A NOVEL APPROACH TO DE NOVO PROTEIN STRUCTURE PREDICTION USING KNOWLEDGE BASED ENERGY FUNCTIONS AND EXPERIMENTAL RESTRAINTS

By

Nils Wötzel

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

Chemistry

December, 2011

Nashville, Tennessee

Approved:

Professor Jens Meiler

Professor B. Andes Hess Jr.

Professor Clare M. McCabe

Professor Phoebe L. Stewart

To Juliane, my parents and my sister

ACKNOWLEDGEMENTS

I thank my dissertation advisor for his scientific and personal support throughout my high school and graduate career. I see him not only as my scientific mentor, but also as a great friend who always had a word of advice. Although I have a great friendship with him, his wife Claudia and his son Jonas, we always worked productively in a professional student-mentor relationship. He influenced my decision to study chemistry, encouraged me to visit the USA for an internship and finally to join his lab for PhD career. He prepared me to take the next step in my life and become an independent scholar.

The six years in graduate school taught me many things. Besides the scientific knowledge and the skills that were required to accomplish the tasks that were set, I could acquire great experiences from the work with colleagues. Rene Staritzbichler started the project with us and went through many iterations of improving aspects of the work. Mert Karakaş has been a valuable peer who took on the challange with me to develop the BCL::Fold method, going through many ups and downs. Nathan Alexander, Ralf Mueller, Brian Weiner and Edward Lowe have been involved in many scientific discussions that increased the quality of the work where we can publish results and can be proud of the tools that can and are used by the scientific community.

My thesis committe was of great help and engaged in constructive scientific discussion. Dr. Clare McCabe and Dr. Andes Hess Jr. helped me see my work from a differnt perspective. Dr. Pheobe Stewart was not only a helpful advisor but also an exceptional collaborator contributing to the efforts to bring many projects to a success.

Besides many funding form Vanderbilt, National Institue of Healt and the National Science foundation for the projects that I worked on, I would like to thank the Department of Chemistry for their support through the Warren research fellowship in the year 2010.

Since I came to Nashville, I was fortunate to have had roomates that accepted my flaws and were always there when I needed them. Ralf Mueller, Andrew Morin and Nathan Alexander were not only roomates, but we became friends beyond peership.

I also want to pay my gratitude to my parents, that supported me in my decision to leave Germany to take a new step in my career in a different country. It is not easy to let the last child leave home. We had to sacrifice many holidays and occasions that we used to celebrate together. My sister always supported me, and besides all the problems there are with older sisters, I love her and look forward to pentding more time with her again.

Lastly, I want to thank my better half, the most valuable person in my life. We met just 2.5 years before I left Germany and ever since we were living apart. Juliane, my fiance, sacrificed 6 years waiting for me, and I hope we can spent many more days together than we were separated. There is no way to ever compensate for that time, but I will try to be the person, that she was waiting for all this time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artifical neural network
BCL	BioChemistry library
CASP	Critical assessment of protein structure prediction
CCC	Cross correlation coefficient
CO	Contact order
CRYOEM	Cryo-electron microscopy
DNA	Deoxyribonucleic acid
EPR	Electron paramagnetic resonance
FN	False negative
FP	False positive
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
RNA	Ribonucleic acid
SSE	Secondary structure element
SVM	Support vector machine
RCO	Relative contact order
RMSD	Root mean square deviation
TN	True negative
TP	True positive
VDW	Van der Waals

SUMMARY

The focus of this work was to develop a method for rapid fitting of atomic resolution structural models into medium resolution electron density maps and Bayesian energy potentials for a *de novo* protein structure prediction method. The developed methods, BCL::EM-Fit, BCL::ScoreProtein and BCL::Fold, were benchmarked on large sets of proteins. All described work is implemented in the object oriented C++ software library termed “BioChemistry Library” (BCL), developed in Meiler Lab.

Chapter I provides an introduction with a limited overview of protein structure and experimental methods for protein structure elucidation. Additionally, computational protein energy evaluation through knowledge and physics based energy functions is introduced. Lastly, methods for protein-protein structure comparison are discussed. Chapter II describes BCL::EM-Fit, the algorithm for rapid fitting of atomic structures into electron density maps based on the image recognition algorithm known as “geometric hashing” employed for three dimensional problems. The chapter discusses how it improves time, accuracy and completeness compared to other methods. Chapter III concentrates on Bayesian energy potentials which are derived to evaluate protein structures focusing on the protein topology represented by the geometrical arrangement of secondary structure elements. This potential is used within 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 excerpt of a benchmark of the method.

Chapter II is a reproduction of the first author paper “BCL::EM-Fit: Rapid fitting of atomic structures into density maps using geometric hashing and real space refinement”

published in 2011 in Journal of Structural Biology [1]. Chapter III and Chapter IV are reproductions of co-first authored manuscripts titled "Knowledge based energy potentials for ranking protein models represented by idealized secondary structure elements" and "*De novo* prediction of complex and large protein topologies by assembly of secondary structure elements" respectively. Both of these manuscripts are currently in the process of being submitted to "PLoS Computational Biology" and are result of collaborative work with Mert Karakaş, another graduate student in the Meiler Lab.

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

CHAPTER I

INTRODUCTION

Central Dogma of Molecular Biology

The central dogma of molecular biology defines that the DNA is transcribed into RNA. RNA is translated into a primary amino acid sequence. It was first formulated by Crick [3]. One can expand this dogma to include that the amino acid sequence defines the secondary and tertiary structure of proteins. Those proteins can interact and form quaternary structures. Transcription can be a bidirectional process, while translation is believed to be only one directional. This dogma defines the working hypothesis for the field of structural biology.

Protein Structure and Function

While the common teaching in school is, that proteins are enzymes, catalyzing reaction by lowering the activation barrier or stabilizing the reaction's transition state, proteins serve many more purposes, from transport over signaling to being structural components in biological systems. A more general definition could be that they are poly peptides with a biological function. The biological function is encoded in the quaternary, tertiary, secondary and primary structure. In reference to the central dogma of molecular biology: the function of a protein is encoded in the DNA.

If one wants to understand the function of a protein and how it serves a purpose, it is unavoidable to elucidate its three dimensional structure. Proteins can consist of multiple amino acids sequences (or chains) that make up the quaternary structure. Each chain is a polymer of peptide-bond linked amino acids. An amino acid is a small molecule and consists of three parts: an amine and a carboxyl group that condenses, building the backbone of the protein; the side chain distinguishes 20 natural occurring amino acids. The length of the primary sequence for a protein can range from a few to over a thousand amino acids.

Secondary structure is defined by the hydrogen bonding interactions between the backbone carboxyl oxygen and the amide hydrogen forming α-helices and β-strands. Tertiary structure is the three dimensional topology of the arrangement of secondary structure elements formed by the interactions of the amino acid side chains.

Protein Structure Elucidation

Proteins can crystallize into regular crystal lattices. This arrangement results in identical distances between the same atoms in two different protein structures. Due to many proteins in the lattice, many identical distances can be found. This phenomenon can be used in x-ray diffraction [4], where these distances can be observed as inverse distances in a x-ray diffraction pattern. For the resulting diffraction pattern the phases are missing, but with proper techniques it is possible to determine the phases and a three dimensional structure of the crystallized protein can be elucidated. This is the most common technique to determine the structure of proteins and accounts for almost 90% of the proteins in the Protein Data Bank [5].

Nuclear magnetic resonance (NMR) [6] can be utilized to determine the atomic distances and angular constraints of proteins in solution. These constraints have to be used to generate possible structures that fulfill as many of the constraints as possible.

As of September 2011 the PDB contains more than 70k protein structures. Although, many of the deposited proteins are similar in sequence, over a quarter of the structures are different in sequence by at least 70%. Other methods besides x-ray crystallography and NMR are used to elucidate structures like cryo electron microscopy, electron paramagnetic resonance of site-directed spin labeled proteins and hybrid methods.

In Silico Protein Structure Prediction

The extended central dogma of molecular biology (RNA => DNA => amino acid sequence => tertiary protein structure) provides a working hypothesis for computational protein structure prediction. Methodologies that can make computational predictions for a protein's tertiary or even quaternary structure from the primary amino acid sequence can be developed. With significant advances in the availability and performance of computational resources, algorithms became applicable to that problem in the recent decade. One the one hand, sampling methods that generate many different protein structures can be extensive and highly detailed, e.g. molecular dynamics can work with full atom representations of the protein structure. On the other hand, energy evaluations of the generated models can be done in a timely manner using coarse or even fine grained energy functions. Some of the more robust computational methods pose alternatives to bench experiments to generate hypotheses about the protein's structure and function. Despite the successes of these algorithms, many predictions need experimental validation and should be understood as an assisting tool in structure elucidation.

The field of protein structure prediction is divided into two classes. If a tertiary structure with high sequence similarity to the protein of interest is known, template-based modeling can be applied [7]. If this template is absent, *de novo* methods are applied [8].

Both classes require the primary sequence of the protein of interest. The goal is an atomic detail tertiary structure. Depending on the class of the problem and the method, one model can be built in a few minutes or many models using many computers are generated. A small portion of the models, that cluster close together or fulfill additional experimental restraints represent the space of possible structures and can be used to test new hypotheses experimentally.

The Critical Assessment for Techniques for Protein Structure Prediction (CASP) experiment provides a platform to test computational methods [9]. The CASP organizers acquire primary sequences from experimental groups and the structural genomics project for proteins that are to be elucidated. Biannually, during a three month summer prediction season, the target sequences are released to participating groups, who apply their method and submit structural models for those proteins. Target proteins for both classes are relayed: template-based modeling targets (TBM) and free modeling (*de novo*) targets (FM), based on the availability and the sequence similarity of template proteins for the given target. Fully automated methods work as server predictors, manual methods that usually employ personal scientific expertise in modeling and model selection are categorized as human predictors. In the 9th round of 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 are released.

The history of computational structure prediction and the current efforts in the field define a frame in which one can develop a competitive computation structure prediction algorithm. Some of the principles to develop knowledge based energy potentials can be used together with new ideas that extend beyond those that have been employed so far. Rapid but sufficiently accurate evaluation of structural models together with a structural sampling algorithm, a methodology can be defined that overcomes current size limitation in *in silico* structure prediction. With the established CASP blind experiment, the method can be tested against other algorithms.

Protein Structure Comparison Methods

To define the difference between two structural models of a protein, different measures have been introduced. They can be used to assess the quality of a structural model against the native protein structure elucidate by an experimental technique. When multiple structural models are built, and it is not feasible to consider all of them, they can be used in clustering, where only centers of clusters with a maximal (or minimal) girth or cluster member differences in the quality measure is allowed [10].

The root mean square deviation (RMSD) of the coordinates of a subset of atoms in the structural models is a widely used measure. It is calculated after optimally superimposing the two structures in question. Commonly, C_α-backbone atoms are used since they define the conformation of the backbone and hence, the topology sufficiently. The RMSD is also used in small molecule structure or position comparison. All backbone atoms can be used for high accuracy evaluation of homology/template-based modeling. An 8Å RMSD cutoff is defined to be a native-like topology in this work, an RMSD observed, when the difference between the native topology to the protein model is a single misplaced or flipped SSE. When overlaying two structural models, one could identify a different in at least one SSE.

The amino acid primary sequence length normalized RMSD: RMSD100 [6] is used to compare a method's performance on multiple protein targets of varying sequence lengths.

RMSD measures only the best global superimposition. Sometimes, optimal local superimposition is desired, if one is interested if a domain of a protein is resembled in a model. Comparison methods have been developed to address this question: MaxSub [11] and Global Distance Test (GDT) [12] 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.

Template Based Protein Structure Prediction

Since the tertiary structure of proteins is a result of the primary sequence, it can be assumed, and has been observed, that similar sequences will adopt the same tertiary structure. In order to identify such a structural model, the sequence in question is aligned against a databank of sequences of proteins with known atomic structure. Sequence with 30% or higher sequence similarity have a high probability to adopt the same tertiary structure, in rare cases templates of sequence similarity as low as 10% can be used to build a structural model. Methods can also use templates for different parts of the sequence to build models of higher quality. The query sequence is then associated with the coordinates of the template according to the optimal alignment of the sequences. Many template based methods are available and did participate in the CASP 9 experiment [13].

De novo Protein Structure Prediction

If a template structure for a protein of known amino acid sequence cannot be identified, *de novo* methods can be used to generate structural models. The structural model has to be assembled from the primary sequence, usually by starting from an extended structure of the chain. The keys to a successful method are the choice of complementing structural sampling and structure evaluation. Many models have to be generated, filtered and clustered to come up with candidate models.

The expected accuracies are lower for *de novo* methods than for template based modeling, due to the reduced representation of the amino acid chain to simplify sampling and evaluation of the structural models. This simplification often omits side chains by replacing them with "centroid" atoms or the first amino acid's side chain atom C_β only. Although this enables faster evaluation of the scoring function for structural models, the energy landscape contains fewer features. In consequence, the global minimum is not significantly differentiated from other native-like topologies.

De novo protein structure prediction typically starts with predicting secondary structure [14-16] and non-local contacts [17]. This is done based only on the primary amino acid sequence. The sequence itself contains sufficient information, so that system-learning approaches can be used. The most commonly, artificial neural networks (ANN), hidden Markov models (HMM), and support vector machines (SVM) [18], [19] are used.

The predictions for features from the primary sequence only can now be used in the following step. For the primary sequence through a sampling algorithm, three dimensional models for the primary amino acid sequence are generated in a sampling trajectory. For each of the structural models during the sampling, the energy is evaluated. Based on that energy, the structural model is accepted, and is subject to the next sampling step, or a previous structure with a more favorable energy is used for further structural sampling.

Assembling fragments of 3 and 9 residues that are homologous to the query sequence, Rosetta literally folds the extended chain with likely phi-psi angles according to the fragments [20-23]. Rosetta is capable of correctly folding about 50% of all sequences with less than 150 amino acids [24].

Protein Structure Prediction using Low Resolution/Sparse Experimental Restraints

Besides x-ray crystallography and NMR experiments, experimental techniques have been established that can derive experimental restraints significantly constraining the *de novo* structure prediction problem. These constraints or restraints limit the conformational space for the protein models that needs to be sampled. Additionally, energy terms can be introduced that lower the energy for models close to the native structure.

Cryo-electron microscopy (cryoEM) yields electron density maps that show an envelope or even the topology of proteins. Viruses and other large macromolecular assemblies can be imaged. At resolutions below 9Å, α-helices can be traced. Above this resolution individual domains or proteins can be depicted.

