of secondary structure prediction accuracy on average of best

RMSD100 models is modest, this effect is not directly related to Q3 values due to the nature of BCL::Fold assembly protocol. One interesting example is 1LKIA, a 180 residue protein with Q3 values of 75.9 (PSIPRED) and 44.1 (JUFO). Although this protein has a mid-range PSIPRED Q3 value, it exhibits the largest deterioration in both RMSD100 and CR, which is more likely to be explained by the high average shift values; 10.3 residues (PSIPRED) and 16.8 residues (JUFO). Another such example is 1LMIA, which has low Q3 value of 53.7 and 42.5 for PSIPRED and JUFO respectively, accompanied by a low rate correct SSE identification of 67%. On the other hand, if the secondary structure prediction is extremely accurate as in the case of 1TQGA, 1J27A, 3FH2A, 2BK8A (all with PSIPRED Q3 > 94.0), RMSD100 values deteriorate less than <0.3Å when moving from perfect to predicted secondary structure. Although accurate secondary structure prediction improves the overall accuracy of BCL::Fold, the results indicate that it is not a requirement. As described in Table 12, BCL::Fold utilizes a set of moves to dynamically resize and split SSEs during the minimization to compensate for the inaccuracies in secondary structure prediction.

The SSE content (percentage of residues in a sequence that reside in an SSE opposed to coil) versus maximum Q3 value of the pool generated (higher of Q3 values calculated for PSIPRED and JUFO predictions) for each benchmark protein is plotted in Figure 21B. Each point is colored according to the best CR value achieved for that target in complete models generated in BCL::Fold runs using predicted SSE pools (Figure 21B left panel). As observed nearly all targets with highest CR values (>60% colored purple) have ~80% or higher Q3 values, although the SSE content for these targets can range from as little as

40% to as high as 85%. Similar plots are also provided for Rosetta models for comparison (Figure 21B right panel).

BCL::Fold BETA was evaluated in CASP9 experiment

All techniques for protein structure prediction are evaluated every two years via the Critical Assessment of Techniques for Protein Structure Prediction (CASP) experiment [70,71,72]. An early version of BCL::Fold (BCL::Fold BETA) participated in CASP9 and predictions were submitted for 58 of 63 targets given in human predictor category. For each target 50,000 models were generated, the top 10,000 by BCL score were selected for clustering analysis. The five best scoring models as well as the best scoring models in each of the large clusters (~20) underwent loop construction and side chain packing using ROSETTA. The five models for submission were selected from these full atom models as the largest cluster centers. In cases were a template was readily available, the fifth model for submission was the BCL::Fold model with the smallest RMSD to the comparative model built by MODELLER [73]. This approach was chosen to test the BCL::Fold sampling independent from BCL::Score (Chapter III).

Targets in CASP9 were biased towards proteins of known fold. In fact, only 14 out of the 60 human targets had no sequence detectable templates [74]. However, BCL::Fold treated all targets "free modeling (FM)" to maximally leverage the blind CASP experiment to test the algorithm. In cases where a template was available we would not expect to perform better than template-based methods. The remaining few cases represent a too small sample size to comprehensively compare BCL::Fold with other *de novo* protein structure prediction methods, also because of the BETA stage of that version. Therefore we present anecdotal examples where the potential of this early version of the

algorithm became apparent. A more detailed evaluation will be performed during CASP10 in summer 2012.

For FM target T0608_1, the first submission by BCL::Fold had an RMSD of 4.3Å and ranked 9th out of 132 groups (Figure 24). BCL::Fold was also able to produce native-like models and pick them for submission for the following targets; T0580 (105 residues 4.4Å RMSD), T0619 (111 residues 5.9Å RMSD), T0602 (123 residues 7.7Å RMSD), T0630 (132 residues 8.4Å RMSD), T0627 (261 residues 8.9Å RMSD).

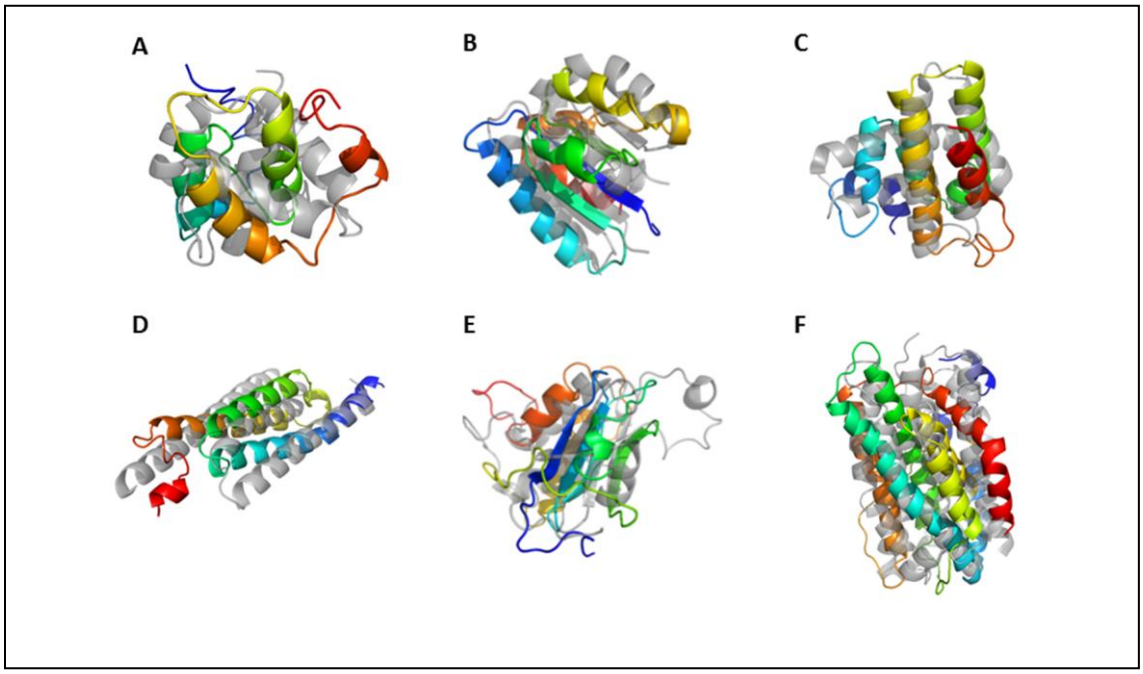

Figure 24: BCL::Fold results from CASP9: The best submitted model out of 5 top submissions by RMSD (rainbow colored) superimposed with the native structure for **(A)** T0608_1 - 89 residues, 4.3Å RMSD **(B)** T0580 - 105 residues 4.44Å RMSD, **(C)** T0619 - 111 residues, 5.86Å RMSD **(D)** T0602 - 123 residues, 7.75Å RMSD **(E)** T0630 - 132 residues, 8.42Å RMSD **(F)** T0627 - 261 residues, 8.90Å RMSD

Conclusion

In conclusion we demonstrate that assembly of SSEs is a viable approach to predict the topology of a protein of unknown fold. BCL::Fold assembles the correct topology for about 3 out of 5 proteins with sequence lengths ranging from 88 residues to 293 residues and 4 to 15 SSEs. BCL::Fold assembly runs range from 1 minute for the smallest protein to 10 minutes for the largest protein with a linear scaling. The impact of predicted versus correct secondary structure is small demonstrating that BCL::Fold can efficiently compensate for inaccuracies in secondary structure prediction. As mentioned above, BCL::Fold currently focuses on topological sampling of SSEs neglecting backbone flexibility within individual SSEs. This leads to increased RMSD100 values especially in β-sheet proteins where despite correct topology, the curvature of β-sheet is not correctly reproduced. With development of more efficient SSE backbone bending strategies BCL::Fold can overcome this limitation.

As expected, BCL::Fold overall performance, in terms of both RMSD100 and CR, is more robust for smaller proteins. There is a linear dependency, more clearly seen with decrease in CR values larger the protein, thus larger the conformational space to be sampled. Out of 31 α-helical proteins tested, BCL::Fold was able generate <8.0Å RMSD100 models for 28 cases (SSE only models) and 15 cases (complete models). Out of 16 β-proteins, this was true for all 16 cases (SSE only models) and 11 cases (complete models). For the remaining 18 αβ proteins, native-like models were generated 16 (SSE only models) and 15 cases (complete models). One of the major reasons of the difficulty experienced with a subset of these targets, as in the case of α-helical proteins 1LKIA, 1Z3XA and 2R0SA, and β-sheet proteins 1LMIA, 2QZQA and 1XAKA, can be

attributed to inaccurate secondary structure predictions in terms of Q3 as well as being unable to identify one or more native SSEs.

As discussed in the introduction, BCL::Fold was designed for combination with limited experimental datasets. A version of BCL::Fold which integrates low resolution restraints from cryo-EM was previously shown to predict the correct topology for α-helical proteins [49]. Incorporation of limited experimental data from NMR and EPR experiments, folding of membrane proteins, and better reproduction of strongly bent SSEs are future directions of our research.

Methods and Materials

BCL::Fold protocol and benchmark analysis

The flowchart of the BCL::Fold protocol is shown in Figure 16. The amino acid sequence and associated secondary structure predictions are utilized to generate a pool of SSEs (Figure 16A). The SSE pool is likely to have multiple copies for the one SSE with varying start and end points. The algorithm then selects one SSE at random from the pool and places it in the origin to start the simulation. The minimization protocol is composed of a Monte Carlo sampling algorithm (Figure 16B) coupled with knowledge-based energy potentials (Figure 16C). Once a specified number of maximum iterations are reached the minimization is ended and the model with the best energy is returned as the final model (Figure 16D). For each of the benchmark proteins, two BCL::Fold runs with 10,000 models each were completed, one using secondary structure definitions provided in the PDB files and one using the secondary structure predictions.

Preparation of benchmark set

The benchmark protein set was collected using PISCES culling server and includes 66 proteins of lengths ranging from 83 to 293 residues with <30% sequence similarity and X-ray determined structures <2.0 Å. The set contains 66 different topologies including 31 all α-helical, 16 all β-strand, and 19 mixed αβ folds (Table 9). The primary sequence and experimental structure of the selected proteins were downloaded from the PDB [75]. The secondary structures were determined using DSSP [69], since the PDB definitions were inconsistent in some places.