Site directed spin labeling can be used to derive amino acid solvent exposure or distance restraints in Eelectron paramagnetic resonance (EPR) experiments [25]. Mass spectrometry can map di-sulfide bonds or using chemical linkers, it can also define inter- and intra-molecule distance restrains [26]. If proteins are challenging, NMR sometimes only provides a few and even unassigned distance and residual dipolar coupling restraints that can be used as orientation restraints [27]. These restraints might not be enough for classical structure elucidation, but can complement computation *de novo* methods.

The given restraints, combined from different methods, decrease the sampling space significantly and introduce new features into the energy function that is used to evaluate structural models [27], [28]. The restraint decrease the native's energy minima relative to all other native like minima. This enables faster model generation and increases the accuracy of and confidence in the final models [2].

BCL::Score

An integral part of *de novo* protein structure elucidation is the evaluation of the generated models. The objective is to quantify the likelihood that a given model is native-like. A protein structure is considered native-like if it could be observed in an experiment. This native-likeliness is classically defined by the energy that comes from the interactions of atoms with each other. Classical potentials are derived from first physical principles. Some of them are derived directly from quantum mechanical calculations. Others are derived from experimental atom distance and bond angles and dihedrals which are centers of harmonic potentials [29]. The disadvantage of this approach is, that the energy evaluation of any given protein model is time consuming and it relies on a full atomic representation of the structural model.

Using the BOLTZMANN relationship, one can derive an all atom statistical potential. Assuming that a non-redundant set of experimental atomic protein structures represents features that follow a BOLTZMANN-like distribution, these potentials are close to the physical truth and energies can be derived, that are close to reality. The relationship itself requires a correct reference state and the databank used is required to adhere to the assumption of a BOLTZMANN distribution [30].

In recent years, Bayesian derived potentials grew in importance. They do not rely on atoms to be evaluated and an absolute reference state is also not required. The potentials are a quasi-BOLTZMANN energy expressing the likelihood of observing a structural feature in a given model, compare to the observed likelihood in a databank of structures. It is corrected by the random chance to observe that feature. They have proven robust and in favorable cases, the energy can be correlated with experimental measured energies.

BCL::Score introduces a Bayesian scoring potential that focuses on evaluating a protein's topology as it is defined by the assembly of secondary structure elements. It works of the hypothesis, that the stability of the protein's fold is defined by the interaction of the core residues of the protein, which pack most densely at interfaces of interacting secondary structure elements. Besides terms that are used in other modeling programs, like an amino acid pair distance potential and an amino acid solvation potential, it introduces terms that are focused on the topology of the model. Secondary structure element (SSE) packing, a loop length and a contact order potential are defined in that respect.

Computational evaluation of the energy terms are quick due to the reduced representation of the models as defined by the assembly of helices and strands with only one side chain atom to represent amino acids. This makes the potential suitable to explore a large conformational space in a small amount of time.

BCL::Fold

The BCL::Fold protein structure prediction method is developed to address the current limitations of *de novo* protein structure methods. Many methods are not applicable 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 abundance of non-local amino acid contacts (C_β-atom-distance < 8Å). Non-local contacts are amino acids in close three dimensional proximity that have a large separation in the primary amino acid sequence. This limitation is the direct result of simulating the folding of a protein by starting from an extended conformation. 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 below 150 residues routinely [31].

BCL::Fold uses 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. The lack of loop connectivity allows more sampling of different placements of SSEs and aims to overcome the size limitation. 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.

Use of Cryo Electron Microscopy as sparse experimental restraint

Cryo Electron Microscopy (cryoEM) is one of the newer techniques in structural biology to acquire insight on the assembly of macromolecular complexes [32]. A rapidly frozen sample of the specimen in question is subject to electron microscopy. Rapid freezing prevents the formation of crystals in the water, which would destroy the specimen. Additionally, it is fixed and can be subjected to imaging. The electron microscope takes an image of a two dimensional projection of the specimen. Since multiple specimen are fixed in different angles, many projections are acquired. A computational method can reconstruct a three dimensional image of the specimen depending on its structural variability and the quality and quantity of the experimental data acquired. The result is an electron density map, a three dimensional representation of the distribution of electrons around the macromolecule. Viruses and the ribosome have been imaged with this technique [33], [34]. Although, routinely only density maps of resolutions of 20Å are obtained, with experimental automation, resolutions higher than 9Å can be achieved [35]. Density maps of lower resolution can be used to localize domains in biological macromolecules. Starting at 9Å resolution helices can be identified, and at resolutions below 5Å strands are resolvable [36]. Density maps give invaluable information about the structure of the system. Features that are required by structural biologists to determine the function and interplay between the components are not readily retrievable. One approach to address this information retrieval problem is to fit atomic detail structures of the individual components into the electron density map.

Rigid body fitting

The rigid body fitting problem for atomic detail protein structures into electron density maps attempts to make the connection between the information that is given within the envelope defined by the electron distribution in the density map. This process connects the structural information that is available for individual components of the macromolecular assembly that is in question.

The objective in the rigid body fitting problem is to find the position within the electron density map that agrees optimally with the structure of the protein fitted. This objective is most commonly measured by the real space cross correlation coefficient between a synthesized density map for the protein in question to the position within the experimental density map. Two difficulties to overcome are: Sample all possible positions within the experimental electron density map and identify the best matching one. The most common technique tests iteratively positions probing 3 rotational degrees of freedom in inverse space by Fourier-transforming the density map and the other 3 translational degrees of freedom in real space. This algorithm is implemented in the widely used package SITUS [37].

BCL::EM-Fit

Current limitations in rigid body fitting are the required time and completeness. Sampling all positions in an electron density map to find the position with the highest cross correlation coefficient (CCC) is time consuming and inefficient. Not all regions in the electron density map contain density. Additionally, the density map already has a crude representation of the protein topologies that can be used to extract likely orientations.

A speed-up of the fitting does not only decrease the time to analyze the results of a cryoEM experiment, it also enables large scale *in silico* experiments. If the proteins or the three dimensional structures of the proteins within a macromolecular assembly are unknown, one could screen a databank of structures against the electron density map. Predicted protein structures can be probed to fit the density map – identifying the best structural model and its position within the map. This can yield not only insight into the composition of the macromolecular assembly, but can also define the protein interfaces, that contribute mostly to the stability of the complex.

BCL::EM-Fit introduces a rapid fitting method, that adopts the technique of geometric hashing to encode likely placements of objects within the density map before any search is started [1]. The same encoding is applied to the protein that is to be fitted, and the fitting is reduced to a lookup within a geometric hash. This reduces fitting time and increases completeness of the fitting problem, meaning that all highly likely placements of the protein model are a result of the algorithm. A Monte-Carlo-Metropolis optimization speeds up the refinement process to seek the local minima by CCC.

BioChemistry Library

BioChemistry Library (BCL) is an object oriented software library for scientific computing written in the programming language C++. It was started by Jens Meiler as the "own library" and assisted in many scientific projects: "DipoCoup: A versatile program for 3D-structure homology comparison based on residual dipolar couplings and pseudocontact shifts." [38], "Fast Determination of 13C-NMR Chemical Shifts Using Artificial Neural Networks." [39], "Generation and Evaluation of Dimension Reduced

Amino Acid Parameter Representations by Artificial Neural Networks.” introducing the JUFO amino acid sequence secondary structure prediction [14]. In 2005 Jens Meiler brought this software with him to start the laboratory at the Vanderbilt University and many materials served as a basis for the code that now constitutes the BCL.

The BCL is the basis for all computational projects in that thesis. Through a collaborative development of all graduate students, the effort for implementing complex algorithms in efficient code is shared. The development from the idea to the first test of the hypothesis is reduced significantly. The possibility to combine methods and algorithms from different fields are endless and helpful to develop new ideas. E.g. the collaborative implementation of GPGPU (general purpose graphical processing unit) code lead to the implementation of a rapid minimization protocol of protein structures within electron density maps using graphics cards [40].

After 6 years of development, the BCL is comprised of 600,000+ lines of code and comments, 3000+ files and a vast number of computational tools describing mathematical procedures, physical phenomena and chemical as well as biological objects. This collection of tools enables computational projects for biological and chemical research. Besides many tools that are available to researchers in the lab, several programs are at a stage where they have been distributed: BCL::Jufo – secondary structure prediction from the amino acid sequence, BCL::Contact residue-residue contact prediction from amino acid sequence, BCL::Cluster – a data analysis tool for protein structure and small molecule clustering in integration with the Pymol graphical visualization program [10], BCL::EM-Fit and BCL::EM-FitMinimize – rapid fitting of atomic protein structures into low resolution electron density maps [1], [40], BCL::Align – an amino acid sequence alignment tool , and BCL::PDBConvert – a tool for protein databank file handling.

The BCL Commons serves as a platform to distribute those programs to scientists http://bclcommons.vueinnovations.com/bclcommons. The Meiler lab website establishes remote server applications at http://www.meilerlab.org for the most prominent tools. Publications of new methods are synchronous with a webserver setup if suitable and the release of binaries under the BCL Commons. The webserver is free of charge to anybody; licensing through the BCL Commons grants access to binaries for onsite use and is free for academic users.

CHAPTER II

BCL::EM-FIT: RIGID BODY FITTING OF ATOMIC STRUCTURES INTO DENSITY MAPS USING GEOMETRIC HASHING AND REAL SPACE REFINEMENT.

This chapter is a reproduction of the identically titled first-author publication which appeared in the "Journal of Structural Biology" co-authored by Steffen Lindert, Phoebe L. Stewart and Jens Meiler [1].

Introduction

Cryo-electron Microscopy (cryoEM) [32] has evolved in the past decade as an important tool to obtain medium resolution structures of biological macromolecular assemblies in the form of density maps. One challenge is to dock high resolution experimental structures, obtained by X-ray crystallography [4] and nuclear magnetic resonance (NMR) [6], or models of individual proteins into these density maps to arrive at quasi atomic-detail representations of the macromolecular assembly. This procedure identifies regions of conformational change and regions that can be assigned to proteins of uncharacterized structure or which are characterized only in isolation.

Several protocols have been developed to fit atomic structures, usually obtained by X-ray crystallography or NMR, into low and medium resolution density maps [41], [42]. The computational problem amounts to determining six degrees of freedom, three rotational and three translational. Exhaustive searches systematically seek within this six-dimensional parameter space to optimize the cross correlation coefficient (*CCC*), which consumes significant amounts of computational time [43], [44]. Computational time can be reduced by the use of a fast Fourier transformation accelerated translational search as implemented in the "COLORES" program within the SITUS package [45]. In this approach only the three rotational degrees of freedom are searched in an exhaustive fashion in real space, while the translational degrees of freedom are searched in Fourier space. For both algorithms the step size impacts the speed of the calculation, but also the reliability and quality of the solution. An optimal local fit can be found with Chimera. It provides the benefit of a graphical user interface and an implementation of gradient refinement [46]. This refinement is only local and requires that the initial placement be closer to the correct solution than the protein diameter. Gradient based local minimization also have been implemented on general purpose graphical processing units (GPGPU) showing speed ups of at least 30 with the same accuracy as a CPU version [40].

To further increase the speed of fitting, vector quantization was introduced [37]. Single molecule data is represented by k so-called codebook vectors for high resolution protein structure data and low resolution density maps. In a search within the k! permutations the best fit is identified by the lowest residual $\mathrm{RMSD}_{C\alpha}$ after superimposition. This "QDOCK" method in the SITUS program is fast and reliable for rigid body docking and can be used for flexible docking as well. Difficulties arise however, if the density map contains different and multiple protein structures.

Protein structures obtained by X-ray crystallography often differ from the form of the protein observed in the cryoEM experiment. This can be the case if the protein was modified to facilitate crystallization or if a comparative model was built from a crystal structure of a homologous protein. In these cases the atomic model might not reflect all of the structural and dynamical properties observed in the cryoEM map. Therefore, flexible docking protocols were developed to overcome the limitations of rigid body fitting. For example, structural alignments of one protein to proteins in the same super family can be used to sample different conformations and improve the *CCC* [47]. Alternatively, normal mode based fitting varies the coordinates of the structure within reasonable limits while docking [48]. Molecular dynamics approaches have also been tested to optimize the fit of an atomic structure into electron density maps [49], [50]. Flexible docking can also be achieved by defining hinges between domains and varying the orientation between them using QDOCK in the SITUS package. Methods such as molecular dynamics, conjugate-gradient minimization, and Monte Carlo optimization can be integrated with different scoring functions in an iterative protocol that combines the strengths of each individual approach [51].

The present work implements for the first time a "geometric hashing" algorithm [52] termed BCL::EM-Fit for the task of fitting atomic-detail protein models into cryoEM densities. Geometric hashing was developed in the robotics field, where feature-recognition and pattern-matching give computers the ability to connect real life objects to abstract computational representations. This technique is already used in structural biology to identify similar binding sites in proteins [53]. A second step in the BCL::EM-Fit approach involves a Monte Carlo [54]/Metropolis [55] (MCM) small perturbation protocol to refine the initial fits by maximizing the *CCC*. The time and robustness of BCL::EM-Fit compares favorably with the widely used Fourier/real space fitting program "COLORES" in the SITUS package [37]. Benchmark results are presented with simulated density, as well as examples that demonstrate fitting with experimental GroEL density

[56] and of adenovirus capsid protein crystal structures into experimental cryoEM density maps [35].

Results

An efficient two-stage low and high resolution fitting protocol

The BCL::EM-Fit protocol consists of several steps including geometric hashing to find initial fits, and Monte Carlo/Metropolis (MCM) optimization for refinement (Figure 1). Features are extracted from the density map and stored in a hash map (either in computer memory or a databank, see also Figure 2a–c). The fitting procedure involves feature extraction from the atomic protein structure and comparison with saved features from the density map. The best initial fits are determined by counting matching quantized features between the atomic structure and density map (see also Figure 2d-g). Finally, a MCM optimization step is used to refine the initial fits based on real space *CCC*. The following paragraphs give a brief summary of the major steps. Implementation details are discussed in the Methods section.