Secondary structure prediction and preparation of SSE pool

JUFO [64,65] and PSIPRED [66] were obtained from the authors of the methods and installed locally. In addition the sequence alignment tool BLAST [76,77] was installed locally to create the position specific scoring matrices for input to JUFO and PSIPRED. These are provided as input to the BCL::SSE application which generates a pool of likely SSEs given secondary structure prediction and BLAST profile. BCL::SSE first generates an initial pool by assigning taking the highest probability for each residue and assigning it the corresponding secondary structure type. A threshold of 0.5 is applied for α-helices and β-strands, if the probability is below the threshold; the residue is assigned as a coil even if the highest probability corresponds to α-helix and β-strand. This initial pool is then refined using a Monte-Carlo based minimization composed of 1000 steps. The minimization employs moves that alter the secondary structure assignment of a single residue or divide a SSE, while the energy function used evaluates the correspondence of the secondary structure predictions to the secondary structure assignments generated (see Chapter III for more details). For both the initial pool as well as the final pools generated

by BCL::SSEs, α-helices shorter than 5 residues and β-strands shorter than 3 residues are excluded.

SSE pool evaluation

Q3 is the most commonly used method for evaluating secondary structure assignments [68]. Q3 evaluates the percentage of residues with correct secondary structure assignments. However, since the actual identification of an SSE is more important than individual secondary structure assignments for BCL::Fold, additional measures the percentage of native SSEs that were correctly identified as well as the shift which is sum of deviations in the begin and ends of predicted SSEs compared to native SSEs.

Monte Carlo-based sampling algorithm and temperature control

BCL::Fold starts the minimizations with a structural model that contains a single SSE picked randomly from the pool. At each iteration, a move is selected randomly from the move set and applied to the model to produce a new structural model. The resultant model is evaluated by energy functions, and whether to accept or reject this model is determined by Metropolis criterion[78],

$$p_{accept} = min\left\{1, e^{\frac{-(E_c - E_b)}{kT}}\right\}$$

where E_c is the energy of the current model, E_b is the energy of the best model observed so far, k is a constant and T is the temperature of the system at that point. Temperature is set to 500 initially and adjusted every 10^{th} step to approach an overall cumulative move acceptance ratio for the trajectory. The target ratio for move acceptance is 0.5 in the beginning decreases linearly to 0.2 at the end.

The evaluation of the Metropolis criterion can lead to four different results (Figure 18); (1) skipped, if the mutate was not able to produce a new model, such as when trying to add a new SSE to a model that is already complete, thus the energy evaluation is skipped, (2) improved, if the energy of current model is better than best energy, (3) accepted and (4) rejected if energy of current model is worse than best energy and Metropolis criterion is used for evaluation. If this step is an "improved" state, the current model replaces the best model and minimization is continued with this model. If this step is a "rejected" or "skipped", the minimization is continued with the best model. If this step is an "accepted" state, the minimization is continued on this model however the best model is not updated.

Sampling of conformational search space

The conformation search space is achieved in BCL::Fold by a variety of moves. Each move is assigned a probability and one of them is randomly picked for each step based on these probabilities. The list of all moves utilized, their associated probabilities and descriptions can be found in Table 12 and Table 13. The moves can divided into following six categories; (1) adds, (2) removes, (3) swaps, (4) single SSE moves, (5) SSE-pair moves, (6) domain moves. For SSE, SSE-pair and domain moves, these are further categorized into specific α-helix, β-strands or α-helix domain, β-sheet moves.

Table 12: Moves used in BCL::Fold protocol:

Move	Type	Stage	description
add_sse_next_to_sse	add	A	add an SSE from the pool to the model using preferred orientations
add_sse_short_loop	add	A	add an SSE from the pool next to an SSE which is a neighbor in sequence
add_strand_next_to_sheet	add	A	add a strand to sheet as the edge strand
remove_random	remove	A	remove a randomly determined SSE from the model
remove_unpaired_strand	remove	A	locate and remove an unpaired strand from the model
swap_sse_with_pool	swap	A	swap an SSE in the model with an SSE from the pool
swap_sse_with_pool_overlap	swap	A	swap an SSE in the model with an SSE from the pool which overlaps
swap_sses	swap	A	swap locations of two SSEs in the model

sse_bend_ramachandran	SSE	R	Change phi/psi angles for a random residue using Ramachandran statistics
sse_bend_random_large	SSE	R	Change phi/psi angles for a random residue by 0 to 20 degrees
sse_bend_random_small	SSE	R	Change phi/psi angles for a random residue by 0 to 5 degrees
sse_furthest_move_next	SSE	A	Locate the SSE in the model furthest from the center and re-place it next to another SSE
sse_move_next	SSE	A	Locate a random SSE in the model and re-place it next to another SSE
sse_move_short_loop	SSE	A	Locate a random SSE in the model and re-place it next to an SSE which has a short loop to it
sse_resize	SSE	A + R	Extend/shrink a random SSE by 1 to 3 residues from one end
sse_rotate_large	SSE	A	Rotate an SSE by 15 to 45 degrees in any direction
sse_rotate_x_large	SSE	A	Rotate an SSE by 0 to 45 degrees around X axis
sse_rotate_y_large	SSE	A	Rotate an SSE by up to 45 degrees around Y axis
sse_rotate_z_large	SSE	A	Rotate an SSE by up to 45 degrees around Z axis
sse_rotate_small	SSE	R	Rotate an SSE by up to 15 degrees in any direction
sse_rotate_x_small	SSE	R	Rotate an SSE by up to 15 degrees around X axis
sse_rotate_y_small	SSE	R	Rotate an SSE by up to 15 degrees around Y axis
sse_rotate_z_small	SSE	R	Rotate an SSE by up to 15 degrees around Z axis
sse_split_JUFO	SSE	A	Split a long SSE (>14 residues for helices, > 8 residues for strands) into two shorter SSE by removing the residue in the SSE with the lowest JUFO prediction for the associated SS type
sse_split_PSIPRED	SSE	A	Same as sse_split_JUFO, but uses PSIPRED predictions instead
sse_translate_large	SSE	A	Translate an SSE 2 to 6Å along any direction
sse_translate_x_large	SSE	A	Translate an SSE up to 6Å along X axis
sse_translate_y_large	SSE	A	Translate an SSE up to 6Å along Y axis
sse_translate_z_large	SSE	A	Translate an SSE up to 6Å along Z axis
sse_transform_large	SSE	A	Transform an SSE in any direction by 2 to 6Å translation and 15 to 45 degree rotation
sse_translate_small	SSE	R	Translate an SSE up to 2Å along any direction
sse_translate_x_small	SSE	R	Translate an SSE up to 2Å along X axis
sse_translate_y_small	SSE	R	Translate an SSE up to 2Å along Y axis
sse_translate_z_small	SSE	R	Translate an SSE up to 2Å along Z axis
sse_transform_small	SSE	R	Transform an SSE in any direction by up to 2Å translation and 15 degree rotation
helix_flip_xy	α-helix	A	Rotate a randomly picked helix by 180 degrees around X or Y axis
helix_flip_z	α-helix	A	Rotate a randomly picked helix by 180 degrees around Z axis
helix_furthest_move_next	α-helix	A	Locate the helix in the model furthest from the center and re-place it next to another SSE
helix_move_next	α-helix	A	Locate a random SSE in the model and re-place it next to another SSE
helix_move_short_loop	α-helix	A	Locate a random SSE in the model and re-place it next to an SSE which has a short loop to it
helix_translate_xy_large	α-helix	A	Translate an helix 2 to 4Å along x axis and y axis
helix_translate_z_large	α-helix	A	Translate an helix up to 4Å along z axis
helix_rotate_xy_large	α-helix	A	Rotate an helix 15 to 45 degrees around x axis and y axis
helix_rotate_z_large	α-helix	A	Rotate an helix 15 to 45 degrees around z axis
helix_transform_xy_large	α-helix	A	Transform a helix by 2 to 4A translation and 15 to 45 degrees rotation in x axis and y axis
helix_transform_z_large	α-helix	A	Transform a helix by 2 to 4A translation and 15 to 45 degrees rotation in z axis
helix_translate_xy_small	α-helix	R	Translate an helix up to 2Å along x axis and up to 2Å along y axis
helix_translate_z_small	α-helix	R	Translate an helix up to 2Å along z axis
helix_rotate_xy_small	α-helix	R	Rotate an helix up to 15 degrees around x axis and up to 15 degrees around y axis
helix_rotate_z_small	α-helix	R	Rotate an helix up to 15 degrees around z axis
helix_transform_xy_small	α-helix	R	Transform a helix by up to 2A translation and up to 15 degrees rotation in z axis
helix_transform_z_small	α-helix	R	Transform a helix by up to 2A translation and up to 15 degrees rotation in z axis
strand_flip_x	β-strand	A	Rotate a randomly picked strand by 180 degrees around X axis
strand_flip_y	β-strand	A	Rotate a randomly picked strand by 180 degrees around Y axis
strand_flip_z	β-strand	A	Rotate a randomly picked strand by 180 degrees around Z axis
strand_furthest_move_next	β-strand	A	Locate the strand in the model furthest from the center and re-place it next to another SSE
strand_furthest_move_sheet	β-strand	A	Locate the strand in the model furthest from the center and re-place it next to a sheet
strand_move_next	β-strand	A	Locate a random strand in the model and re-place it next to another SSE
strand_move_sheet	β-strand	A	Locate a random strand in the model and re-place it next to a sheet
strand_translate_z_large	β-strand	A	Translate a strand up to 2Å along z axis
strand_translate_z_small	β-strand	R	Translate a strand 2 to 4Å along z axis