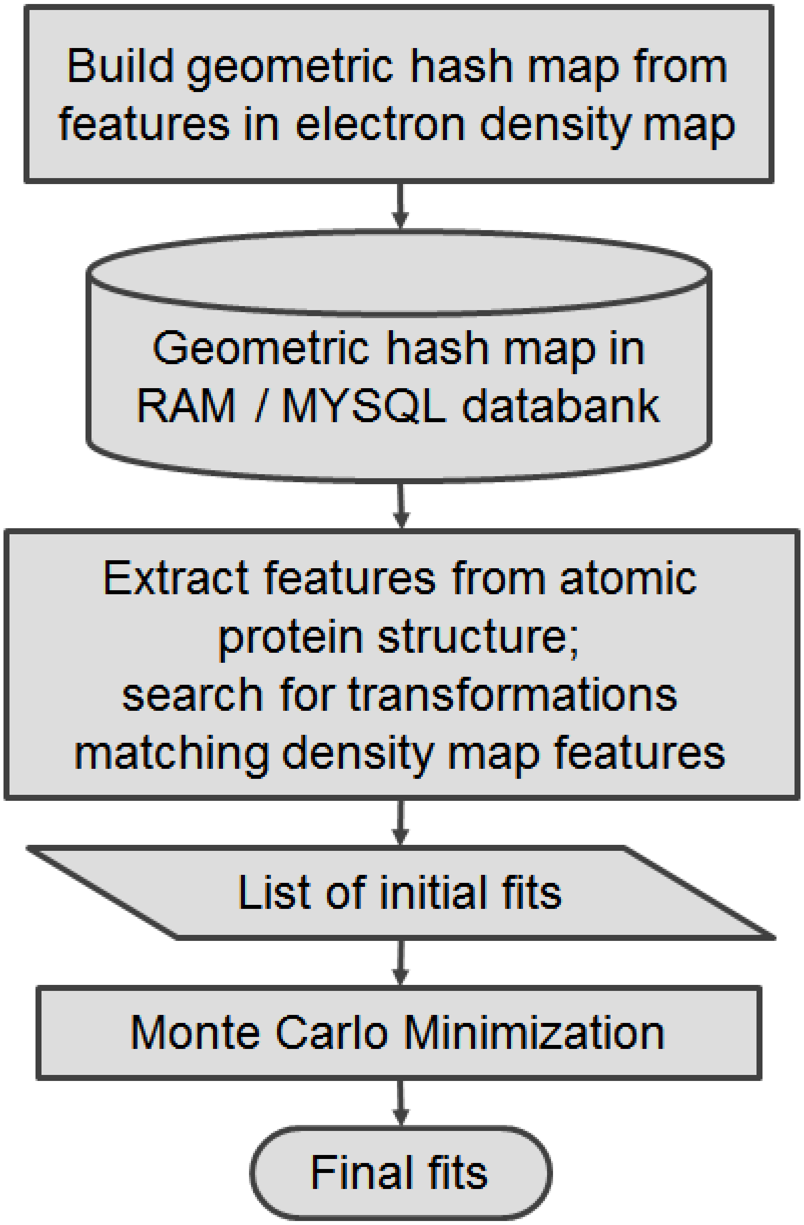
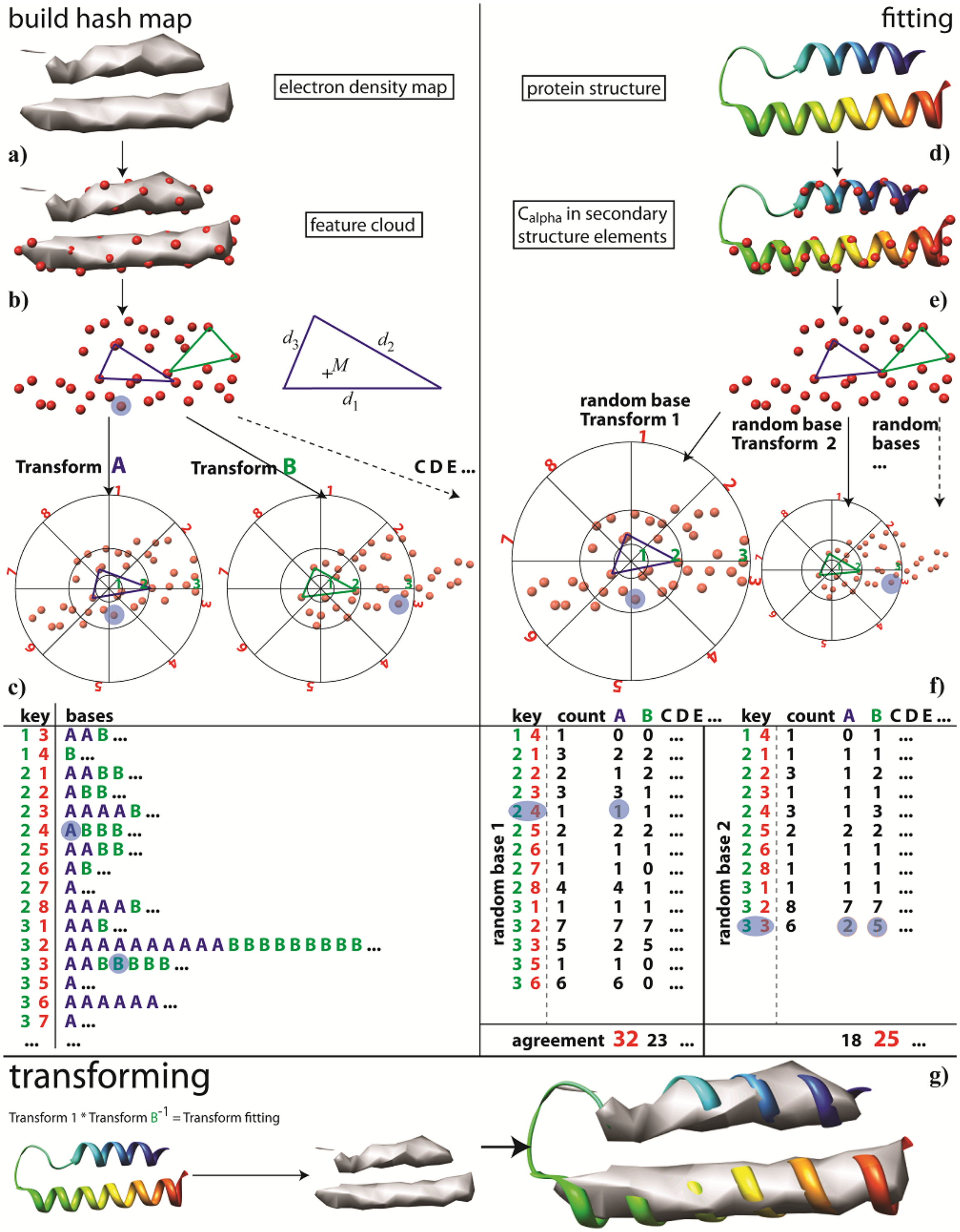

Build geometric hash map from features in electron density map
Geometric hash map in RAM / MYSQL databank
Extract features from atomic protein structure; search for transformations matching density map features
List of initial fits
Monte Carlo Minimization
Final fits

Figure 1 Schematic flowchart of BCL::EM-Fit

The general scheme of BCL::EM-Fit starts with the extraction of geometric features from the density map. These features are transformed into different orientations and saved together with their respective transformation in a hash map that is stored in computer RAM or in a MySQL databank. This process must be completed once for an experimental density map. In order to dock an atomic structure representative features are extracted from the coordinate set and compared to the hash map. The geometric hashing algorithm identifies a list of transformations that maximize the number of shared features between density map and atomic structure. Each of these initial fits is optimized in a MCM refinement step.

In the first step the density map is converted into a feature cloud using several user inputs, such as the number of structural features expected in the density map and minimal distance between structural features (Figure 2a). Regions of high intensity and with large intensity gradients are automatically selected from the density map. High intensity regions describe the centers of structural features, such as observed density rods for α-helices, which typically have high intensity values. Large gradients are observed along iso-surfaces of structural features and can be thought of as points along structural edges. This information is stored in a feature cloud corresponding to the selected Voxel (volume pixel) centers. Within this feature cloud triangular bases are selected according to minimal and maximal distances between the three points (Equation (2) and Figure 2b). These triangular bases serve as a coordinate framework in which all other features of the cloud are expressed. Each triangular base is described by a unique transformation consisting of three rotational and three translational parameters. After transforming the feature cloud for each triangular base, the features within a given feature radius (Equation (3)) are quantized (Equation (4) – Equation (6)) and stored in a geometric hash map together with the respective transformation (Figure 2c). The feature radius is chosen depending on the dimensions of the atomic structures to be fitted. This procedure effectively stores the feature cloud as seen from many different perspectives in space. This preprocessing procedure is only performed once for a given density map.



key	bases
1 3	A A B ...
1 4	B ...
2 1	A A B B ...
2 2	A B B ...
2 3	A A A A B ...
2 4	A B B B ...
2 5	A A B B ...
2 6	A B ...
2 7	A ...
2 8	A A A A B ...
3 1	A A B ...
3 2	A A A A A A A A A A B B B B B B B B B ...
3 3	A A B B B B B ...
3 5	A ...
3 6	A A A A A A ...
3 7	A ...
...	...

random base 1

key	count	A	B	C D E ...
1 4	1	0	0	...
2 1	3	2	2	...
2 2	2	1	2	...
2 3	3	3	1	...
2 4	1	1	1	...
2 5	2	2	2	...
2 6	1	1	1	...
2 7	1	1	0	...
2 8	4	4	1	...
3 1	1	1	1	...
3 2	7	7	7	...
3 3	5	2	5	...
3 5	1	1	0	...
3 6	6	6	0	...
agreement		32	23	...

random base 2

key	count	A	B	C D E ...
1 4	1	0	1	...
2 1	1	1	1	...
2 2	3	1	2	...
2 3	1	1	1	...
2 4	3	1	3	...
2 5	2	2	2	...
2 6	1	1	1	...
2 8	1	1	1	...
3 1	1	1	1	...
3 2	8	7	7	...
3 3	6	2	5	...
		18	25	...

Figure 2 Detailed flowchart of geometric hashing protocol

The geometric hashing protocol is illustrated with an example protein structure and its density map in two dimensions. Building the hash map starts with (a) extracting a feature cloud from the density map. (b) Each possible combination of three features represents a triangular base with the sides d_1, d_2, and d_3. Triangles that satisfy represent a base that is transformed to be the origin of a new coordinate system. (c) All remaining points that satisfy Equation (3) in terms of their distance to the base (outermost circle) are transformed and quantized using a spherical coordinate system (Equation (4) – Equation (6)). Quantized coordinates are stored in the hash map with the respective triangular base. The blue highlighted point will occur in the hash map multiple times affiliated with different keys and different bases. Steps (a) – (c) are performed once for every density map. (d) The fitting starts with extracting features from the protein structure i.e. C_α-atoms in α-helices. (e) Subsequently random bases are picked in this feature cloud and all features of the protein structure are transformed with respect to these random bases. (f) Now, all keys affiliated with a random base are looked up in the hash map. From this procedure original triangular bases are identified that share a maximum number of keys. Each shared key represents one agreeing feature between protein and density map and increments the hash score by one. The blue highlighted point adds to the agreement, if it corresponds to the matching base in the hash map. (g) The transformations with the highest hash scores will be chosen as the best initial fit.

In order to fit a given atomic model into the previously encoded density map, a user-defined subset of backbone atoms (C_α, N, O, or C) within secondary structure elements must be extracted from the full coordinate file (Figure 2d) (see details in Methods section). The rationale for using only backbone atoms is that these atoms are usually close to the edges of high-density regions in the density map and typically define edges of regular secondary structure elements such as α-helices. From within this set of atoms three features are chosen as a triangular base and all other features are transformed so that the triangular base ends up at the origin (Figure 2e). The transformed features within the feature radius are quantized and then searched for within the hash map representation of the density map (Figure 2f). The geometric hashing algorithm results in the identification of transformations that superimpose a maximum number of features between the atomic resolution model and the density map (Figure 2g). Henceforth the maximum number of superimposable features will be termed the "hash score".

In the second stage of the BCL::EM-Fit protocol, a small number of top scoring initial placements are refined with MCM optimization applying rotational and translational perturbations (Figure 3). The real space CCC (Equation (7)) is maximized between a simulated density map based on the atomic model and the experimental density map. The refined placements are ranked by *CCC*.

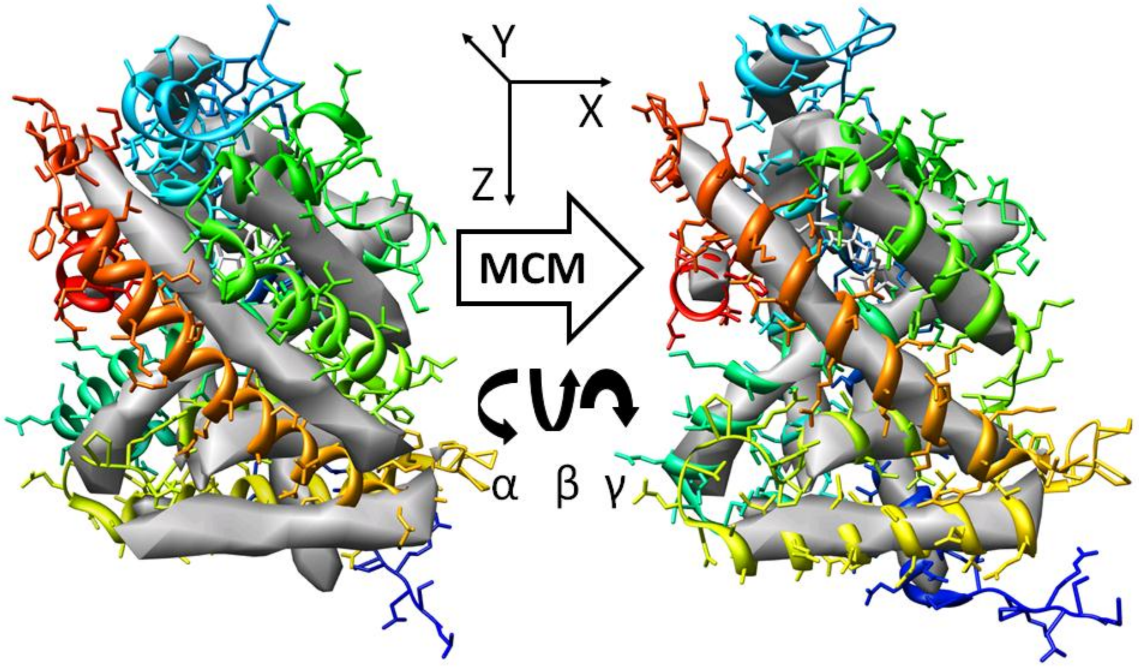


Figure 3 MCM refinement through a real-space rigid body six-dimensional search

Schematic representation of the Monte Carlo Metropolis (MCM) refinement step in which rigid body movements (translations in X, Y, and Z and rotational changes around α, β, and γ) are applied to the atomic protein structure relative to the density map in order to maximize *CCC*. After each movement the *CCC* between the experimental density and the simulated density map (derived from the atomic protein structure) is calculated.

Protein fitting procedure is highly reliable for resolutions of 10 Å or better

In order to evaluate the reliability of the BCL::EM-Fit algorithm a benchmark was performed with 21 α-helical, 7 β-strand and 22 α/β proteins (Table 1). Specific parameters can be found in the Methods section. Figure 4 presents the BCL::EM-Fit results for all of the benchmark proteins fit within their simulated density maps with various noise levels as a function of resolution (5-19 Å). The results were analyzed for each atomic model/simulated map combination to see if at least one of the initial 10 best fits by hash score was refined by MCM to have a final placement with an $\mathrm{RMSD}_{C\alpha}$ value of < 5 Å with respect to the correct position. Note that for the set of α-helical benchmark proteins fit within the noise-free maps, essentially all of the BCL::EM-Fit runs resulted in at least one MCM refined fit with an $\mathrm{RMSD}_{C\alpha} < 5$ Å. This is shown in Figure 4a as black bars with heights of 20%, or close to 20%, at all resolutions in the range of 5-19 Å. Since the noise-free maps represent 20% of the total maps tested, this level represents the fact that a correctly fit solution was found for almost all atomic model/simulated density combinations in the α-helical benchmark proteins category using noise-free maps. As the plot indicates, the BCL::EM-Fit results are not quite as good with the noise-added maps. Nevertheless, an overall success rate of 90% is achieved for the α-helical benchmark proteins with simulated density maps up to ~ 14 Å resolutions. The BCL::EM-Fit results for the set of α/β benchmark proteins (with more than 2 helices and 2 strands in the structure) indicate an overall success rate of 90% with simulated density maps up to ~11 Å resolution (Figure 4b). The β-only benchmark proteins were the most challenging, with a 70% success rate up to ~10Å resolution (Figure 4c).

Table 1 Overview of benchmark proteins.

All 50 benchmark proteins are listed with their PDB-ID, sorted by size and with some relevant statistics. They have been selected to vary in size from 150 to 350 amino acids and to be single chain biological molecules without ligands.

PDB	α or β	SCOP	CATH class	CATH architecture	#α-helices	#β-strands	#amino acids
1x91	α	all α	-	-	6	-	153
1icx	αβ	α + β	αβ	2-layer sandwich	6	7	155
1jl1	αβ	α/β	αβ	2-layer sandwich	5	5	155
1bj7	β	all β	mainly β	β-barrel	5	9	156
2yv8	β	-	-	-	1	13	164
3gbw	β	-	-	-	0	12	164
1gs9	α	all α	mainly α	up-down bundle	5	-	165
1bgc	α	all α	mainly α	up-down bundle	5	-	174
1wba	β	all β	mainly β	trefoil	-	12	175
1xqw	α	-	mainly α	α-horseshoe	10	-	176
1lki	α	all α	mainly α	up-down bundle	6	2	180
1vgi	αβ	-	-	-	5	9	184
1nfn	α	all α	mainly α	up-down bundle	5	-	191
2qvk	β	-	-	-	-	10	192
1dus	αβ	α/β	αβ	3-layer(αβα) sandwich (rossmann)	6	9	194
2osa	α	-	-	-	12	-	202
1chd	αβ	α/β	αβ	3-layer(αβα) sandwich (rossmann)	10	9	203
1xkr	αβ	α + β	αβ	3-layer(αβα) sandwich	6	8	206
2iu1	α	-	-	-	13	-	208
1iap	α	all α	-	orthogonal bundle	11	-	211
1oa9	β	all β	mainly β	β-barrel	9	7	214
2fm9	α	all α	-	-	10	-	215
1uai	αβ	all β	mainly β	sandwich	2	16	224
1oxf	β	α + β	mainly β	β-barrel	6	11	225
1wnh	αβ	-	-	-	4	10	225
1g8a	αβ	α/β	αβ	2 domains (CATH) - 2-layer sandwich & 3-layer(αβα) sandwich - RNA binding protein	7	12	227
1wr2	αβ	-	-	-	9	10	238
3b5o	α	-	-	-	13	-	244
1prz	αβ	α + β	-	ab	6	11	252
1qkm	αβ	all α	mainly α	orthogonal bundle	13	2	255
2e3s	αβ	-	-	-	5	12	255
1xqo	α	all α	mainly α	2 domains (CATH) - orthogonal bundle	14	-	256
2ax6	αβ	all α	mainly α	orthogonal bundle	12	4	256
1ie9	αβ	all α	mainly α	orthogonal bundle	14	3	259
2yvt	αβ	-	-	-	9	14	260
2ilr	α	-	-	-	17	-	264
2of3	α	-	-	-	19	-	266
1n83	αβ	all α	mainly α	orthogonal bundle	12	3	270

1ouv	α	all α	mainly α	alpha horseshoe	15	0	273
1uek	αβ	α + β	αβ	2-layer sandwich	11	10	275
2opw	αβ	-	-	-	10	15	291
2zco	α	-	-	-	18	-	293
1gcu	αβ	α + β	αβ	2 domains (SCOP + CATH) 3-layer(αβα) sandwich (rossman fold) & 2-layer sandwich	14	13	295
1vk4	αβ	α/β	Aβ	3-layer sandwich	15	15	298
2cl2	αβ	-	-	-	12	19	298
1lkf	β	membrane and cell surface proteins	mainly β	distorted sandwich	3	18	299
2cwc	αβ	-	-	-	18	18	303
1v9m	α	membrane and cell surface protein	-	-	21	-	323
1z1l	αβ	-	-	-	23	-	345

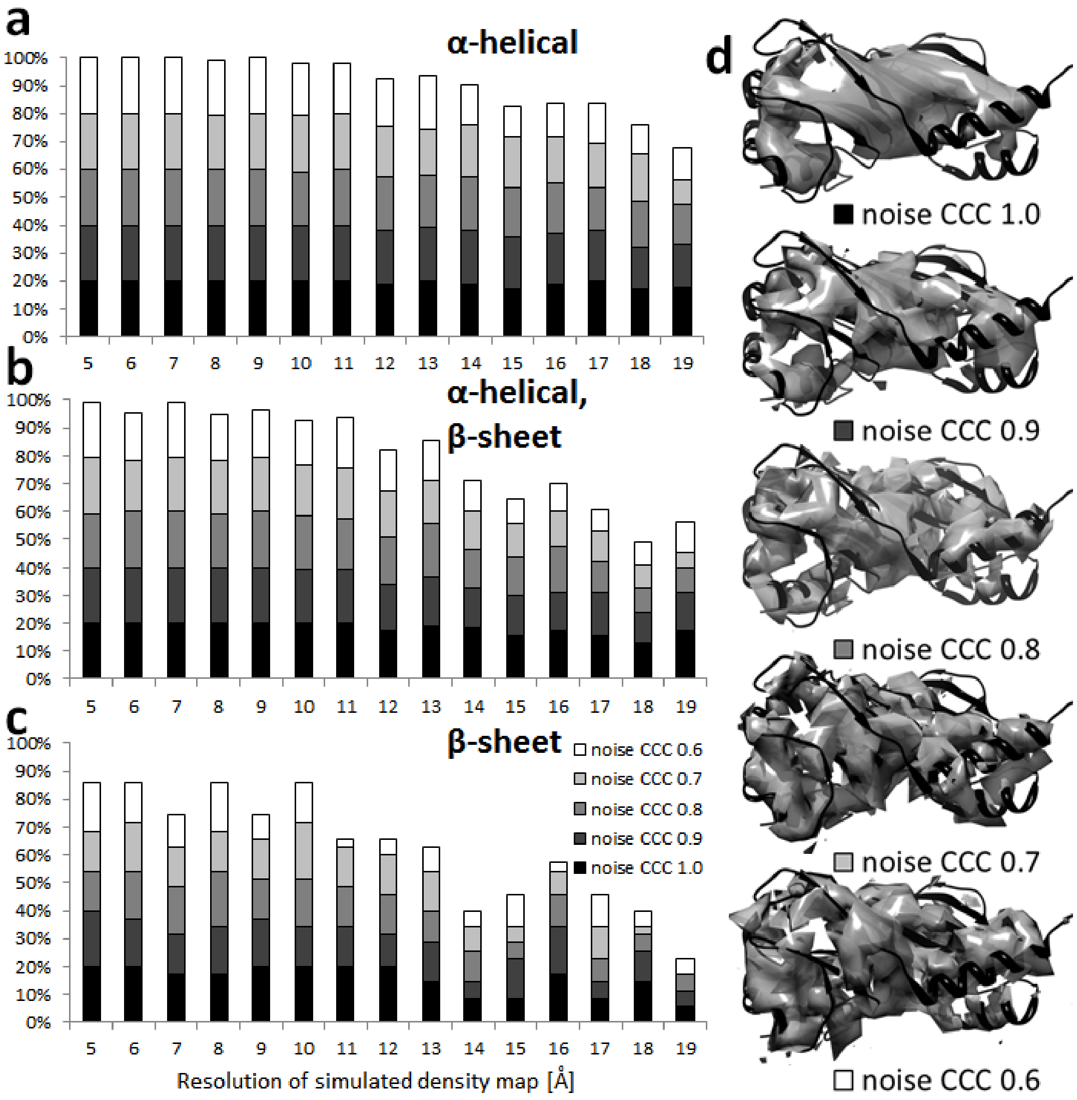


Figure 4 Fitting of benchmark proteins at different resolutions

(a) Results of fitting 21 α-helical proteins into simulated density maps calculated in the resolution range of 5 to 19 Å both with and without added noise. The *CCCs* of the noise-added maps to the noise-free maps are 0.9, 0.8, 0.7 and 0.6. The x-axis represents the resolution of the simulated density map in Å. The y-axis represents the percentage of atomic model/simulated map combinations that had at least one fit within the initial 10 best fits by hash score that refined to the correct position (within RMSDCα < 5 Å). The results with noise-free maps (noise *CCC* 1.0) are plotted with black bars, and those with noise-added maps are plotted in shades of gray to white. The maximum height of any bar (noise-free, or with noise) is 20%, corresponding to the percentage for that category of maps. (b) Results of fitting 22 α/β proteins. (c) Results of fitting 7 β-sheet proteins. (d) Simulated density maps for one of the α/β benchmark proteins (1prz) at 10 Å resolution with and without added noise shown together with the input atomic structure.

As the results presented in Figure 4 show, there are combinations of atomic models and simulated density maps for which refinement of the initial 10 best fits by hash score did not result in any correct final positions (i.e., within $RMSD_{C\alpha} < 5$ Å). However, the trends reflected in Figure 4 indicate that fitting failures occur with greater frequency when simulated maps with higher noise levels or of lower resolution are used. This implies that at a certain point the simulated density maps lack a sufficient number of unique features for this method to find the correct fit of the atomic model within the best 10 placements.

In general, these benchmark tests show that α-helical proteins are fitted with higher success rates than α/β-proteins, followed by β-strand proteins. It should be noted that these benchmark tests were designed to reveal the theoretical limits of the hashing algorithm and the MCM protocol. Admittedly, the benchmark tests were performed with single protein molecules in isolation and do not reflect the results one might expect when there are neighboring molecules or symmetry related subunits present in the density map. Also other than Gaussian noise, no attempts were made to mimic additional sources of error that might be present in an experimental cryoEM density map. These include errors due to conformational flexibility and heterogeneity. However, these benchmarks do show that the BCL::EM-Fit protocol performs well for isolated α-helical proteins, mixed α/β and β-strand proteins, albeit with different resolution limitations. In addition, they can serve as a useful guide for the experimentalist regarding the resolutions that may be required for robust fitting of atomic coordinates for α-helical proteins, mixed α/β and β-strand proteins.

BCL::EM-Fit identifies the correct density for a given atomic resolution structure, homolog, or comparative model

Often atomic resolution structures of proteins are placed into cryoEM density maps of macromolecular systems in order to assign density regions to specific proteins. This proves even more challenging if no experimental atomic resolution structure is available and the structure of a homolog or comparative model is used. To test the robustness of the algorithm in this respect a *cross-fitting* experiment was performed where 9 of the α-helical benchmark proteins were fitted into all 12 Å resolution noise-free density maps (Table 2). The experiment was repeated for 6 β-strand proteins with 11 Å resolution noise-free density maps (Table 3). In all cases the correct match was identified with *CCCs* of 1.00. The best fit into a wrong density map never had a CCC higher than 0.95.

This experiment was repeated using homologous structures, identified by bioinfo.pl metaserver [57], and comparative models generated by MODELLER [58], for three of the α-helical and two β-strand benchmark proteins (Table 4). Density maps were generated with a resolution of 11 Å and with noise levels designed to yield *CCCs* of 0.8 with respect to the noise-free maps. All but one homologous structure showed the highest *CCC* when fitted into the density of the respective homologous protein (Table 4left). The exception is 1LN1, which is a homolog of β-strand protein 2E3S. In tests with the 1LN1 atomic coordinates, roughly equivalent *CCC* values (0.73 to 0.75) were found after docking into four different simulated maps. One of these four maps was the intended simulated map for the homolog 2E3S, but there was not a clear peak in the *CCC* value with the correct simulated density map (Table columns). Similarly, the simulated density map for 2E3S had high correlations (0.71-0.75) with coordinates of 3 non-homologues structures (Table rows). The lesson implied by these results, which is not unexpected, is that some protein folds will be more difficult to fit than other folds at certain resolution cutoffs.

Comparative models were built with MODELLER using these same homologous structures as templates and the bioinfo.pl alignment. Details are given in the Methods section. For all comparative models the highest *CCC* value was found for the correct density map, as indicated by the diagonal (Table 4right). Correct placement of the model into the density was validated by visual inspection. Although the comparative models did not have a significantly higher *CCC* for the fitted structures compared to the values found for the homologous structures (Table 4 compare left and right), the comparative models were fit unambiguously to the correct density maps.

Table 2 Cross fitting matrix for the α-helical proteins and 12 Å resolution simulated density maps.