ssepair_translate_large	SSE pair	A	Locate two packed SSEs, translate one of them 1 to 3Å along the packing axis
ssepair_translate_no_hinge_large	SSE pair	A	Locate two packed SSEs, translate one of them 2 to 4Å in any axis of the other one
ssepair_rotate_large	SSE pair	A	Locate two packed SSEs, rotate one of them 10 to 45 degrees around the packing axis
ssepair_transform_large	SSE pair	A	Locate two packed SSEs, transform one of them using the packing axis by 1 to 3Å translation and 10 to 45 degrees rotation
ssepair_translate_small	SSE pair	R	Locate two packed SSEs, translate one of them up to 3Å along the packing axis
ssepair_translate_no_hinge_small	SSE pair	R	Locate two packed SSEs, translate one them up to 2Å in any axis of the other one
ssepair_rotate_small	SSE pair	R	Locate two packed SSEs, rotate one of them up to 15 degrees around the packing axis
ssepair_transform_small	SSE pair	R	Locate two packed SSEs, transform one of them using the packing axis up to 1Å translation and up to 15 degrees rotation
helixpair_rotate_z_large_hinge	α-pair	A	Locate two packed helices, rotate both 15 to 45 degrees around z axis of one of them
helixpair_rotate_z_large_no_hinge	α-pair	A	Locate two packed helices, rotate one 15 to 45 degrees around z axis of the other one
helixpair_rotate_z_small_hinge	α-pair	R	Locate two packed helices, rotate both up to 15 degrees around z axis of one of them
helixpair_rotate_z_small_no_hinge	α-pair	R	Locate two packed helices, rotate one up to 15 degrees around z axis of the other one
helixdomain_flip_ext	α-domain	A	Locate a domain of helices, rotate them 180 degrees externally along a common x,y or z axis
helixdomain_flip_int	α-domain	A	Locate a domain of helices, rotate them 180 degrees internally along x,y or z axis
helixdomain_shuffle	α-domain	A	Locate a domain of helices, swap locations of 1 or 2 pairs of helices
helixdomain_translate_large	α-domain	A	Translate a domain of helices 2 to 6Å along any direction
helixdomain_rotate_large	α-domain	A	Rotate a domain of helices 15 to 45 degrees along any axis
helixdomain_transform_large	α-domain	A	Transform a domain of helices by 2 to 6Å translation and 15 to 45 degrees rotation along any axis
helixdomain_translate_small	α-domain	R	Translate a domain of helices up to 2Å along any direction
helixdomain_rotate_small	α-domain	R	Rotate a domain of helices up to 15 degrees along any axis
helixdomain_transform_small	α-domain	R	Transform a domain of helices by up to 2Å translation and up to 30 degrees rotation
sheet_shuffle	β-sheet	A	Locate a sheet, swap locations of 1 or 2 pairs of strands
sheet_switch_strand	β-sheet	A	Remove a edge strand from a sheet and add it to another sheet
sheet_cycle	β-sheet	A	Locate a sheet, cycle the locations of 2 to 4 strands in the sheet by 1 to 3 positions
sheet_cycle_intact	β-sheet	A	Locate a sheet, cycle the locations of all strands in the sheet by 1 to 3 positions , while keeping relative parallel/antiparallel orientations intact
sheet_cycle_subset	β-sheet	A	Same as sheet_cycle, but instead of all strands, only moves 2 to 4 strands
sheet_cycle_subset_intact	β-sheet	A	Same as sheet_cyle_subset, but keeps the relative parallel/antiparallel orientations intact
sheet_divide	β-sheet	A	Locate a sheet of at least 4 strands and divide it to two sheets of at least 2 strands each and then translate one sheet away from up to 4Å in each direction
sheet_divide_sandwich	β-sheet	A	Locate a sheet of at least 4 strands and divide it to two sheets of at least 2 strands each and then pack one of the new sheets against the other one in beta-sandwich form
sheet_flip_ext	β-sheet	A	Rotate all strands in a sheet externally along a common x, y or z axis
sheet_flip_int	β-sheet	A	Rotate all strands in a sheet internally along x, y or z axis
sheet_flip_int_sub	β-sheet	A	Rotate a subset of strands in a sheet internally along x,y or z axis
sheet_flip_int_sub_diff	β-sheet	A	Rotate a subset of strands in a sheet along different axes
sheet_pair_strands	β-sheet	A	Locate unpaired strands and pair them with each other, if there is only one unpaired strand, then add it to a sheet
sheet_register_fix	β-sheet	R	Fix the hydrogen bonding pattern of a located sheet by applying small translations
sheet_register_shift	β-sheet	A	Shift the hydrogen bonding register of two strands in a sheet by a translation in the amoun of two residue lengths
sheet_register_shift_flip	β-sheet	A	Shift the hydrogen bonding register of two strands in a sheet by a translation in the amount of one residue length coupled with a 180 degrees rotation around x or y axis
sheet_translate_large	β-sheet	A	Translate a sheet by 2 to 4Å along any axis
sheet_rotate_large	β-sheet	A	Rotate a sheet by 15 to 45 degrees around any axis
sheet_transform_large	β-sheet	A	Transform a sheet by 2 to 4Å translation and 15 to 45 degreess rotation

sheet_twist_large	β-sheet	A	Adjust the twist angle of all strands in a sheet by up to 10 degrees rotations
sheet_translate_small	β-sheet	R	Translate a sheet by up to 2 Å along any axis
sheet_rotate_small	β-sheet	R	Rotate a sheet by up to 15 degrees around any axis
sheet_transform_small	β-sheet	R	Transform a sheet by up to 2 Å translation and up to 15 degrees rotation
sheet_twist_small	β-sheet	R	Adjust the twist angle of all strands in a sheet by up to 2 degrees rotations
total	TOTAL		

All moves used in BCL::Fold are listed along with the subcategory they belong to and whether they are utilized in assembly (A) or refinement (R) stage. The last column gives a short description of what each move does.

Table 13: Statistics for the moves used in BCL::Fold protocol:

Move	Type	Stage	$\%_{improved}$	$\%_{accepted}$	$\%_{rejected}$	$\%_{skipped}$	Δ_{mean}
add_sse_next_to_sse	add	A	1.7	4.3	8.8	85.2	-392.88
add_sse_short_loop	add	A	2.4	4.5	7.1	85.9	-401.83
add_strand_next_to_sheet	add	A	2.0	2.0	2.0	94.0	-458.46
remove_random	remove	A	0.2	16.5	82.8	0.5	-236.96
remove_unpaired_strand	remove	A	0.1	6.9	11.4	81.6	-220.60
swap_sse_with_pool	swap	A	1.0	4.3	5.1	89.6	-241.83
swap_sse_with_pool_overlap	swap	A	4.0	37.1	58.0	0.9	-126.58
swap_sses	swap	A	0.8	18.3	78.6	2.3	-208.59
sse_bend_ramachandran	SSE	R	7.8	19.4	72.8	0.0	-21.85
sse_bend_random_large	SSE	R	7.8	23.0	69.2	0.0	-27.76
sse_bend_random_small	SSE	R	20.3	36.9	42.8	0.0	-20.18
sse_furthest_move_next	SSE	A	1.1	15.0	84.0	0.0	-289.20
sse_move_next	SSE	A	0.5	11.0	88.5	0.0	-264.67
sse_move_short_loop	SSE	A	0.8	11.5	76.1	11.7	-276.90
sse_resize	SSE	A + R	14.7	28.2	45.2	11.9	-106.50
sse_rotate_large	SSE	A	1.4	20.2	78.5	0.0	-98.23
sse_rotate_x_large	SSE	A	2.4	23.1	74.4	0.0	-79.79
sse_rotate_y_large	SSE	A	4.0	28.5	67.5	0.0	-126.57
sse_rotate_z_large	SSE	A	9.1	47.3	43.6	0.0	-40.11
sse_rotate_small	SSE	R	3.3	17.0	79.7	0.0	-33.28
sse_rotate_x_small	SSE	R	7.5	23.5	69.0	0.0	-20.90
sse_rotate_y_small	SSE	R	10.2	26.6	63.1	0.0	-27.47
sse_rotate_z_small	SSE	R	17.9	42.3	39.8	0.0	-11.22
sse_split_JUFO	SSE	A	1.8	24.2	69.3	4.7	-88.80
sse_split_PSIPRED	SSE	A	2.1	24.6	68.5	4.8	-84.75
sse_translate_large	SSE	A	0.6	16.6	82.7	0.0	-148.39
sse_translate_x_large	SSE	A	2.1	27.1	70.9	0.0	-106.53
sse_translate_y_large	SSE	A	1.7	21.5	76.8	0.0	-110.62
sse_translate_z_large	SSE	A	7.4	45.6	47.0	0.0	-59.38
sse_transform_large	SSE	A	0.4	14.1	85.5	0.0	-136.66
sse_translate_small	SSE	R	3.0	18.1	78.9	0.0	-50.03
sse_translate_x_small	SSE	R	12.7	31.9	55.4	0.0	-18.13
sse_translate_y_small	SSE	R	9.2	27.0	63.8	0.0	-30.08
sse_translate_z_small	SSE	R	15.0	41.8	43.2	0.0	-7.62
sse_transform_small	SSE	R	1.1	11.8	87.1	0.0	-45.24
helix_flip_xy	α-helix	A	2.8	32.9	64.0	0.3	-132.11
helix_flip_z	α-helix	A	3.7	40.8	55.2	0.3	-109.61
helix_furthest_move_next	α-helix	A	1.1	15.5	83.2	0.3	-295.12
helix_move_next	α-helix	A	0.6	12.6	86.6	0.3	-274.89
helix_move_short_loop	α-helix	A	0.9	13.4	73.4	12.3	-278.55
helix_translate_xy_large	α-helix	A	1.6	26.7	71.4	0.3	-128.36
helix_translate_z_large	α-helix	A	8.4	46.9	44.5	0.3	-59.78
helix_rotate_xy_large	α-helix	A	2.0	26.4	71.3	0.3	-91.11
helix_rotate_z_large	α-helix	A	13.7	53.1	33.0	0.3	-40.98
helix_transform_xy_large	α-helix	A	0.9	21.0	77.8	0.3	-123.62
helix_transform_z_large	α-helix	A	4.5	38.4	56.9	0.3	-88.31