9 benchmark α-helical proteins (horizontal) were docked into simulated density maps at 12 Å resolution (vertical). *CCCs* above 0.95 are in bold; with additional coefficients above 0.90 in italics. In each case the highest correlation value was found for the correct fit, as indicated by the numbers along the diagonal.

mrc\pdb	**1IE9**	**1N83**	**1OUV**	**1QKM**	**1V9M**	**1XQO**	**1Z1L**	**2AX6**	**2CWC**
1IE9	**1.00**	*0.95*	0.75	*0.95*	0.90	*0.94*	0.90	*0.95*	0.89
1N83	*0.95*	**1.00**	0.72	*0.93*	0.87	*0.91*	0.90	*0.94*	0.89
1OUV	0.74	0.72	**1.00**	0.70	0.68	0.73	0.74	0.71	0.64
1QKM	0.90	*0.93*	0.71	**1.00**	0.89	*0.92*	0.90	***0.96***	*0.91*
1V9M	0.90	0.87	0.71	0.89	**1.00**	*0.91*	*0.93*	0.90	*0.94*
1XQO	*0.94*	*0.92*	0.74	*0.93*	*0.92*	**1.00**	*0.91*	*0.94*	*0.92*
1Z1L	*0.91*	0.90	0.75	0.90	*0.93*	*0.91*	**1.00**	*0.91*	*0.92*
2AX6	*0.93*	*0.94*	0.72	0.96	0.90	*0.94*	0.90	**1.00**	*0.92*
2CWC	0.90	0.88	0.65	*0.91*	*0.94*	*0.93*	*0.91*	*0.92*	**1.00**

Table 3 Cross fitting matrix for the β-strand proteins and 11 Å resolution simulated density maps.

5 benchmark β-strand proteins (horizontal) were docked into simulated density maps at 11 Å resolution (vertical). *CCCs* above 0.95 are in bold; with additional coefficients above 0.90 in italics. In each case the highest *CCC* value was found for the correct fit, as indicated by the numbers along the diagonal.

mrc\pdb	**1IFB**	**1LKF**	**1OXF**	**1UAI**	**2CL2**	**2E3S**
1IFB	**1.00**	0.76	*0.91*	*0.91*	0.84	0.87
1LKF	0.76	**1.00**	0.79	0.78	0.82	0.77
1OXF	*0.91*	0.82	**1.00**	*0.92*	*0.93*	0.89
1UAI	*0.91*	0.78	*0.91*	**1.00**	0.90	*0.92*
2CL2	0.86	0.81	*0.93*	*0.91*	**1.00**	*0.92*
2E3S	0.87	0.79	0.92	*0.92*	0.90	**1.00**

Table 4 Cross-docking *CCC* matrix for benchmark proteins with homologous structures and comparative models.

Density map[a]	**Homologous structures**[b]					**Comparative models**[c]				
	1RJK	1PVL	1L2J	1T5J	1LN1	1RJK	1PVL	1L2J	1T5J	1LN1
%seq sim.	91	71	98	26	17					
CATH	α	β	α	α	αβ	α	β	α	α	αβ
#residues	292	301	271	313	214	259	299	255	303	255
helix/strand	13/3	3/22	12/2	20/2	6/17	10/3	1/19	8/2	14/0	6/10
$\mathrm{RMSD}_{C\alpha}$[d]	2.75	1.51	2.58	3.13	3.92	3.16	1.09	1.68	3.48	5.35
SSE-$\mathrm{RMSD}_{C\alpha}$[e]						0.42	0.65	0.91	1.52	3.03
1IE9	0.82	0.68	0.74	0.70	0.73	0.81	0.67	0.74	0.70	0.70
1LKF	0.68	0.83	0.66	0.62	0.63	0.68	0.82	0.67	0.62	0.60
1QKM	0.68	0.63	0.82	0.72	0.73	0.77	0.63	0.81	0.73	0.71
2CWC	0.72	0.58	0.73	0.79	0.75	0.72	0.58	0.73	0.81	0.74
2E3S	0.73	0.60	0.71	0.75	0.73	0.71	0.61	0.74	0.74	0.78

[a] Simulated density maps for five proteins: three α-helical (1IE9, 1QKM, 2CWC) and two β-strand (1LKF, 2E3S) at 11 Å resolution and with added noise (CCC 0.8 with respect to noise-free map).

[b] Homologous structures were identified with Bioinfo.pl.

[c] Comparative models were built for 1IE9, 1LKF, 1QKM, 2CWC, and 2E3S from the homologous structures (1RJK, 1PVL, 1L2J, 1T5J, and 1LN1, respectively) using Modeller.

[d] RMSDCα of the original PDB vs. the homologous structure (using mammoth structure alignment) and vs. the comparative model

[e] SSE-RMSDCα only using secondary structure elements that are common to both PDBs

Adenovirus capsid proteins are docked with high confidence into cryoEM density

The crystal structures for two adenovirus capsid proteins were docked into two experimental cryoEM density maps of adenovirus at 6.8 Å [2], [35] and 9.0 Å resolution [59] (FSC 0.5 criterion). Note that the 6.8 Å resolution map is of the Ad35F (Ad5.F35) vector, which contains human adenovirus type 5 (HAdV5) hexon and penton base capsid proteins. The 9.0 Å resolution map is HAdV12 in complex with integrin and is based on a subset of the full dataset in order to limit the resolution to 9 Å. The penton base structure (pdb: 1X9T) [60] is a homopentamer (2615 residues) formed by an N-terminally truncated form of HAdV2 penton base (residues 49-571) together with a 21 residue N-terminal tail of the HAdV2 fiber. The hexon structure (pdb: 1P30) [61] is a homotrimer (2853 residues) with 951 residues per monomer of the HAdV5 hexon. The hexon and penton base proteins from HAdV2, 5, and 12 are all highly homologous, with percent identities in the range of 77% to 99%.

The penton base fitting experiments were performed using comparable segments from the same location in the two different resolution cryo-EM density maps. The segments contained one tightly cut copy of the penton base oligomer. Due to the five-fold symmetry of the penton base five distinct fitting positions are possible. Three different fits within 7 correct solutions were identified by BCL::EM-Fit among the best 10 scoring fits for the 6.8 Å segment (Figure 5a), 2 different fits were identified among the 10 best scoring fits within the 9.0 Å density segment. *CCC* values between 0.06 and 0.31 were found for the 6.8 Å segment and *CCCs* between 0.02 and 0.54 for the 9.0 Å segment before the refinement (Table 5).

The MCM refinement procedure was performed on the 10 top-scoring initial placements to optimize the *CCC* further. Details are given in the Methods section. For 7 of the initial placements the *CCC* was optimized to 0.53 or better for the 6.8 Å segment; two placements were refined to *CCC* 0.66 for the 9.0 Å segment (Table 5). The accurate placement of the penton base was confirmed visually (6.8 Å segment is shown in Figure 5b). Comparison of the initial and refined positions for the atomic coordinates yields $\mathrm{RMSD}_{\mathrm{C}\alpha}$ values in the range of 6.2 – 11.6 Å, indicating movements on this order during refinement.

Table 5 Docking of penton base into adenovirus cryoEM density maps at 6.8 and 9.0 Å resolution with BCL::EM-Fit

Map resolution [Å]	rank by hash score	Hash score	Initial *CCC*	Optimized[b] *CCC*	RMSD$_{C\alpha}$[c] of optimized to initial fit [Å]
6.8	**5[a]**	**181**	**0.18**	**0.54**	**11.59**
6.8	**1[a]**	**192**	**0.31**	**0.53**	**6.19**
6.8	**4[a]**	**182**	**0.29**	**0.53**	**6.32**
6.8	**6[d]**	**181**	**0.18**	**0.53**	**10.03**
6.8	**10[d]**	**179**	**0.19**	**0.53**	**12.23**
6.8	**3[d]**	**186**	**0.30**	**0.53**	**6.65**
6.8	**2[d]**	**191**	**0.27**	**0.53**	**8.29**
6.8	8	181	0.14	0.16	6.07
6.8	9	180	0.10	0.12	6.91
6.8	7	181	0.06	0.10	7.49
9.0	**1[a]**	**128**	**0.54**	**0.66**	**9.28**
9.0	**2[a]**	**128**	**0.48**	**0.66**	**11.29**
9.0	4	126	0.15	0.32	16.59
9.0	3	127	0.29	0.31	2.73
9.0	6	125	0.19	0.31	17.58
9.0	8	125	0.12	0.18	11.83
9.0	7	125	0.10	0.13	8.80
9.0	9	125	0.02	0.12	12.31
9.0	10	125	0.04	0.12	12.53
9.0	5	126	0.05	0.12	13.18

[a] Best independent fits after MCM optimization by *CCC*. The three best fits that yield different placements with respect to the 6.8 Å resolution map are shown in Figure 5a.

[b] MCM refinement (see Methods). The refined positions of the three best independent fits with respect to the 6.8 Å resolution map are shown in Figure 5b.

[c] The RMSD$_{C\alpha}$ of initial to refined fit is shown to indicate the amount of movement of the atomic model during the refinement step.

[d] Fits which duplicate positions of the three best fits marked [a].

[a,d] All of the fits that are correct have a high *CCC* value after optimization (bold).

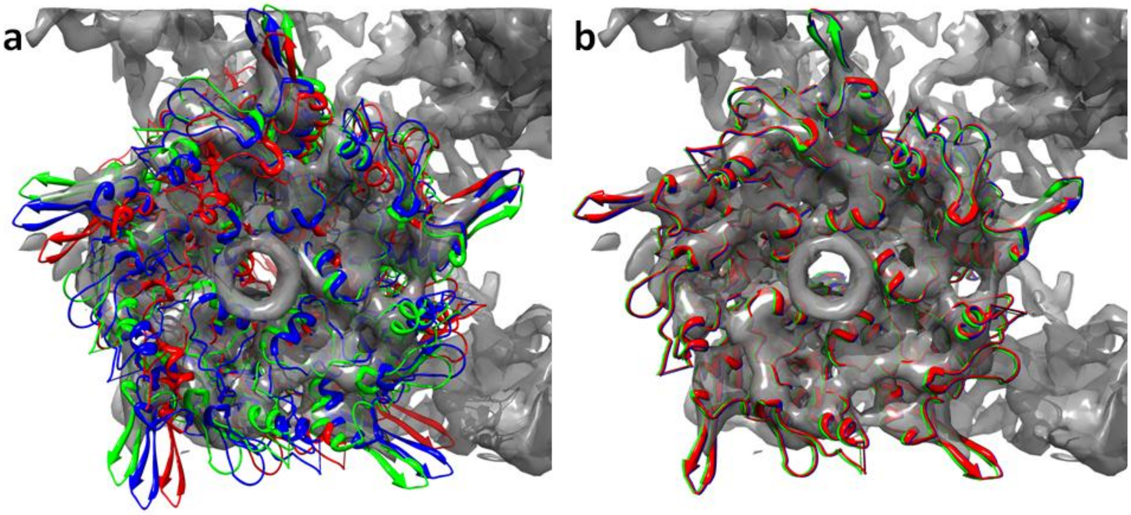


Figure 5 BCL::EM-Fit docking of penton base into adenovirus cryoEM density map segment at 6.8 Å resolution.

(a) The best three unique fits out of ten initial fits by *CCC* are shown docked into the cryoEM density segment (gray) displayed with an isosurface level chosen to reveal the strongest density features. (b) The same three fits after 250 steps of MCM refinement. The optimal placement of all three fits is confirmed visually by the good superimposition of α-helices with density rods.

The hexon capsid protein was docked into different segments of the reconstructed adenovirus density maps at 6.8 Å and 9.0 Å resolution, which contained all four independent positions of this protein within the asymmetric unit. Seven correct placements were identified in the 6.8 Å resolution density segment, of which four represent symmetrically independent, non-overlapping positions in the asymmetric unit (Table 6). These four initial fits have *CCCs* above 0.13, with the best being 0.25. Figure 6 shows superimpositions of the transformed hexon with the 6.8 Å resolution density segment confirming correct placements for this protein. A MCM refinement was performed on the 50 best initial placements. After optimization the *CCCs* for the symmetrically unrelated copies were in the range of 0.47 to 0.48 (Table 6 and Figure 7). Ten correct placements were identified in the 9.0 Å resolution density segment, of which four are symmetrically independent and non-overlapping positions. These four positions have *CCCs* above 0.53. After MCM refinement *CCCs* are between 0.68 and 0.73 (Table 6).

The adenovirus capsid protein fitting experiments indicate that the BCL::EM-Fit algorithm can identify initial fits of the atomic structures in question. The subsequent MCM refinement procedure delivers results in visually improved fits with higher CCCs.

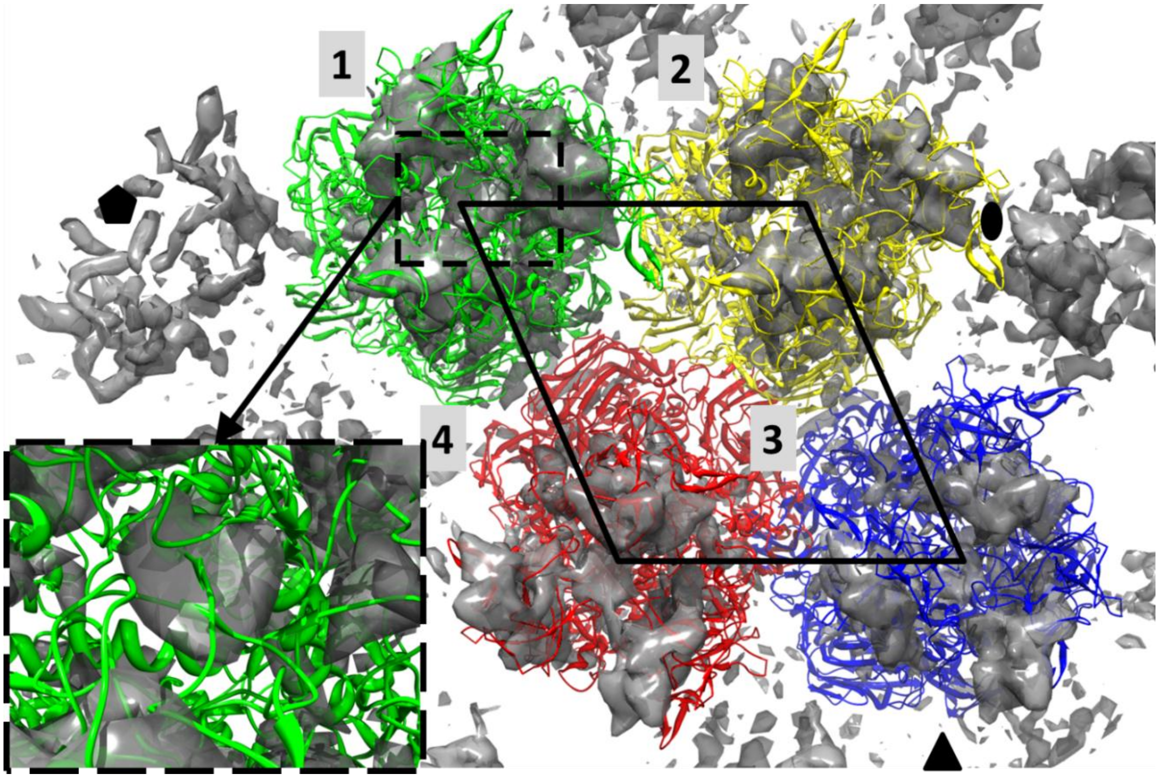


Figure 6 BCL::EM-Fit docking of hexon into a segment of an adenovirus 6.8 Å resolution cryoEM density map after the initial fit step.