helix_translate_xy_small	α-helix	R	4.8	30.3	64.8	0.1	-17.50
helix_translate_z_small	α-helix	R	16.1	46.6	37.2	0.1	-8.01
helix_rotate_xy_small	α-helix	R	5.0	26.2	68.8	0.1	-23.11
helix_rotate_z_small	α-helix	R	18.4	51.0	30.5	0.1	-7.01
helix_transform_xy_small	α-helix	R	2.3	20.3	77.3	0.1	-31.73
helix_transform_z_small	α-helix	R	9.4	40.4	50.1	0.1	-12.49
strand_flip_x	β-strand	A	1.6	26.6	69.8	1.9	-180.72
strand_flip_y	β-strand	A	1.5	26.1	70.5	2.0	-188.09
strand_flip_z	β-strand	A	8.8	53.7	35.5	2.0	-34.21
strand_furthest_move_next	β-strand	A	0.7	11.7	85.7	1.9	-237.35
strand_furthest_move_sheet	β-strand	A	1.5	17.6	66.6	14.3	-310.50
strand_move_next	β-strand	A	0.4	8.7	89.0	1.9	-232.05
strand_move_sheet	β-strand	A	0.7	12.7	72.3	14.3	-257.51
strand_translate_z_large	β-strand	A	9.7	47.4	41.0	2.0	-48.79
strand_translate_z_small	β-strand	R	13.1	35.1	50.7	1.1	-7.86
ssepair_translate_large	SSE pair	A	1.1	12.3	25.7	60.9	-101.64
ssepair_translate_no_hinge_large	SSE pair	A	0.3	7.2	31.7	60.8	-155.56
ssepair_rotate_large	SSE pair	A	1.3	10.3	27.4	61.0	-91.34
ssepair_transform_large	SSE pair	A	0.4	7.8	30.8	61.0	-132.55
ssepair_translate_small	SSE pair	R	4.3	14.6	22.9	58.2	-19.08
ssepair_translate_no_hinge_small	SSE pair	R	0.9	7.7	33.3	58.1	-33.95
ssepair_rotate_small	SSE pair	R	2.9	10.7	28.2	58.1	-17.29
ssepair_transform_small	SSE pair	R	1.8	10.2	29.9	58.2	-24.52
helixpair_rotate_z_large_hinge	α-pair	A	1.1	19.6	66.4	12.9	-146.72
helixpair_rotate_z_large_no_hinge	α-pair	A	1.2	19.9	66.0	12.9	-143.44
helixpair_rotate_z_small_hinge	α-pair	R	4.2	26.0	60.4	9.4	-11.85
helixpair_rotate_z_small_no_hinge	α-pair	R	4.5	26.3	59.7	9.4	-11.54
helixdomain_flip_ext	α-domain	A	0.1	3.8	18.9	77.2	-192.21
helixdomain_flip_int	α-domain	A	0.2	5.6	16.7	77.5	-137.44
helixdomain_shuffle	α-domain	A	0.4	16.4	82.0	1.2	-259.28
helixdomain_translate_large	α-domain	A	0.3	13.5	85.1	1.2	-186.58
helixdomain_rotate_large	α-domain	A	0.2	9.9	88.8	1.1	-140.06
helixdomain_transform_large	α-domain	A	0.1	8.6	90.1	1.2	-156.38
helixdomain_translate_small	α-domain	R	1.0	17.6	81.4	0.1	-30.96
helixdomain_rotate_small	α-domain	R	0.5	9.2	90.3	0.1	-37.29
helixdomain_transform_small	α-domain	R	0.0	3.2	96.7	0.1	-59.14
sheet_shuffle	β-sheet	A	1.0	17.1	75.8	6.1	-192.62
sheet_switch_strand	β-sheet	A	0.9	7.5	27.4	64.1	-380.80
sheet_cycle	β-sheet	A	0.5	12.3	68.5	18.7	-256.54
sheet_cycle_intact	β-sheet	A	0.5	11.8	69.2	18.5	-225.13
sheet_cycle_subset	β-sheet	A	0.7	28.3	52.6	18.4	-182.83
sheet_cycle_subset_intact	β-sheet	A	0.7	27.8	52.7	18.7	-175.25
sheet_divide	β-sheet	A	0.7	8.6	54.5	36.2	-154.32
sheet_divide_sandwich	β-sheet	A	0.2	3.3	60.1	36.5	-371.36
sheet_flip_ext	β-sheet	A	0.7	41.6	51.7	6.1	-147.15
sheet_flip_int	β-sheet	A	1.4	24.5	67.9	6.2	-102.42
sheet_flip_int_sub	β-sheet	A	2.1	25.4	66.4	6.2	-90.10
sheet_flip_int_sub_diff	β-sheet	A	1.4	20.0	72.6	6.1	-128.56
sheet_pair_strands	β-sheet	A	0.8	2.7	4.6	91.9	-457.81
sheet_register_fix	β-sheet	R	1.0	13.0	66.6	19.4	-23.22
sheet_register_shift	β-sheet	A	1.7	25.7	53.9	18.7	-83.74
sheet_register_shift_flip	β-sheet	A	3.4	33.6	44.4	18.5	-71.42
sheet_translate_large	β-sheet	A	1.0	42.7	55.9	0.5	-139.46
sheet_rotate_large	β-sheet	A	0.6	38.5	60.4	0.5	-99.94
sheet_transform_large	β-sheet	A	0.4	37.7	61.4	0.6	-109.11
sheet_twist_large	β-sheet	A	7.5	26.9	47.0	18.6	-128.74
sheet_translate_small	β-sheet	R	1.6	43.9	54.4	0.1	-33.51
sheet_rotate_small	β-sheet	R	0.9	38.3	60.7	0.1	-53.27
sheet_transform_small	β-sheet	R	0.4	36.2	63.4	0.1	-71.91
sheet_twist_small	β-sheet	R	10.4	21.9	48.3	19.4	-27.86
total	TOTAL		2.7	19.6	59.1	18.6	-73.74

All moves used in BCL::Fold are listed along with the subcategory they belong to and whether they are utilized in assembly (A) or refinement (R) stage. This is followed by percentages on minimization steps where each move was used along with what kind of Metropolis result these steps have led to; percentage of improved steps(P_I), accepted steps (P_A), rejected steps (P_R), skipped steps(P_S). This is followed by Δ_{MEAN},

which represents the average energy decrease in the energy from the last improved model for cases where the move has led to an improved step.

Loop building

Missing loop residues were built on to the model predicted by BCL::Fold using an in-house CCD based loop building protocol [38]. The protocol first removes a single residue from each side of all the SSEs in the model to increase the chance of being able to close the loop. Then, missing loop residues are added to the model with phi/psi angles biased by Ramachandran distribution for given amino acid type. The initial conformations of the residues are optimized using BCL scoring functions including amino acid clash and amino acid environment and a bias to close the chain breaks. This step ensures that initial positions can be found for all residues without causing any clashes. In the next stage, a CCD-based minimization is applied to ensure all loops are closed.

Composite knowledge-based energy function

The composite energy function is described in detail in Chapter III. In brief, the energy functions consists of eleven individual terms for (1) amino acid pair distance clash, (2) amino acid pair distance, (3) amino acid solvation, (4) SSE pair clash, (5) SSE pair packing, (6) β-strand pairing, (7) loop length, (8) strictly enforcing loop closure, (9) radius of gyration, (10) SSE prediction for JUFO (11) SSE prediction for PSIPRED and lastly (12) contact order. The scores for amino acid solvation and SSE predictions also come with entropic counterparts which evaluate all the residues not represented in the model, using the corresponding potentials. All scoring functions are implemented within the BCL.

All knowledge based potentials have been derived from a databank that contained 3,409 high resolution x-ray crystallography protein structures compiled using the PISCES server [79]. The collected statistical representations are converted into a free energy using the inverse Boltzmann relation and applying the appropriate normalizations. The weights for individual energy functions were optimized using a benchmark of models composed of *de novo* folded models by Rosetta [28], BCL::Fold as well as perturbed models of native structures generated by perturbation protocol within BCL. The finalized weights for energy functions used can be found in Table 14.

Table 14: Weight set for the energy function in BCL::Fold:

energy function	weight
aa_clash	500.0
aa_dist	0.4
aa_neigh	50.0
aa_neigh_ent	50.0
sse_clash	500.0
sse_pack	8.0
strand_pair	20.0
loop	10.0
loop_closure	500.0
rgyr	5.0
co	0.5
ss_JUFO*	5.0
ss_JUFO_ent*	5.0
ss_PSIPRED*	20.0
ss_SIPRED_ent*	20.0
entropy	1.0

Following scores were used in the energy function in BCL::Fold; amino acid clash score (aa_clash), amino acid distance score (aa_dist), amino acid environment potential and entropic counterpart (aa_neigh & aaneigh_ent), SSE clash score (sse_clash), SSE packing score (sse_pack), β-strand pairing score (strand_pair), loop score (loop), loop closure score (loop_closure), radius of gyration score (rgyr), contact order score (co) contact order score, SSE prediction scores and their entropic counterparts using methods JUFO (sse_JUFO && ss_JUFO_ent) and PSIPRED (ss_PSIPRED && ss_PSIPRED_ent).

Benchmark analysis

For each BCL::Fold run of 10,000 models for each of the 66 proteins in the benchmark set, an initial filtering is done to remove any incomplete models. The models produced by BCL::Fold benchmarks are evaluated by looking at following quality measures root-mean-square-deviation (RMSD), RMSD100 [80] and CR. These measures are calculated over Cα atoms of all the residues in α-helices and β-strands in the models. In addition, contact order [43] values were calculated by computing average sequence separation of contacts defined as having C_β ($H_{\alpha 2}$ for Glycine) atoms within 8Å distance. Relative contact order (RCO) values were calculated by normalizing contact order values by the length of the sequence. Normalized contact orders were calculated by dividing the square of the contact order by the length of the sequence. An additional quality measure was developed named contact recovery (CR) which evaluates the percentage of native contacts with a minimal sequence separation of 12 residues that are recovered in the models.

Protein structure prediction using Rosetta

Rosetta[28] protein structure prediction program was used to predict 10,000 models for each of the benchmark proteins in order to provide a comparison for analysis of BCL::Fold. The models were produced using *de novo* mode of Rosetta, and fragment files provided as input to Rosetta were pre-filtered to remove any fragments for homologous proteins. The resultant models underwent the same analysis as the models produced by BCL::Fold. Secondary structures in Rosetta models was determined using DSSP[69] and the quality calculations were completed considering C_α atoms from identified α-helices and β-strands where applicable.

BCL::Fold availability

All components of BCL::Fold, including scoring, sampling, and clustering methods are implemented as part of the BioChemical Library (BCL) that is currently being developed in the Meiler laboratory (www.meilerlab.org). BCL BCL::Fold will be freely available for academic use along with several other components of BCL library via BCLCommons (http://bclcommons.vueinnovations.com/bclcommons). In the meantime, an executable can be obtained by contacting the authors.

CHAPTER V

DISCUSSION

This thesis work focused on two projects; development of a residue-residue contact prediction method using artificial neural networks and development of a *de novo* protein structure prediction method that relies on assembly of secondary structure elements. The following sections will provide insights into what was achieved, how projects evolved through their courses and how both of these methods can be improved by future work.

BCL::Contact

A structure-based contact prediction method was previously developed by Dr. Jens Meiler before the start of this project and was one of the top ranking methods at the CASP6 competition. The aim was to develop a sequence-based method, which did not rely on any structure-information, in order to predict contacts in a more rapid fashion without any external dependence. BCL::Contact achieved this with a novel approach to contact prediction by using a combination of five neural networks, each specialized for a specific contact type in their training.

The first aim for BCL::Contact was participation at the 7^{th} round of CASP competition. When competing with structure-based methods, the expectation was not to perform better than other methods for all targets, but instead focus on a few targets where there are no available template sequences with determined structures which most structure-based methods rely on. Nonetheless, BCL::Contact was able to predict long distance residue

contact in CASP7 with up to 40% accuracy. The highlight of CASP7 for BCL::Contact was target T0356_3 where it was able to rank 4th with an accuracy of 20.8% and coverage of 14.8%.

The main motivation for development of BCL::Contact was, as mentioned, to be able to very rapidly get contact predictions for a given amino acid sequence so that the prediction can be provided as restraints to *de novo* protein structure prediction. In order to assess this, Rosetta was used to predict structural models, and then the runs were repeated twice once with predictions from the BCL::Contact sequence-based method and once with predictions from BCL::Contact structure-based method used as additional restraints. The comparison of the RMSD and MAXN% distributions of the generated models revealed that with both methods significant improvements were observed when compared to Rosetta runs with no additional input.

The next question for BCL::Contact is whether the accuracy can be improved via integration of correlated mutations. As mentioned, correlated mutations can be crucial for contact prediction, since when multiple sequence alignments are examined, it is very likely to find pairs of amino acids that are in contact to be mutated at the same time from one sequence to another sequence. The reasoning behind this is that if a certain residue is mutated through evolution, the other residue/s that are in contact in three dimensional space with it are also very likely to mutate in order to compensate and preserve structural integrity. Multiple sequence alignment is already possible in the BCL library via BCL::Align. The next stage would be implementation of algorithms that extract correlated mutation information from these alignments and convert them to be additional descriptors to be used in neural network training.

In addition to correlated mutations, BCL::Contact can also gain from addition of pairwise descriptors. Currently all descriptors used for describing each amino acid in sequence windows rely solely on the type of that amino acid. Pairwise descriptors on the other hand rely on both amino acid identities for which the contact probability is to be predicted. The statistics for amino acid pair contact potential are already available in BCL as histograms used for amino acid distance pair score as part of BCL::Fold.

Currently BCL::Contact does not differentiate between the orientations of the sequence given to neural networks, meaning both in training and prediction, all sequence windows have residues ordered in the N to C terminal direction. However when sequence stretches for two α-helices that pack against each other are considered, whether they pack parallel or anti-parallel would make a difference when using a window representation for description generation for neural networks. In certain cases, only one of the orientations might be possible due to large side chain clashes between residues. If this information is reflected in the order of the descriptors, it can lead to a decreased noise in training and increase the accuracy of prediction.

BCL::Score

The main motivation of the BCL::Score project was to develop robust energy potentials that could be used with BCL::Fold and any other *de novo* protein structure prediction method that relies on assembly of SSEs. The current set of knowledge-based energy potentials were able to achieve the goal of discriminating native-like topologies, as indicated by the enrichment values and also the impressive results from BCL::Fold

benchmarks. Each individual term as well as the sum function was assessed for their enrichment values on a large set of decoys composed of Rosetta *de novo* folded, BCL::Fold *de novo* folded and BCL::Fold perturbed structures for 53 benchmark proteins. The enrichments were calculated in order to discriminate good models by RMSD100 (<8Å) and GDT_TS (>25%) measures. Although all scores were able to achieve enrichments for large subsets of the benchmark proteins, depending on the nature of the potential, some were not applicable to models generated by one or more of the three methods used. For example, the loop closure score was not able to enrich for Rosetta models since the loop residues are already explicitly modeled and therefore rendering loop closure score inapplicable.

Another great aspect of energy terms that are part of BCL::Score is their efficiency. Special attention was given to develop frameworks that would allow caching of scores to ensure they are only recalculated when needed and only for the entities in the model that have been changed. My personal opinion is that BCL::Score as it is now, encapsulates nearly all the functionality needed for coupling with *de novo* protein structure prediction. In accordance, the improvements to BCL::Score in the future should be focused on further weight optimization for the score sum function. The main concern regarding the weight optimization is based on the following question. Can one scoring weight set be optimal for driving *de novo* structure prediction and at the same time for filtering generated models? As observed over last six years, especially in the case of BCL::Fold where special measures have to be taken to push models to completion, the answer to this question is, in my opinion, no.

Optimizing scoring weight sets specifically for driving protein structure prediction is possible through iterative folding runs where at each run the weights are adjusted and at the end of each stage, the quality of the models generated are used to determine the changes that would need to be applied in the next round of folding runs. Currently there is no such framework in BCL specifically for this purpose, although similar processes are used in descriptor selection when training support vector machines and artificial neural networks.

BCL::Fold

BCL::Fold was benchmarked using 66 proteins with diverse topologies, SSE contents, varying sequence lengths in the range of 88 to 293 amino acids and a RCO range of 0.12 to 0.47 with an average of 0.30 ± 0.07. 10,000 SSE-only models were generated for each of the 64 proteins using the native SSE pool and then runs were repeated using predicted SSE pools. For all models, loops were completed using an in-house loop building protocol. The results have shown that BCL::Fold, despite being at an early stage, was able to sample models below 8Å RMSD100 for 50 proteins (75%) when using native SSE definitions and impressively for 41 proteins (62%) when using predicted SSE pools. When SSE only models are considered, the correct topology was found for 61 proteins (92%) using native SSE definitions and for 61 proteins (92%) using predicted SSE pools. Further detailed analysis of results can be found in Chapter IV.

These results are very promising for BCL::Fold and its novel approach to *de novo* protein structure prediction. Further investigations into the types of proteins BCL::Fold was

unable to capture the correct topology, or not frequently enough, would likely reveal new ways for improvement. There are already a few approaches that are currently in progress to improve the sampling capability and the accuracy of BCL::Fold. One major project, as headed by Dr. Brian Weiner, is introduction of templates. However, unlike other template-based approaches, this approach uses sequence-independent templates for describing a "fold"; the full set of geometries of SSEs as well as their locations with respect to each other. Just like the novel sampling approach used in BCL::Fold, the exclusion of loop residues provides additional opportunities for template-based modeling such as easy and rapid mixing of templates to produce novel templates. There has been significant progress in this project and this method is currently being benchmarked to be published before the end of 2011.

I have implemented a set moves and classes in BCL::Fold for using a similar template approach for sampling backbone conformations for β-sheets. As mentioned in Chapter IV in the results section, BCL::Fold does a great job in sampling different topologies defined by the different ordering and orientations of β-strands within a β-sheet. However, at certain models generated by BCL::Fold, the RMSD values can be higher than expected even when the topology is correct due to the curvature of the β-sheet, which can be extreme in certain cases. BCL::Fold currently has few moves for sampling backbone conformations; however these moves change phi/psi angles of only a single residue for a single SSE at a time. In order to go from a β-sheet conformation formed of relatively idealized β-strands to a conformation where the β-sheet is curved, a significant number of such moves would have to be applied in a row. Moreover, it is likely that certain phi/psi changes would lead to clashes, causing the moves to be rejected. Instead, these changes

can be done in a coordinated way through use of β-sheet templates which contain relative and internal geometries including curvature for each strand in a β-sheet. Such templates are currently collected from the previously described non-redundantly culled ~4000 protein set and are categorized by the total number of strands. A corresponding move is already implemented in BCL::Fold. This move first locates a β-sheet within the model, and then picks a sheet template randomly out of all the templates which have the same number of strands as the β-sheet in the model. Following this, strands from β-sheet in the model are fitted geometrically one by one to the strands in the picked sheet-template. Since the sheet templates do not store the actual sequence nor the sequence order for the strands; the correspondence between strands in the β-sheet and the sheet template are determined by the order of the strands in the sheet geometry, meaning starting from one edge strand and iterating until the other edge strand of both sheets. Although they are sequence-independent, these templates and the corresponding moves allow a rapid sampling of different sheet curvatures and in return increase the accuracy of the predicted models. The effect of sheet-templates in BCL::Fold will also be assessed as part of the fold templates project.

For sampling backbone conformations of α-helices, a similar approach could be utilized. Again, instead of just relying on single residue phi/psi change moves, the geometry of an α-helix randomly picked from the fold templates library could be applied. The fragment assembly methods such as Rosetta have shown this to be an efficient way for sampling backbone flexibility of α-helices with the difference that Rosetta and similar methods rely on sequence-dependent templates whereas this can be completely sequence-independent in BCL::Fold.

One other area that could be improved in BCL::Fold concerns the selection of moves to apply during minimization. All moves have individual weights that determine how likely they are to be picked. These weights are normalized depending on the composition of the SSE pool before the minimization and they are kept constant throughout the minimization. As a direct result of this, a certain fraction of moves that are picked lead to skipped steps because they are not applicable (e.g. "add" moves being picked even when the model is complete). This can be resolved either through a dynamic adjustment of weights for the moves based on their acceptance ratios or turning certain moves on or off based on the composition of the model.

In summary, BCL::Fold was successfully developed and benchmarked as a method for *de novo* protein structure prediction. More importantly, due to the extreme attention paid to design, modularity, efficiency as well as expandability, it is currently serving as a framework for a large number of new methods currently being developed in the Meiler Laboratory for integrating a variety of experimental constraints/restraints for protein structure prediction. BCL::Fold being a product of only six years of work, it can be considered a young method especially when compared to other methods which have been around for nearly one to two decades. However, I am very hopeful that BCL::Fold will be an essential tool for protein structure prediction and will become widely adopted in the upcoming years.

APPENDIX

General Comments

Work presented in all three chapters of this dissertation corresponds to applications developed in BCL. The latest release of BCL library can be accessed by executing `bcl.exe` on the command line. If a specific SVN repository revision number is provided, as in the case of BCL::Fold, then the user would need to check out the corresponding revision from BCL SVN repository and compile it.