The best four out of 50 initial fits (by *CCC*) for the hexon protein of adenovirus are shown docked within a cryoEM density segment (gray) at an isosurface level chosen to emphasize secondary structure elements. One asymmetric unit of the icosahedral capsid is outlined. The four unique hexon positions within the asymmetric unit are numbered 1-4 (1:green, 2:yellow, 3:blue, 4:red). Crystallographic symbols are shown for the 2-fold and 3-fold and 5-fold icosahedral axes. An enlarged view (box in lower left corner) shows a slab of density through one hexon (~30 Å thick).

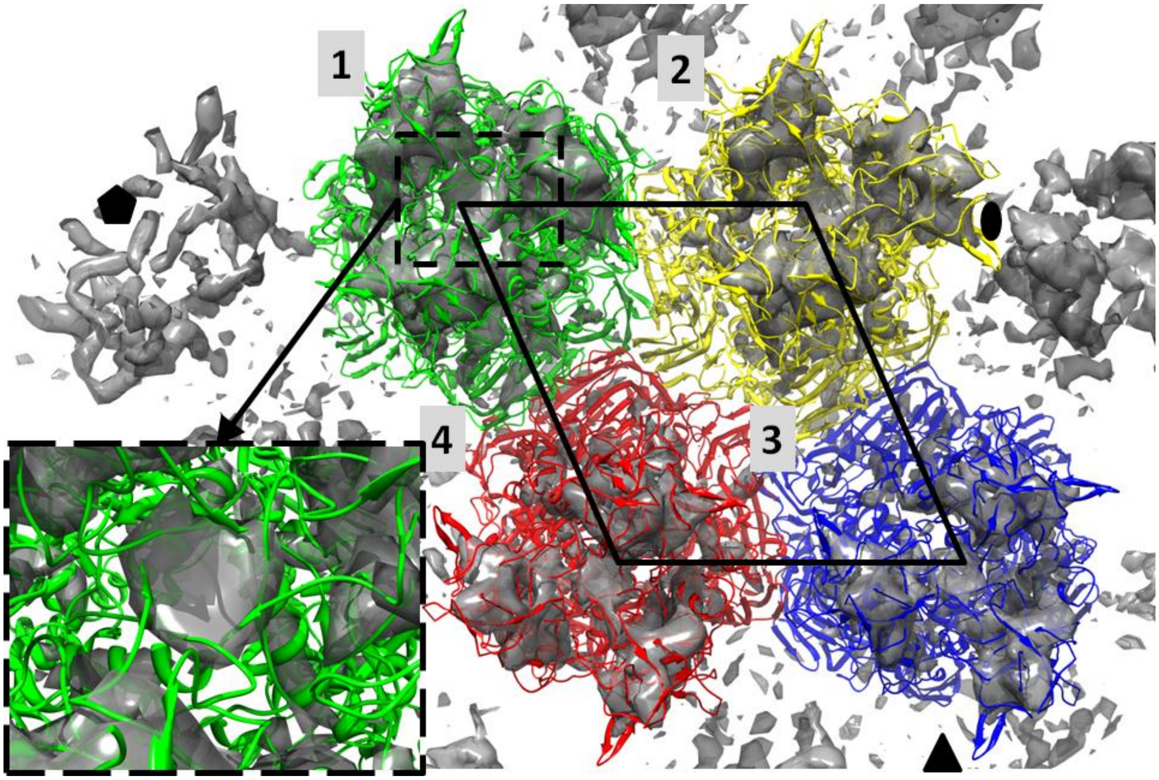


Figure 7 BCL::EM-Fit docking of hexon into a segment of an adenovirus 6.8 Å resolution cryoEM density map after the MCM refinement step.

The best four out of 50 initial fits (by *CCC*) for the hexon protein of adenovirus are shown in their final positions, after 250 steps of MCM refinement, docked within the cryoEM density segment (gray) at an isosurface level chosen to emphasize secondary structure elements. The asymmetric unit, hexon positions, and symmetry axes are indicated as in An enlarged view (box in lower left corner) shows a slab of density through one hexon (~30 Å thick). Note the after MCM refinement a better alignment is noted for α-helices with respect to density rods (compare with Figure 6).

Table 6 Docking of hexon into adenovirus cryoEM density maps with BCL::EM-Fit

Best ten placements by CCC after initial fit of the hexon protein of adenovirus into 6.8 Å and 9.0 Å resolution sections of the adenovirus cryoEM density maps. The best 50 initial fits by CCC were optimized in the MCM refinement with a maximum of 250 steps, and with termination after 50 steps without improvement. A maximal translation of 1.0 Å and a maximal rotation of 0.034 radians (~2°) were applied to the structure in every step. For the 6.8 Å density map section 7 correct fits (bold) were identified, of which 3 (italic) were symmetrically related. The 4 symmetrically independent fits are shown in Figure 6 and Figure 7. The $\mathrm{RMSD}_{C\alpha}$ of initial to refined fit is shown to indicate the amount of movement of the atomic model during the refinement step.

Map resolution [Å]	Rank by hash score	Hash score	Initial *CCC*	Optimized *CCC*	$\mathrm{RMSD}_{C\alpha}$ of optimized to initial fit [Å]
6.8	**21**	**110**	**0.20**	**0.48**	**11.07**
6.8	***6***	***116***	***0.20***	***0.48***	***7.71***
6.8	***19***	***110***	***0.12***	***0.48***	***12.55***
6.8	**34**	**107**	**0.15**	**0.48**	**11.84**
6.8	**3**	**117**	**0.25**	**0.48**	**7.07**
6.8	**39**	**106**	**0.13**	**0.47**	**15.15**
6.8	***12***	***113***	***0.14***	***0.47***	***11.27***
6.8	11	113	0.10	0.13	2.48
6.8	29	108	0.10	0.11	5.33
6.8	7	115	0.09	0.11	3.78
9.0	**1**	**162**	**0.68**	**0.73**	**4.38**
9.0	***4***	***149***	***0.48***	***0.73***	***13.19***
9.0	**9**	**144**	**0.55**	**0.70**	**7.42**
9.0	***15***	***142***	***0.54***	***0.70***	***8.14***
9.0	***12***	***142***	***0.50***	***0.70***	***10.53***
9.0	**2**	**151**	**0.53**	**0.69**	**10.17**
9.0	***7***	***146***	***0.61***	***0.69***	***6.23***
9.0	***37***	***130***	***0.51***	***0.69***	***10.66***
9.0	**34**	**132**	**0.58**	**0.68**	**6.07**
9.0	***14***	***142***	***0.42***	***0.68***	***13.61***

4 copies of 1OELG are docked into the chaperonin GroEL density map at 5.4 Å resolution

A single chain (id: G) of the crystal structure of the chaperonin GroEL (pdb: 1OEL) [62] was docked into the complete 5.4 Å resolution density map of GroEL (EMDB: 1457) [56], [63]. GroEL is a dual heptameric particle with a main 7-fold axis and a perpendicular 2-fold axis (dihedral 7-fold symmetry). The BCL::EM-Fit algorithm identified six correct fits (Table 7) which could be confirmed visually. Four of them are in different positions (Figure 8). Initial fits had *CCCs* between 0.39 and 0.62; refined fits had *CCCs* between 0.62 and 0.75. The entire procedure took 51 minutes on a single core of an Intel(R) Xeon(R) CPU W3570 @ 3.20GHz.

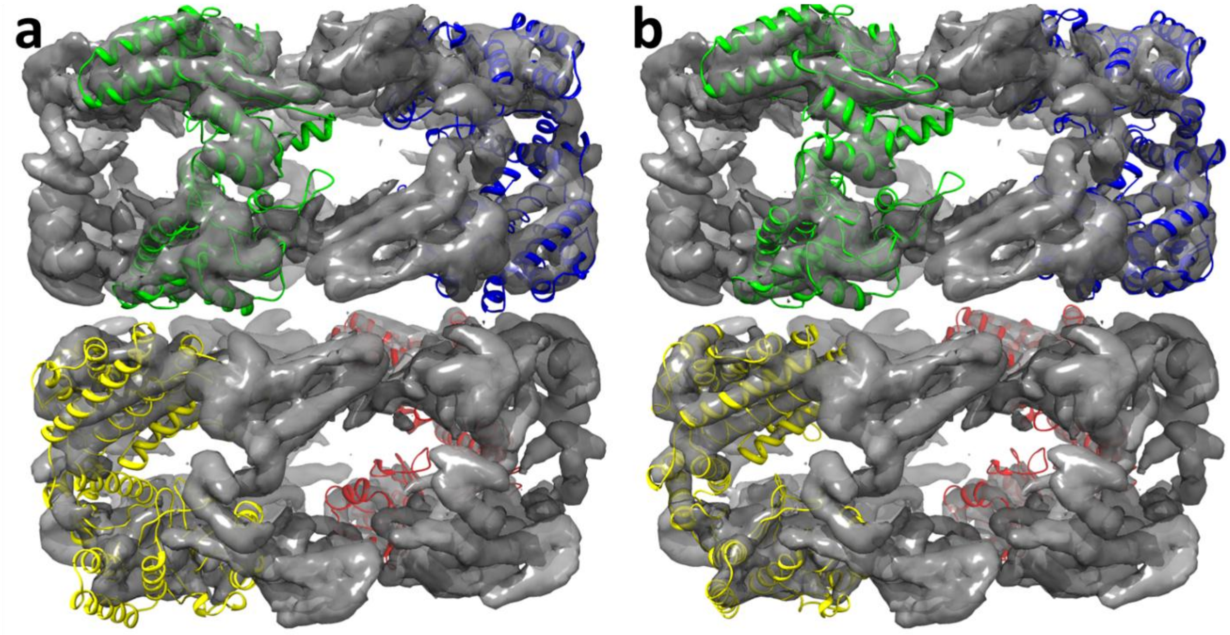


Figure 8 BCL::EM-Fit docking of 1OELG into 5.4 Å resolution density map of GroEL

The best 4 unique fits out of 50 fits (by *CCC*) for the 1OEL chain G of the chaperonin GroEL are shown in their initial (a) and final (b) positions, after 250 steps of MCM refinement. The coordinates were docked within the cryoEM density map (EMDB: 1457) at 5.4 Å resolution (gray), which is shown at an isosurface level chosen to emphasize secondary structure elements.

Table 7 Docking of 1OELG in 5.4 Å resolution density map

Best ten placements by *CCC* after initial fit of 1OEL chain G of the chaperonin GroEL into a 5.4 Å resolution cryoEM density map of GroEL (EMDB:1457). The best 50 initial fits by *CCC* were optimized by MCM refinement with a maximum of 250 steps, and with termination after 50 steps without improvement. A maximal translation of 1.0 Å and a maximal rotation of 0.034 radians (~2°) were applied to the structure in every step. Six correct fits (**bold**) could be identified, of which 2 (***italic***) are duplicates. The four independent fits are shown in Figure 8. The $\text{RMSD}_{C\alpha}$ of initial to refined fit is shown to indicate the amount of movement of the atomic model during the refinement step.

Rank by hash score	Hash score	Initial *CCC*	Optimized *CCC*	$\text{RMSD}_{C\alpha}$ of optimized to initial fit [Å]
18	**64**	**0.51**	**0.75**	**4.86**
4	**68**	**0.51**	**0.75**	**5.74**
3	**69**	**0.39**	**0.74**	**8.95**
2	**69**	**0.49**	**0.74**	**5.46**
8	***67***	***0.59***	***0.74***	***3.49***
22	***64***	***0.62***	***0.62***	***0.29***
10	66	0.31	0.38	6.96
7	67	0.30	0.36	5.83
13	66	0.22	0.35	8.53
24	64	0.27	0.35	3.16

Correct handedness of a density maps can be verified by the CCC of the initial fit

Imaging a macromolecular assembly by transmission electron microscopy results in the loss of the absolute hand of the structure because the three-dimensional information is projected into a two-dimensional plane. Several methods for determining the absolute hand of a cryoEM single particle reconstruction have been developed, which involve collecting tilted images [64], [65]. Often however the absolute hand of a cryoEM structure is not experimentally determined, and thus both possible hands of the density should be tested when docking atomic resolution structures. To test the BCL::EM-Fit algorithm’s ability to distinguish correct from incorrect handedness, two versions of the experimental density map segment around the adenovirus penton base were created (correct and flipped). The refined fits for the correct map have CCCs of as high as 0.54. In contrast, the refined fits for the flipped map have a CCC only as high as 0.27 (Table

8). This indicates that given a density map with a sufficiently high resolution (6.8 Å resolution in this example), the BCL::EM-Fit algorithm can differentiate between the two possible hands of the density map and select the map with the correct hand.