This appendix is accompanied with a DVD where files are categorized into two subfolders; contact/ and fold/. In each folder and subfolders, you can find a text file named "readme.txt" which contains information regarding files in that directory.

The information given in the appendix and in the readme.txt files on DVD consists of BCL applications and a combination of Perl and awk scripts. For any BCL application, the help of the application should have all necessary information. For Perl scripts, calling the script without any arguments or with "–h" flag provides the list of required arguments and a detailed description of all arguments where available. This appendix is provided as a guide to the applications and scripts used for the research described in previous chapters. It is expected that, especially regarding BCL::Fold, the arguments and the performance of the application will change over time.

BCL::Contact

Artificial neural networks trained, as explained in Chapter II, were converted to C++ classes and are part of BCL library. Contact prediction application is a released application of BCL library and is also accessible through servers on Meiler lab website (www.meilerlab.org).

BCL::Contact can be accessed by calling "bcl.exe ContactPrediction" on the command line. The application supports a single fasta file with flag "input_filename" or a batch of filenames with flag "pdb_list". The application requires the existence of Blast profile files with extension ".ascii" and the same prefix as the given fasta file. The program produces a ".contact" file which includes for each residue pair, 6 values composed of predictions from helix-helix, helix-strand, strand-helix, strand-strand and sheet-sheet ANNs as well as a last value which provides the merged prediction. Flag "threshold" can be used to remove low probability predictions, while "real_contacts" flag provides the list of native contact if a PDB file was provided. The output files are written to the working directory unless a specific directory is specified with "output_path" flag. Following is sample command line that could be run in contact/ folder. "bcl.exe ContactPrediction 1UBIA.fasta –threshold 0.5".

BCL::Fold

BCL::Fold application can be accessed by calling "bcl.exe Fold". Due to the extensive nature of flags and modes of this application, detailed information is provided in fold/ folder at "readme.txt" file. In addition to the BCL application, a collection of Perl/awk

scripts are used to generate directories for benchmarking, corresponding PBS jobs and submission scripts as well as detailed analysis of the results. Some of this functionality is currently being ported to BCL. All the related scripts can be found under fold/scripts subdirectory as well at the SVN repository. Following sections outline the steps required for preparing, running and analyzing a BCL::Fold benchmark. More detailed descriptions are provided under fold/ directory.

Determining and preparing benchmark set

First step includes determination of a benchmark set. In this study, PISCES culling server and PDB website was used for determining the set of proteins to benchmark BCL::Fold on. Once the 5 letter PDB tags are determined, download all pdb files if not already available at /blue/meilerlab/PISCES/ folder. Generate fasta, blast and secondary structure prediction files as well as BCL generated pdbs.

Setting up benchmark directory

It is advised to set up an individual folder for each large-scale benchmark. The scripts used assume a certain subdirectory structure. In order to comply with this use "make_paths.pl" script to generate sub-folders for each protein. Ideally the BCL executable to be used should be placed in the benchmark directory with name "bcl.exe" although this can be changed through flags to the scripts.

In addition to native PDB file, the list of necessary input files depends upon the Fold protocol that will be used. A stage file is commonly used to set up the different stages to be used in folding runs with corresponding energy weight sets and move probabilities if needed.

Generating PBS job submission files

An extensive script named "generate_fold_pbs.pl" was developed to ease the generation of PBS job submission scripts for BCL::Fold runs. This script allows various settings including but not limited to; benchmark directory, list of PDB tags to generate jobs for, whether predicted pools or native SSE definitions will be used in the folding runs, whether loops will be built after assembly protocol, the number of models requested, the number of CPUs requested, and specialization for Piranha or ACCRE cluster.

Analysis

Analysis of a BCL::Fold run consists of collecting scores from generated PDB files, calculating statistics, generating Pymol sessions for best models by specified measures, as well PNGs for these models again using Pymol, generation of snapshot PNGS that provide histograms, plots and other requested information. All of these tasks can be easily done by using "run_fold_analysis.pl" script. Please refer to the help of the script for more details.

BIBLIOGRAPHY

1. Karakas, M., N. Woetzel, and J. Meiler, *BCL::Contact-Low Confidence Fold Recognition Hits Boost Protein Contact Prediction and De Novo Structure Determination.* J Comput Biol, 2009.
2. Lindert, S., et al., *EM-fold: De novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps.* Structure, 2009. **17**(7): p. 990-1003.
3. Westbrook, J., et al., *The Protein Data Bank and structural genomics.* Nucleic Acids Res, 2003. **31**(1): p. 489-91.
4. Zemla, A., et al., *Processing and evaluation of predictions in CASP4.* Proteins, 2001. **Suppl 5**: p. 13-21.
5. Kryshtafovych, A., et al., *Protein structure prediction center in CASP8.* Proteins, 2009. **77 Suppl 9**: p. 5-9.
6. Carugo, O. and S. Pongor, *A normalized root-mean-square distance for comparing protein three-dimensional structures.* Protein science : a publication of the Protein Society, 2001. **10**(7): p. 1470-3.
7. Siew, N., et al., *MaxSub: an automated measure for the assessment of protein structure prediction quality.* Bioinformatics, 2000. **16**(9): p. 776-85.
8. Moult, J., et al., *Critical assessment of methods of protein structure prediction (CASP): round IV.* Proteins, 2001. **Suppl 5**: p. 2-7.
9. Cozzetto, D., et al., *Evaluation of template-based models in CASP8 with standard measures.* Proteins, 2009. **77 Suppl 9**: p. 18-28.
10. Zhang, Y., *I-TASSER: fully automated protein structure prediction in CASP8.* Proteins, 2009. **77 Suppl 9**: p. 100-13.
11. Zhou, H., S.B. Pandit, and J. Skolnick, *Performance of the Pro-sp3-TASSER server in CASP8.* Proteins, 2009. **77 Suppl 9**: p. 123-7.
12. Raman, S., et al., *Structure prediction for CASP8 with all-atom refinement using Rosetta.* Proteins, 2009. **77 Suppl 9**: p. 89-99.
13. Rost, B., *PHD: predicting one-dimensional protein structure by profile-based neural networks.* Methods Enzymol., 1996. **266**(Computer Methods for Macromolecular Sequence Analysis): p. 525-539.
14. Karplus, K., et al., *Predicting protein structure using hidden Markov models.* Proteins, 1997. **Suppl 1**: p. 134-9.
15. Meiler, J. and D. Baker, *Coupled Prediction of Protein Secondary and Tertiary Structure.* PNAS, 2003. **100**(21): p. 12105-12110.
16. Ward, J.J., et al., *Secondary structure prediction with support vector machines.* Bioinformatics, 2003. **19**(13): p. 1650-5.
17. Kuhn, M., J. Meiler, and D. Baker, *Strand-loop-strand motifs: prediction of hairpins and diverging turns in proteins.* Proteins, 2004. **54**(2): p. 282-8.
18. Jones, D.T. and J.J. Ward, *Prediction of disordered regions in proteins from position specific score matrices.* Proteins, 2003. **53 Suppl 6**: p. 573-8.

19. Linding, R., et al., *Protein Disorder Prediction: Implications for Structural Proteomics.* Structure, 2003. **11**: p. 1453-1459.
20. Grana, O., et al., *CASP6 assessment of contact prediction.* Proteins, 2005. **61 Suppl 7**: p. 214-24.
21. Liu, J. and B. Rost, *Comparing function and structure between entire proteomes.* Protein Sci, 2001. **10**(10): p. 1970-9.
22. Galzitskaya, O.V. and B.S. Melnik, *Prediction of protein domain boundaries from sequence alone.* Protein Sci, 2003. **12**(4): p. 696-701.
23. Chivian, D., et al., *Ginzu domain parsing and fold detection method* 2005.
24. Valencia, A. and F. Pazos, *Computational methods for the prediction of protein interactions.* Curr Opin Struct Biol, 2002. **12**(3): p. 368-73.
25. Ben-Hur, A. and W.S. Noble, *Kernel methods for predicting protein-protein interactions.* Bioinformatics, 2005. **21 Suppl 1**: p. i38-i46.
26. Rost, B., *Prediction in 1D: secondary structure, membrane helices, and accessibility.* Methods Biochem Anal, 2003. **44**: p. 559-87.
27. Rost, B., *Review: protein secondary structure prediction continues to rise.* J Struct Biol, 2001. **134**(2-3): p. 204-18.
28. Bradley, P., et al., *Free modeling with Rosetta in CASP6.* Proteins, 2005. **61 Suppl 7**: p. 128-34.
29. Bradley, P., et al., *Rosetta in CASP5: Progress in ab initio protein structure prediction.* Proteins: Struct., Funct., Genet., 2003. **53**(Suppl 6): p. 457-468.
30. Simons, K.T., et al., *Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions.* J. Mol. Biol., 1997. **268**: p. 209-225.
31. Bonneau, R., et al., *De Novo Prediction of Three-dimensional Structures for Major Protein Families.* J. Mol. Biol., 2002. **322**(1): p. 65-78.
32. Fleishman, S.J., et al., *Quasi-symmetry in the cryo-EM structure of EmrE provides the key to modeling its transmembrane domain.* Journal of molecular biology, 2006. **364**(1): p. 54-67.
33. Lindert, S., et al., *Cryo-electron microscopy structure of an adenovirus-integrin complex indicates conformational changes in both penton base and integrin.* Journal of virology, 2009. **83**(22): p. 11491-501.
34. Van Horn, W.D., et al., *Solution nuclear magnetic resonance structure of membrane-integral diacylglycerol kinase.* Science, 2009. **324**(5935): p. 1726-9.
35. Qian, B., et al., *High-resolution structure prediction and the crystallographic phase problem.* Nature, 2007. **450**(7167): p. 259-64.
36. Meiler, J. and D. Baker, *Rapid Protein Structure Elucidation Utilizing Unassigned NMR Data.* PNAS, 2003. **100**(26): p. 15404-15409.
37. Raman, S., et al., *NMR structure determination for larger proteins using backbone-only data.* Science, 2010. **327**(5968): p. 1014-8.
38. Alexander, N., et al., *De novo high-resolution protein structure determination from sparse spin-labeling EPR data.* Structure, 2008. **16**(2): p. 181-95.
39. Kuhlman, B. and D. Baker, *Native protein sequences are close to optimal for their structures.* Proc. Natl. Acad. Sci. U. S. A., 2000. **97**(19): p. 10383-10388.
40. Aszodi, A., M.J. Gradwell, and W.R. Taylor, *Global fold determination from a small number of distance restraints.* J Mol Biol, 1995. **251**(2): p. 308-26.

41. Huang, E.S., R. Samudrala, and J.W. Ponder, *Ab initio fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions.* J Mol Biol, 1999. **290**(1): p. 267-81.
42. Bonneau, R., et al., *Contact order and ab initio protein structure prediction.* Protein Sci, 2002. **11**: p. 1937-1944.
43. Baker, D., *A surprising simplicity to protein folding.* Nature, 2000. **405**(6782): p. 39-42.
44. Izarzugaza, J.M., et al., *Assessment of intramolecular contact predictions for CASP7.* Proteins, 2007. **69 Suppl 8**: p. 152-8.
45. Plaxco, K.W., et al., *Evolutionary conservation in protein folding kinetics.* J Mol Biol, 2000. **298**(2): p. 303-12.
46. Punta, M. and B. Rost, *Protein folding rates estimated from contact predictions.* J Mol Biol, 2005. **348**(3): p. 507-12.
47. Olmea, O., B. Rost, and A. Valencia, *Effective use of sequence correlation and conservation in fold recognition.* J Mol Biol, 1999. **293**(5): p. 1221-39.
48. Cheng, J. and P. Baldi, *A machine learning information retrieval approach to protein fold recognition.* Bioinformatics, 2006. **22**(12): p. 1456-63.
49. Gobel, U., et al., *Correlated mutations and residue contacts in proteins.* Proteins, 1994. **18**(4): p. 309-17.
50. Olmea, O. and A. Valencia, *Improving contact predictions by the combination of correlated mutations and other sources of sequence information.* Fold Des, 1997. **2**(3): p. S25-32.
51. Shindyalov, I.N., N.A. Kolchanov, and C. Sander, *Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations?* Protein Eng, 1994. **7**(3): p. 349-58.
52. Hamilton, N., et al., *Protein contact prediction using patterns of correlation.* Proteins, 2004. **56**(4): p. 679-84.
53. Halperin, I., H. Wolfson, and R. Nussinov, *Correlated mutations: advances and limitations. A study on fusion proteins and on the Cohesin-Dockerin families.* Proteins, 2006. **63**(4): p. 832-45.
54. Kundrotas, P.J. and E.G. Alexov, *Predicting residue contacts using pragmatic correlated mutations method: reducing the false positives.* BMC Bioinformatics, 2006. **7**: p. 503.
55. Fariselli, P., et al., *Prediction of contact maps with neural networks and correlated mutations.* Protein Eng, 2001. **14**(11): p. 835-43.
56. Lund, O., et al., *Protein distance constraints predicted by neural networks and probability density functions.* Protein Eng, 1997. **10**(11): p. 1241-8.
57. Fariselli, P. and R. Casadio, *A neural network based predictor of residue contacts in proteins.* Protein Eng, 1999. **12**(1): p. 15-21.
58. Fariselli, P., et al., *Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations.* Proteins, 2001. **Suppl 5**: p. 157-62.
59. Pollastri, G. and P. Baldi, *Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners.* Bioinformatics, 2002. **18 Suppl 1**: p. S62-70.

60. Pollastri, G., et al., *Improved prediction of the number of residue contacts in proteins by recurrent neural networks.* Bioinformatics, 2001. **17 Suppl 1**: p. S234-42.
61. Punta, M. and B. Rost, *PROFcon: novel prediction of long-range contacts.* Bioinformatics, 2005. **21**(13): p. 2960-8.
62. Cheng, J. and P. Baldi, *Improved residue contact prediction using support vector machines and a large feature set.* BMC Bioinformatics, 2007. **8**: p. 113.
63. Zhang, Y. and J. Skolnick, *Automated structure prediction of weakly homologous proteins on a genomic scale.* Proc Natl Acad Sci U S A, 2004. **101**(20): p. 7594-9.
64. Wu, S. and Y. Zhang, *A comprehensive assessment of sequence-based and template-based methods for protein contact prediction.* Bioinformatics, 2008. **24**(7): p. 924-31.
65. Shao, Y. and C. Bystroff, *Predicting interresidue contacts using templates and pathways.* Proteins, 2003. **53 Suppl 6**: p. 497-502.
66. Sali, A. and T.L. Blundell, *Comparitive Protein Modelling by Satisfaction of Spatial Restraints.* J. Mol. Biol., 1993. **234**: p. 779-815.
67. Chivian, D., et al., *Prediction of CASP-6 structures using automated Robetta protocols.* Proteins, 2005.
68. Shackelford, G. and K. Karplus, *Contact prediction using mutual information and neural nets.* Proteins, 2007. **69 Suppl 8**: p. 159-64.
69. Wu, S. and Y. Zhang, *LOMETS: a local meta-threading-server for protein structure prediction.* Nucleic Acids Res, 2007. **35**(10): p. 3375-82.
70. Lee, S.Y. and J. Skolnick, *Benchmarking of TASSER_2.0: an improved protein structure prediction algorithm with more accurate predicted contact restraints.* Biophys J, 2008. **95**(4): p. 1956-64.
71. Skolnick, J., D. Kihara, and Y. Zhang, *Development and large scale benchmark testing of the PROSPECTOR_3 threading algorithm.* Proteins, 2004. **56**(3): p. 502-18.
72. Li, W., Y. Zhang, and J. Skolnick, *Application of sparse NMR restraints to large-scale protein structure prediction.* Biophys J, 2004. **87**(2): p. 1241-8.
73. Pollock, D.D., W.R. Taylor, and N. Goldman, *Coevolving protein residues: maximum likelihood identification and relationship to structure.* J Mol Biol, 1999. **287**(1): p. 187-98.
74. Altschuh, D., et al., *Coordinated amino acid changes in homologous protein families.* Protein Eng, 1988. **2**(3): p. 193-9.
75. Wang, G. and R.L. Dunbrack, Jr., *PISCES: a protein sequence culling server.* Bioinformatics, 2003. **19**(12): p. 1589-91.
76. Rychlewski, L. and D. Fischer, *LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction.* Protein Sci, 2005. **14**(1): p. 240-5.
77. Meiler, J., et al., *Generation and Evaluation of Dimension Reduced Amino Acid Parameter Representations by Artificial Neural Networks.* J. Mol. Model., 2001. **7**(9): p. 360-369.
78. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res., 1997. **25**: p. 3389-3402.

79. Ginalski, K., et al., *ORFeus: Detection of distant homology using sequence profiles and predicted secondary structure.* Nucleic Acids Res, 2003. **31**(13): p. 3804-7.
80. McGuffin, L.J. and D.T. Jones, *Improvement of the GenTHREADER method for genomic fold recognition.* Bioinformatics, 2003. **19**(7): p. 874-81.
81. Bujnicki, J.M., et al., *LiveBench-1: continuous benchmarking of protein structure prediction servers.* Protein Sci, 2001. **10**(2): p. 352-61.
82. Jones, D.T., *GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences.* J Mol Biol, 1999. **287**(4): p. 797-815.
83. Russell, R.B., et al., *Recognition of analogous and homologous protein folds--assessment of prediction success and associated alignment accuracy using empirical substitution matrices.* Protein Eng, 1998. **11**(1): p. 1-9.
84. Lundstroem, J., et al., *Pcons: A neural-network –based consensus predictor that improves fold recognition.* Protein Sci., 2001. **10**: p. 2354-2362.
85. Fischer, D., *3DS3 and 3DS5 3D-SHOTGUN meta-predictors in CAFASP3.* Proteins, 2003. **53 Suppl 6**: p. 517-23.
86. Ginalski, K., et al., *3D-Jury: a simple approach to improve protein structure predictions.* Bioinformatics, 2003. **19**(8): p. 1015-8.
87. Ginalski, K. and L. Rychlewski, *Detection of reliable and unexpected protein fold predictions using 3D-Jury.* Nucleic Acids Res, 2003. **31**(13): p. 3291-2.
88. Jaroszewski, L., et al., *FFAS03: a server for profile--profile sequence alignments.* Nucleic Acids Res, 2005. **33**(Web Server issue): p. W284-8.
89. Tomii, K., T. Hirokawa, and C. Motono, *Protein structure prediction using a variety of profile libraries and 3D verification.* Proteins, 2005. **61 Suppl 7**: p. 114-21.
90. Shi, J., T.L. Blundell, and K. Mizuguchi, *FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties.* J Mol Biol, 2001. **310**(1): p. 243-57.
91. Skolnick, J. and D. Kihara, *Defrosting the frozen approximation: PROSPECTOR--a new approach to threading.* Proteins, 2001. **42**(3): p. 319-31.
92. Chivian, D. and D. Baker, *Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection.* Nucleic Acids Res, 2006.
93. Karplus, K., et al., *SAM-T04: what is new in protein-structure prediction for CASP6.* Proteins, 2005. **61 Suppl 7**: p. 135-42.
94. Karplus, K. and B. Hu, *Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set.* Bioinformatics, 2001. **17**(8): p. 713-20.
95. Debe, D.A., et al., *STRUCTFAST: protein sequence remote homology detection and alignment using novel dynamic programming and profile-profile scoring.* Proteins, 2006. **64**(4): p. 960-7.
96. Fischer, D., et al., *CAFASP3: the third critical assessment of fully automated structure prediction methods.* Proteins, 2003. **53**(Suppl 6): p. 503-16.
97. Zhang, W., S. Liu, and Y. Zhou, *SP5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model.* PLoS ONE, 2008. **3**(6): p. e2325.