Table 8. Comparison of the initial fitting and refinement step by BCL::EM-Fit for penton base into the correct and the symmetry-inverted density maps at 6.8 Å resolution

Correct				Flipped[a]			
Rank by hash score	Hash score	Initial *CCC*	Optimized *CCC*	Rank by hash score	Hash score	Initial *CCC*	Optimized *CCC*
10	179	0.19	0.54	4	177	0.16	0.27
2	191	0.27	0.53	6	175	0.18	0.27
3	186	0.30	0.53	2	179	0.17	0.18
6	181	0.19	0.53	8	173	0.06	0.15
5	181	0.19	0.53	7	175	0.10	0.13
1	192	0.31	0.53	3	179	0.11	0.12
4	182	0.29	0.53	1	179	0.10	0.11
8	181	0.14	0.17	9	173	0.08	0.10
9	180	0.10	0.16	0	179	0.07	0.08
7	181	0.06	0.07	5	175	0.05	0.08
Mean	183	0.20	0.41		176	0.11	0.15
SD	5	0.09	0.19		3	0.05	0.07

[a]The flipped density map was created to have the opposite handedness compared to the correct density map.

Discussion

Docking works best when secondary structural elements are resolved within the density map

A new algorithm, BCL::EM-Fit, is presented for rapid and accurate docking of atomic resolution structures within moderate resolution (5-12 Å) density maps. The protocol consists of feature extraction from the density map and encoding of this information into a geometric hash map, followed by searching of the hash map with features extracted from the coordinate file of an atomic resolution structure or model. The resulting initial fits are then refined in an MCM refinement step. Docking experiments with benchmark proteins demonstrate reliable fitting of atomic structures if the density map has a resolution of ~ 10 Å or better. The docking experiments also indicate that the *CCC* between simulated and experimental density maps is a satisfactory way to identify optimal positions, since the highest CCC is observed for positions that have an RMSD < 5Å to the correct placement.

Benchmark tests were performed with α-helical proteins, mixed α/β-proteins, and predominantly β-strand proteins. The algorithm works reliably for α-helical proteins with nearly no incorrect fits at resolutions up to 12 Å. The algorithm also works well for α/β and β-strand proteins for resolutions up to ~11 or 10 Å, respectively. The better performance of BCL::EM-Fit with mostly α-helical proteins is attributed to the fact that α-helices can be resolved at more moderate resolution than β-strands (Zhou, 2008). For resolutions in the range of 12 to 19 Å the secondary structure elements that help to accurately position atomic models are not well enough resolved for the BCL::EM-Fit algorithm to find the correct fit in all cases.

BCL::EM-Fit correctly identifies and places homologous structures and comparative models

A cross fitting experiment with five simulated density maps and homologous structures or comparative models was performed (Table 4). The ambiguous docking results with one simulated density map (that of 2E3S, a mostly β-strand benchmark protein) might have been alleviated if higher resolution density maps were used. The results indicate that

BCL::EM-Fit works reasonably well with both homologous structures and comparative models, however better docking results were obtained with comparative models.

BCL::EM-Fit is applicable to fitting of large adenovirus capsid proteins

For human adenovirus penton base and hexon capsid protein were fitted within 6.8 and 9 Å resolution sections of experimental cryoEM density maps of the entire virus. The generated fits of the atomic resolution protein structures cover all symmetrically unrelated placements which can be used to rebuild the 3D structure of the entire virus capsid. BCL::EM-Fit was further capable of identifying the correct handedness of the reconstructed cryoEM density map by superior hash score and *CCCs* at the initial and refinement stage of fitting.

BCL::EM-Fit can fit subunits within a larger assembly

In addition to the tests with the multimeric adenovirus capsid proteins, BCL::EM-Fit was also used to successfully fit a single chain of 1OEL into the GroEL density map at 5.4 Å resolution. Although only 4 of the 14 copies were found, the knowledge of the 7-fold dihedral symmetry of GroEL would enable the construction of the complete assembly from only one correctly docked subunit. Alternatively, one could refine more of the initial fits and expect to find more independent positions at the cost of a longer fitting time.

BCL::EM-Fit and flexible docking

All benchmarks and examples shown here are rigid body fitting experiments that provide an initial fit. This experimental design allows testing the geometric hashing approach which is tailored for the rigid body fitting problem. One possible way to explore protein flexibility on the domain level is to separate the coordinates of the protein of interest into independent domains and fit them into the density map separately. Internal flexibility could be simulated with molecular dynamics programs and a selected set of representative conformations could be saved and subsequently fit into a density map. Additional tools have been developed that perform flexible docking once an initial fit is identified, e.g. using BCL::EM-Fit. These include QPLASTY in the SITUS package [37], ROSETTA [28], molecular dynamics flexible docking (MDFF) [50] and DireX [49].

Advantages and disadvantages of Geometric Hashing compared to Fourier/Real Space fitting

The geometric hashing approach is presented as an alternative method for fitting atomic resolution structures into multiple positions within large density maps. The BCL::EM-Fit results demonstrate good performance for fitting proteins into density maps of a resolution up to 12 Å. All orientations and positions of interest for the hexon and penton base proteins in adenovirus could be determined within sections of the virus density map at 6.8 and 9 Å resolution. A time comparison to the exhaustive Fourier/Real Space search method as implemented in COLORES revealed a 3-fold advantage for BCL::EM-Fit using a single CPU (Table 9). COLORES may still be advantageous in several scenarios. It samples all regions of the density map evenly and therefore it can identify matches that might be missed by the geometric hashing approach. This is especially true for lower resolution density maps (> 12Å) that often lack distinctive features. A second advantage relates to the fact that closely packed protein domains in obligate oligomers might appear as one continuous domain to the feature matching algorithm of EM-Fit. In cases like this a Fourier/Real Space search has an increased chance of identifying all monomeric copies of the protein. These disadvantages of BCL::EM-Fit will be addressed in future versions of the program. Nevertheless, given the growing importance of docking atomic models into cryoEM density maps it should prove useful to have multiple algorithms to accomplish this task.

Table 9 Time comparison between COLORES and BCL::EM-Fit

Target	Method	Hash map setup[b] [min]	Initial fit[c] [min]	Optimization[d] [min]	Number of Fits found[e]	Total time [min]
penton base	Colores[a]	0	404	213	4	616
penton base	GH/MCM	6	31	41	3	72
hexon	Colores[a]	0	729	164	3	893
hexon	GH/MCM	39	139	117	7	256

[a] The COLORES jobs were performed with a 20° angular search step size during the initial fit and with 10 positions optimized during refinement.

[b] Extracting features from the density map, and storing all possible bases with quantized features in a hash map

[c] Initial fits generated by each method.

[d] COLORES uses a gradient based method, GH/MCM uses Monte Carlo optimization.

[e] Number of independent fits that differ either in their rotation around a symmetry axis (penton base and hexon), or their translation within the density segment (hexon).

Methods

Geometric hashing re-casted for searching density maps with protein structures

The following paragraph gives a general overview of the steps required before a more detailed description of the present implementation is given. The basic idea of geometric hashing was developed for image recognition in robotic applications. Critical points of a complex image (features) are extracted into a feature cloud. A large number of possible rotations and translations of this feature cloud are encoded *a priori* in a hash map [52] which later allows a rapid search for objects within this image. For BCL::EM-Fit the 3D image will be the cryoEM density map. The objects to be recognized will be protein structures which will also be represented as feature clouds. Each combination of a rotation (three degrees of freedom) and translation (three degrees of freedom) of the feature cloud is a transformation with six degrees of freedom.

The general scheme for generating the geometric hash is to define many possible transformations for the density map feature cloud and store these in a memory-efficient, rapidly searchable hash map. In this process the features are “quantized”, i.e. not the actual position of a feature but only the specific space bin that contains the feature is stored. This procedure not only saves memory and accelerates the search, it also limits the search to a finite (but large) set out of all possible transformations. Further it compensates for experimental noise in the density map and protein structure. In the recognition step this hash map is searched with a feature cloud of the protein to be docked. It is expected that one of the original transformations puts the feature cloud of the density map in good overlap with the feature cloud of the protein. This can be recognized by the number of shared features, i.e. features that end up in the same space bin.

This procedure speeds up the search as not the complete image but only the features deemed important are considered. Further, not every possible transformation is considered but only a finite subset. In contrast to robotics the problem of scaling the image is absent for feature-recognition in a distance invariant cryoEM density because the units of length in the density map and atomic models are the same. Further, 3D images have an increased complexity over 2D pictures that a robot usually sees using a single camera, which changes the protocol slightly compared to plain 2D picture recognition.

Extraction of feature cloud from density map intensities (Figure 2a)

The user inputs a density map that will be completely encoded as a point cloud for rapid fitting. If the user wants to fit into a specific segment of the density map, it is necessary to extract that from the original map in a pre-processing step. In order to generate a representation of the features in the density map two pieces of information are used (Figure 2a): the absolute intensity of a Voxel and the intensity difference to its neighboring Voxel, a gradient. The higher the intensity the more likely it is that a structurally compact region such as a secondary structure element can be found in the respective position of the density maps. The higher the intensity gradient the more likely the edge of a secondary structure element can be found here. Often there is an intensity drop at the edge of secondary structure elements due to less rigid amino acid side chain atoms. The edge regions are usually close to backbone atoms of secondary structure elements and encode most of the information within the density map. In order to define the total number of features extracted from a density map Equation (1) was derived empirically:

$$N_{points} = N_{Voxel\ Atoms} \times \frac{V_{Voxel}}{Max\left(\frac{\pi}{6}d_{fd}^{3}, V_{Voxel}, \rho_{Atoms\ Protein}^{-1}\right)} \tag{1}$$

$N_{Voxel\ Atoms}$ - Number of Voxels the atoms would occupy when mapped to grid of the density map

V_{Voxel} - Volume of Voxel

$\frac{\pi}{6}d_{fd}^{3}$ - Volume that one point occupies according to feature distance

V_{vox} - Volume that one point occupies according to a Voxel's volume

$\rho_{Atoms\ Protein}^{-1}$ - Volume that one point occupies according to the density of selected atoms for fitting in the protein

The number of features that represent the density map should be proportional to the number of Voxels that are occupied when the selected atoms in the protein structure that is to be fitted is mapped to the grid defined by the Voxel size of the density map ($N_{Voxel\ Atoms}$). This number is reduced by the maximal volume that one feature can occupy. The maximum is given by one feature occupying one Voxel (V_{Voxel}) which reduces (1) to $N_{points} = N_{Voxel\ Atoms}$. If the density of atoms that are to be fitted is low, the expected Volume one feature is occupying is high which reduces (1) to $N_{points} = N_{Voxel\ Atoms} \times \frac{V_{Voxel}}{\rho_{Atoms\ Protein}^{-1}}$. If the feature distance is chosen high, the volume one feature occupies is high which reduces Equation (1) to $N_{points} = N_{Voxel\ Atoms} \times \frac{V_{Voxel}}{\frac{\pi}{6}d_{fd}^{3}}$. A good estimate for the number of features reduces the size of the hash map since less triangular bases are constructed and fewer features have to be transformed, quantized, and stored (read below). In addition a sufficient number of features guarantee enough triangular bases, to achieve a high precision for the fits. Custom optimization of Equation (1) or its parameters might be required for optimal results. However, the algorithm proved robust in the presented work with respect to deviations in N_{points} of up to 25%. Hence, Equation (1) should be applicable for most scenarios. A default choice for the feature distance is 0.15 * r_{gyr} (radius of gyration of protein to be fitted), which has proven robust for the presented experiments, but can be modified. A smaller feature distance will lead to more overall features and longer fitting times. The actual scaling for the time cannot be determined since the feature distance also influences the number of triangular bases. To a first approximation, the overall time should scale quadratically with the reduction of the feature distance. Setting the feature distance to a value larger than the default value may lead to the possibility that an insufficient number of features are encoded.

The actual features are extracted by iterating over all Voxels. For each Voxel the intensity is added to the gradient intensity of the neighboring Voxels. The gradient is the sum of all absolute differences to the neighboring Voxels, i.e. 6 Voxels adjacent on the faces, 12 on the edges and 8 on the vertices. The absolute differences are normalized by the distance between the Voxels, e.g. Voxels adjacent on the yz-faces are normalized by Voxel length in x-direction (vl_x) or Voxels on the vertices by $\sqrt{vl_x^2 + vl_y^2 + vl_z^2}$. The Voxel is converted into a feature by adding the maps indices to the Voxel's indices and by multiplying with the Voxel side and adding the maps origin afterwards. Half of the Voxel side lengths are also added to center the feature in the Voxel. The feature is inserted in a list with its intensity and gradient sum and is sorted by the sum. Finally, starting with the highest intensity-gradient-sum, the list is searched for all features that are within the feature radius of that feature, which have to be removed. Then the list is searched for all overlapping features with the second highest by the intensity-gradient-sum. This happens until no overlapping features remain. The list is then cut down to the requested number of features removing the lowest intensity-gradient-sum features.

Selection of triangular bases for coordinate transformations (Figure 2b)

Triplets of the features f_1, f_2 and f_3 within the density map are treated as an origin of a coordinate system – a so called triangular base. Transforming all remaining features within a specified feature radius of the triangular base, this coordinate system encodes the relative position of the features with respect to this base. The internal coordinate system represented by the triangular base is invariant to the absolute position of the structure in space but encodes only relative positions of features.

It was critical to *not* consider all possible triplets of features as base. Rules were imposed that ensured that the distances $d_1 = \|f_2 - f_3\|$, $d_2 = \|f_1 - f_3\|$, and $d_3 = \|f_1 - f_2\|$ between the features f_1, f_2 and f_3 are chosen to be between 0 and the radius of gyration of the structure to be fitted. The rationale for this approach is that within this range the relative arrangement of secondary structure elements is defined. This is ultimately the structural entity to be recognized in the search procedure. Further, it is advantageous to ensure that d_1, d_2 and d_3 are significantly different from each other and can be sorted (read below). For that purpose three thresholds are defined: r_{gyr}, the radius of gyration of the protein to be fitted, a high and a low threshold t_h and t_l. These are determined by binning all pairwise distance into 100 equal sized distance bins in the range [0, r_{gyr}]. The resulting distance histogram is used to find the two bins, at which 1/3 of all distance (t_l) and 2/3 of all distances (t_h) were observed, which typically turns out to be close to 0.5 and 0.75 times the radius of gyration of the protein to be fitted. The distances d_1, d_2 and d_3 have to fulfill the conditions:

$$r_{gyr} > d_1 > t_h > d_2 > t_l > d_3 > 0 \tag{2}$$

The arithmetic center of the triangle f_1, f_2 and f_3 is used as the origin of the coordinate system, letting f_1 be on the positive x-axis, f_2 in the positive *xy*-plane. This generates an ordered triplet of features and a unique transformation T_D for those three features. Without an ordering $d_1 > d_2 > d_3$, it would be necessary to store all six possible transformations for a triangular base (starting from f1, f2 or f3, clockwise or counter clockwise) increasing the computational time by a factor of 6 respectively. Additionally, the geometric hashing fit step would also need to consider 6 different transformations for the chosen triangular base totaling to a factor of 36.