98. Torda, A.E., J.B. Procter, and T. Huber, *Wurst: a protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices.* Nucleic Acids Res, 2004. **32**(Web Server issue): p. W532-5.
99. Ortiz, A.R., C.E.M. Strauss, and O. Olmea, *MAMMOTH (Matching molecular models obtained from theory): An automated method for model comparison.* Protein Sci., 2002. **11**: p. 2606-2611.
100. Kendrew, J.C., et al., *A three-dimensional model of the myoglobin molecule obtained by x-ray analysis.* Nature, 1958. **181**(4610): p. 662-6.
101. Wüthrich, K., *NMR of Proteins and Nucleic Acids (1H-NMR shifts of amino acids)*. Vol. ISBN 0-471-82893-9. 1986, New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons. 17.
102. Berman, H.M., *The Protein Data Bank: a historical perspective.* Acta crystallographica. Section A, Foundations of crystallography, 2008. **64**(Pt 1): p. 88-95.
103. Loll, P.J., *Membrane protein structural biology: the high throughput challenge.* J Struct Biol, 2003. **142**(1): p. 144-53.
104. Lepault, J., F.P. Booy, and J. Dubochet, *Electron microscopy of frozen biological suspensions.* Journal of microscopy, 1983. **129**(Pt 1): p. 89-102.
105. Saban, S.D., et al., *Visualization of alpha-helices in a 6-angstrom resolution cryoelectron microscopy structure of adenovirus allows refinement of capsid protein assignments.* Journal of virology, 2006. **80**(24): p. 12049-59.
106. Read, R.J. and G. Chavali, *Assessment of CASP7 predictions in the high accuracy template-based modeling category.* Proteins, 2007. **69 Suppl 8**: p. 27-37.
107. Simons, K.T., et al., *Improved Recognition of Native-Like Protein Structures Using a Combination of Sequence-Dependent and Sequence-Independent Features of Proteins.* Proteins: Structure, Function, and Genetics, 1999. **34**: p. 82-95.
108. Bradley, P., K.M. Misura, and D. Baker, *Toward high-resolution de novo structure prediction for small proteins.* Science, 2005. **309**(5742): p. 1868-71.
109. Das, R., et al., *Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home.* Proteins, 2007. **69 Suppl 8**: p. 118-28.
110. Rohl, C.A., *Protein structure estimation from minimal restraints using Rosetta.* Methods in enzymology, 2005. **394**: p. 244-60.
111. Cavalli, A., P. Ferrara, and A. Caflisch, *Weak temperature dependence of the free energy surface and folding pathways of structured peptides.* Proteins, 2002. **47**(3): p. 305-14.
112. Lee, M.R., et al., *Molecular dynamics in the endgame of protein structure prediction.* Journal of molecular biology, 2001. **313**(2): p. 417-30.
113. Brooks, B.R., et al., *CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations.* J. Comp. Chem., 1983. **4**: p. 187-217.
114. Ponder, J.W. and D.A. Case, *Force fields for protein simulations.* Advances in protein chemistry, 2003. **66**: p. 27-85.
115. Wang, G. and R.L. Dunbrack, Jr., *PISCES: recent improvements to a PDB sequence culling server.* Nucleic Acids Res, 2005. **33**(Web Server issue): p. W94-8.
116. De Boor, C., *A practical guide to splines*2001, New York: Springer

346.
117. Hsin, J., et al., *Using VMD: an introductory tutorial.* Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.], 2008. **Chapter 5**: p. Unit 5 7.
118. Durham, E., et al., *Solvent accessible surface area approximations for rapid and accurate protein structure prediction.* Journal of molecular modeling, 2009. **15**(9): p. 1093-108.
119. Rao, S.T. and M.G. Rossmann, *Comparison of super-secondary structures in proteins.* Journal of molecular biology, 1973. **76**(2): p. 241-56.
120. Chothia, C., *Principles that determine the structure of proteins.* Annual review of biochemistry, 1984. **53**: p. 537-72.
121. Kabsch, W. and C. Sander, *Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features.* Biopolymers, 1983. **22**: p. 2577-2637.
122. Berman, H.M., et al., *The Protein Data Bank.* Acta Crystallogr D Biol Crystallogr, 2002. **58**(Pt 6 No 1): p. 899-907.
123. Daga, P.R., R.Y. Patel, and R.J. Doerksen, *Template-based protein modeling: recent methodological advances.* Current topics in medicinal chemistry, 2010. **10**(1): p. 84-94.
124. Stevens, R.C., S. Yokoyama, and I.A. Wilson, *Global efforts in structural genomics.* Science, 2001. **294**(5540): p. 89-92.
125. Lesley, S.A., et al., *Structural genomics of the Thermotoga maritima proteome implemented in a high-throughput structure determination pipeline.* Proc Natl Acad Sci U S A, 2002. **99**(18): p. 11664-9.
126. DiMaio, F., et al., *Improved molecular replacement by density- and energy-guided protein structure optimization.* Nature, 2011. **473**(7348): p. 540-3.
127. Bill, R.M., et al., *Overcoming barriers to membrane protein structure determination.* Nature biotechnology, 2011. **29**(4): p. 335-40.
128. Oberai, A., et al., *A limited universe of membrane protein families and folds.* Protein Sci, 2006. **15**(7): p. 1723-34.
129. Alber, F., et al., *Determining the architectures of macromolecular assemblies.* Nature, 2007. **450**(7170): p. 683-94.
130. Yooseph, S., et al., *The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families.* PLoS biology, 2007. **5**(3): p. e16.
131. Zhou, H. and J. Skolnick, *Ab initio protein structure prediction using chunk-TASSER.* Biophys J, 2007. **93**(5): p. 1510-8.
132. Dahiyat, B.I. and S.L. Mayo, *De novo protein design: fully automated sequence selection.* Science, 1997. **278**(5335): p. 82-7.
133. Dunbrack, R.L., Jr., *Rotamer libraries in the 21st century.* Curr Opin Struct Biol, 2002. **12**(4): p. 431-40.
134. Smith, J.A., et al., *Structural models for the KCNQ1 voltage-gated potassium channel.* Biochemistry, 2007. **46**(49): p. 14141-52.
135. Eswar, N., et al., *Comparative protein structure modeling using Modeller.* Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.], 2006. **Chapter 5**: p. Unit 5 6.

136. Canutescu, A.A. and R.L. Dunbrack, *Cyclic coordinate descent: A robotics algorithm for protein loop closure.* Protein Sci, 2003. **12**: p. 963-972.
137. Rohl, C.A., et al., *Modeling structurally variable regions in homologous proteins with rosetta.* Proteins, 2004. **55**(3): p. 656-77.
138. Grantcharova, V., et al., *Mechanisms of protein folding.* Curr Opin Struct Biol, 2001. **11**(1): p. 70-82.
139. Plaxco, K.W., et al., *Topology, stability, sequence, and length: defining the determinants of two-state protein folding kinetics.* Biochemistry, 2000. **39**(37): p. 11177-83.
140. Zimmer, J., Y. Nam, and T.A. Rapoport, *Structure of a complex of the ATPase SecA and the protein-translocation channel.* Nature, 2008. **455**(7215): p. 936-43.
141. Sibanda, B.L., D.Y. Chirgadze, and T.L. Blundell, *Crystal structure of DNA-PKcs reveals a large open-ring cradle comprised of HEAT repeats.* Nature, 2010. **463**(7277): p. 118-21.
142. Skrisovska, L., M. Schubert, and F.H. Allain, *Recent advances in segmental isotope labeling of proteins: NMR applications to large proteins and glycoproteins.* Journal of biomolecular NMR, 2010. **46**(1): p. 51-65.
143. Singh, P., A. Panchaud, and D.R. Goodlett, *Chemical cross-linking and mass spectrometry as a low-resolution protein structure determination technique.* Analytical chemistry, 2010. **82**(7): p. 2636-42.
144. Kazmier, K., et al., *Algorithm for selection of optimized EPR distance restraints for de novo protein structure determination.* Journal of structural biology, 2011. **173**(3): p. 549-57.
145. Hirst, S.J., et al., *RosettaEPR: an integrated tool for protein structure determination from sparse EPR data.* Journal of structural biology, 2011. **173**(3): p. 506-14.
146. Kalkhof, S., et al., *Computational modeling of laminin N-terminal domains using sparse distance constraints from disulfide bonds and chemical cross-linking.* Proteins, 2010. **78**(16): p. 3409-27.
147. Hussain, S.A., F. Carafoli, and E. Hohenester, *Determinants of laminin polymerization revealed by the structure of the alpha5 chain amino-terminal region.* EMBO reports, 2011. **12**(3): p. 276-82.
148. Kolinski, A. and J. Skolnick, *Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model.* Proteins, 1998. **32**(4): p. 475-94.
149. Latek, D. and A. Kolinski, *CABS-NMR--De novo tool for rapid global fold determination from chemical shifts, residual dipolar couplings and sparse methyl-methyl NOEs.* Journal of computational chemistry, 2011. **32**(3): p. 536-44.
150. Barth, P., B. Wallner, and D. Baker, *Prediction of membrane protein structures with complex topologies using limited constraints.* Proceedings of the National Academy of Sciences of the United States of America, 2009. **106**(5): p. 1409-14.
151. Meiler, J. and D. Baker, *Coupled prediction of protein secondary and tertiary structure.* Proceedings of the National Academy of Sciences of the United States of America, 2003. **100**(21): p. 12105-12110.

152. Meiler, J., et al., *Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks.* Journal of Molecular Modeling, 2001. **7**(9): p. 360-369.
153. Jones, D.T., *Protein secondary structure prediction based on position-specific scoring matrices.* Journal of Molecular Biology, 1999. **292**(2): p. 195-202.
154. Rost, B., C. Sander, and R. Schneider, *Redefining the Goals of Protein Secondary Structure Prediction.* J. Mol. Biol., 1994. **235**: p. 13-26.
155. Moult, J., *A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction.* Curr Opin Struct Biol, 2005.
156. Moult, J., et al., *Critical assessment of methods of protein structure prediction (CASP): round II.* Proteins, 1997. **Suppl 1**: p. 2-6.
157. Fiser, A. and A. Sali, *Modeller: generation and refinement of homology-based protein structure models.* Methods in enzymology, 2003. **374**: p. 461-91.
158. Bernstein, F.C., et al., *The Protein Data Bank: a computer-based archival file for macromolecular structures.* J Mol Biol, 1977. **112**(3): p. 535-42.
159. Altschul, S.F., et al., *Basic local alignment search tool.* J. Mol. Biol., 1990. **215**(3): p. 403-410.
160. Metropolis, N.a.U., S., *The Monte Carlo Method.* J. Amer. Stat. Assoc, 1949. **44**: p. 335-341.
161. Wang, G.L. and R.L. Dunbrack, *PISCES: recent improvements to a PDB sequence culling server.* Nucleic Acids Research, 2005. **33**: p. W94-W98.