The maximal distance of features from the coordinate base is limited by a feature radius (Figure 2b)

Only coordinates that are within the feature radius (outer most circle of the spherical coordinate system, Figure 2b) are transformed and quantized. The rationale for the feature radius is that only features within the size of a typical protein domain need to be encoded. Features outside this radius arise from noise in the density map or neighboring domains and fitting results would not be improved even when considering these features. This radius restriction is particularly important if a large density map of multiple proteins is searched for individual proteins or domains. In this case the feature radius helps to reduce the memory required for storing the hash map and to reduce the computational time.

The feature radius can be seen as a maximum size of objects that can be reliably detected within the encoded density map. Hence, the feature radius should be chosen based on the size of structures that will be fitted and should have a value between the radius of gyration and the longest extent of that object. By default it is chosen to be 1.25*r_{gyr}. All features f_i considered for transformation have to be within the distance r of the middle point $M = \frac{1}{3}(f_1 + f_2 + f_3)$ of the three features f_1, f_2 and f_3 used as the origin.

$$r > \left\| f_i - \frac{1}{3}(f_1 + f_2 + f_3) \right\| \tag{3}$$

Quantization of features accounts for finite number of transformations, low resolution of the density map, and experimental noise (Figure 2b-c)

To generate the keys from the transformed features f_i a quantization procedure is applied. Quantization assigns the feature to some bin in space based on its position. The advantage of such binning is that only a finite number of bins exist which will be the keys of the hash map. The precision of the quantization adjusts also the tolerance in the feature matching step (read below), i.e. features in the density map that would map to atoms in the protein can deviate significantly if they are distant from the triangular base but should still count as a match. The density maps extracted features represent edges and high intensity density features. The feature cloud of the protein represents certain atoms (read below). However, it is not expected that these points superimpose precisely as features mark general regions not precise points. Both density map and protein structure are experimental data affiliated with errors and uncertainties. Hence, a certain tolerance between features of the density map and features of the atomic structure should be allowed for matches.

The precision of the quantization needs to be tuned to the resolution of the density map. A lower precision will tolerate a larger distance between an atomic feature and a density feature in the fit. The number of distinct keys will be small and the fitting will be faster, but accuracy might suffer. A higher precision on the other hand will give closer and more reliable fits. It will produce more distinct keys, require more time for the fitting, and should be used with higher resolution density maps.

In the present implementation a Spherical coordinate system was used to define the bins rather than a Cartesian coordinate system. The radius of the bins was chosen to increase logarithmically. The choice of the coordinate system has certain advantages and disadvantages: The use of a spherical coordinate system requires the conversion of the point cloud coordinates from Cartesian to Spherical coordinates. In contrast to the Cartesian coordinate system in the Spherical coordinate system the bin sizes increase with distance from the origin, i.e. a spherical coordinate system has a lower resolution for points that are farther away from the origin. This is beneficial as small changes in the transformation will disproportionately affect the position of features distant from the origin. In a Spherical quantization these points may remain in the same bin and can be recognized as overlapping features (read below) while in a Cartesian quantization they would wander into the next space bin. Spherical quantization gave slightly better results than Cartesian quantization in benchmark experiments (data not shown). The following equations were used to convert Cartesian coordinates $\overrightarrow{pos}_{Cartesian} = (x, y, z)$ into Spherical coordinates $\overrightarrow{pos}_{Spherical} = (r, \vartheta, \varphi)$:

$$\overrightarrow{pos}_{Spherical} = \begin{pmatrix} r \\ \vartheta \\ \varphi \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2 + z^2} \\ arccos\left(\frac{z}{r}\right) \\ arctan2(y, x) \end{pmatrix} \tag{4}$$

For quantization the following equations were applied to the Spherical coordinates:

$$\begin{pmatrix} r_q \\ \vartheta_q \\ \varphi_q \end{pmatrix} = \left\lfloor \begin{matrix} \dfrac{\log(r)}{\log\left(\dfrac{2\pi}{res} + 1\right)} \\ \vartheta * \dfrac{res}{\pi} \\ \dfrac{\varphi}{\pi} * \left\lfloor \sin\left(\dfrac{\pi * \vartheta_q}{res}\right) * res + 1 \right\rfloor \end{matrix} \right\rfloor \tag{5}$$

where res is the resolution of the key and influences the quantization. The smaller res is, the more points will fall in the same bin and the more the initial fit deviates from the correct fit. Hence the hash key resolution behaves in the opposite manner to the density map resolution. A typical value is twelve, which creates twelve angular bins for φ_q on the equator of the spherical coordinate system each spanning an angle of 24°. Since $\frac{\varphi}{\pi}$ is in the range [0,1] and for the equator $\vartheta_q = \left\lfloor \frac{\pi}{2} * \frac{15}{\pi} \right\rfloor = 7$ the term $\frac{\varphi}{\pi} * \left\lfloor \sin\left(\frac{\pi * \vartheta_q}{res}\right) * res + 1 \right\rfloor = \frac{\varphi}{\pi} * \left\lfloor \sin\left(\frac{\pi * 7}{15}\right) * 15 + 1 \right\rfloor = \frac{\varphi}{\pi} * \lfloor 0.99 * 15 + 1 \rfloor = \frac{\varphi}{\pi} * 15$ creating a range after applying [0,15) of integer values, where 15 is at the open end of the interval because of the quantization of the floor function. The function $\lfloor x \rfloor$ = floor(x) returns the largest integer not greater than x (e.g. $\lfloor 1.1 \rfloor = 1; \lfloor 7.7 \rfloor = 7$). The key was assembled as one number using:

$$key = r_q * 10{,}000 + \vartheta_q * 100 + \varphi_q \tag{6}$$

The factors 10,000 and 100 have to be increased, if the hash resolution increases to guarantee that there is no overlap between the individual quantized terms.

Hash map architecture (Figure 2c)

For a specific transformation T_D every feature f_i within the feature radius f_r is converted into a key and stored in the hash map together with its respective transformation T_D. The resulting keys can be rapidly looked up in the hash map and all transformations T_D affiliated with a single key will be returned. It is very likely that there are multiple bases for one key, and it is also likely that certain keys will never be observed. Preprocessing of the density map and storing the hash map is the most memory and time consuming part of the algorithm. The actual implementation uses a SQL databank for larger hash maps, but can be stored in the RAM of a computer for smaller density maps accelerating the search.

Atoms within secondary structure elements are used as features to represent the protein (Figure 2d)

A feature cloud for the protein to be fitted needs to be created. Since the atomic structure of the target protein is given it is possible to use the coordinates of atoms as features, preferably atoms that are close to regions which have high intensities in density maps. For the present purpose these are the backbone atoms within secondary structure elements. The relative rigidity of these regions coupled with the density in conjugated peptide bonds gives rise to high-intensity regions, i.e. the frequently discussed “density rods” seen for α-helices [35], [66]. It is sufficient to include a fraction of all backbone atoms, i.e. C_α atoms, to reduce the number of features to be matched minimizing the time for fitting (Figure 2d). Usage of any other backbone atom instead of C_α did not affect the accuracy of the protocol significantly (data not shown). It is recommended that the Cα atoms of all secondary structure elements be used as the feature cloud of the protein. For this purpose the program uses the secondary structure definition as given in HELIX and SHEET section of the PDB entry to automatically select the respective atoms. Atom names are taken from the ATOM lines in the PDB file. The user can alter which secondary structure regions to consider by changing the minimal length of the three secondary structure types (helix, sheet, loop) from the default values (0, 0, 999). Additionally, the user can pass a list of backbone atoms to be used although it is recommended to only use the Cα atoms as the use of additional atoms will increase the runtime and may not improve the results.

Initial fits are determined that superimpose the maximum number of features (Figure 2e-g)

Once the feature set of the target is extracted, a possible triangular base is identified. In this procedure the same criteria are applied with respect to f_1, f_2 and f_3 that were used to encode the density map (Figure 2e). Applying the resulting transformation T_P to the remaining features within the feature radius r and quantizing them yields a set of keys. This set of keys is now looked up in the hash map and transformations T_D are identified that are common among a maximum number of keys (Figure 2f). Such transformations superimpose target protein and density with a maximum number of agreeing features and create a ranked list of initial fits. The transformation T_{fit} needed to fit the protein into the density is defined as $T_{fit} = T_P \times T_D^{-1}$ (Figure 2g).

Since it cannot be expected that any three features of the target protein are necessarily represented in the feature cloud of the density map, the fitting is repeated multiple times (Figure 2e) and all transformations are ranked by the number of agreeing features (identical keys, Figure 2f). The number of agreeing features is a quality measure for the initial fit. Since a large number of triangular bases within the target can be used, the following method is used to assure that the target is sampled equally, i.e. different bases with centers at sufficiently different locations within the target are picked. All bases are binned with their base centers on a Cartesian grid, with a grid width chosen, so that there are more grid elements occupied than fitting trials requested. Now, a grid element is picked randomly, and marked to not be picked again. A random triangular base within that grid element is chosen for the geometric hash fit procedure.

The accuracy of the initial fit depends on the number of features extracted from the density map and the number of features extracted from the protein model. More features increase the resolution and possibly the accuracy of the fit as more features in space are represented and more triangular bases can be identified. Since each base represents a set of translations and rotations, the space of transformations is sampled more densely. A higher agreement resulting from more superimposed features in the initial fit also results in a higher *CCC* with the density map. However, a large number of features results in longer computation times. Hence, the minimal number of features required to accurately represent the experimental information within the cryoEM density map should be used.

The estimate for the number of features in the density map given in Equation (1) represents a compromise between accuracy and computation time.

Filtering fits by translational and rotational distance

For the fitting of the penton base, hexon and GroEL, independent fits were defined by specified minimal rotational and translational differences before the geometric hash step. This is necessary, because the geometric hashing algorithm has an intrinsic property that leads to nearly identical fits being found in multiple searches with different triangular bases. In order to find a comprehensive list of independent and highly scoring fits, it is necessary to remove non-independent fits so that a few solutions do not dominate the output list.

The initial fits have to be optimized (Figure 3)

For the purpose of optimizing initial fits, a simulated density map is computed from the atomic structure of the target with a resolution comparable to that of the experimental cryoEM density map. Starting from the position of the initial fit, small random translations and rotations are applied to the protein in order to maximize the *CCC* (Equation (7)) in a Monte-Carlo/Metropolis (MCM) simulated annealing protocol (Figure 3).

$$CCC = \frac{k\sum_{y<k}^{y=0}\rho_S(y)\rho_E(y) - \sum_{y<k}^{y=0}\rho_S(y)\sum_{y<k}^{y=0}\rho_E(y)}{\sqrt[2]{\mathrm{k}\sum_{y<k}^{y=0}\rho_E(y)^2 - \left(\sum_{y<k}^{y=0}\rho_E(y)\right)^2} * \sqrt[2]{k\sum_{y<k}^{y=0}\rho_S(y)^2 - \left(\sum_{y<k}^{y=0}\rho_S(y)\right)^2}} \qquad (7)$$

ρ_s and ρ_E are simulated and experimental overlapping densities. k is the number of overlapping Voxels for which $\rho_s > 0$. This condition represents an "envelope" around the experimental density which will ignore noise in the region where no density was simulated from the fitted atomic structure. Y is the iteration index over all Voxel pairs that fulfill the $\rho_s < 0$ condition. The value of *CCC* will be 1 for best correlation, 0 for no correlation and -1 for anti-correlation.

Compared to gradient based methods Monte Carlo/Metropolis optimization is capable of sampling multiple local minima on a rugged objective function but is nevertheless accurate and fast. The scoring function is rugged due to experimental noise in the density map and due to the fact that Voxel spacing quantizes the function. The input parameters for the protocol include maximum amplitude for rotations and translations, a maximum number of total iterations, and a maximum number of subsequent steps with no improvement in *CCC*. Typical translational step sizes are 0-1.0 Å; rotations are limited to 0.035 radians (~2°). An average optimization explores between 100 and 200 steps, stops at a maximum of 250 steps, but terminates after 50 steps without an improvement in the *CCC*. The temperature parameter for the Metropolis criterion is adjusted automatically to match a certain ratio between accepted and rejected steps. This "simulated annealing" protocol starts with an estimated 50% ratio of accepted vs. rejected steps and ends with an approximate 20% ratio over the maximum of 250 steps, i.e. the final ratio of accepted steps is typically close to 0%.

Addition of noise to the synthesized density maps

Density maps were synthesized from coordinates following an implementation of pdb2vol in the SITUS package, using trilinear interpolation and Gaussian flattening kernel. This method produces density maps with zero intensity outside an envelope surrounding the protein. Different experimental deviations between the electron density map and the atomic structure can occur. First, there may be deviations in the structure or dynamics of the protein between the cryoEM conditions and the conditions used to determine the atomic-detail model. For example packing artifacts in crystals used for X-ray crystallography can result in different protein conformations than observed by cryoEM where the samples are preserved in near native conditions. Both can differ from structure and dynamics of an isolated dissolved protein observed in an NMR experiment. Further, differences in the actual proteins can occur such as length of the constructs or mutations. These deviations are not accounted for in the present algorithm but could in part be addressed through a flexible docking protocol.

However, a careful analysis was performed to test the robustness of the algorithm in the presence of noise. The noise added was Gaussian noise to mimic some of the error that is inherent in experimental density maps. While iterating over all Voxels a normally distributed number was added to each Voxel's intensity. After iterating over all Voxels, the *CCC* between the noise-free and noise-added map was calculated. This process of adding noise was repeated, until the desired *CCC* to the noise-free density map was reached.

Specific parameters used for benchmark of 50 diverse proteins with simulated density maps

The proteins selected for the test have between 150 and 300 residues. Fifteen density maps in the resolution range of 5 Å to 19 Å in 1 Å steps were simulated from each of the crystal structures with Gaussian flattening [37]. The Voxel size was chosen to be 1/3 of the resolution. For each protein/resolution combination four additional density maps were calculated with different levels of Gaussian noise added. The noise levels were adjusted so that the *CCC* values of the noise-added maps to the noise-free maps would be approximately 0.9, 0.8, 0.7 and 0.6. The *CCC* values were calculated according to Equation (7). Figure 4d shows one of the α/β benchmark proteins (1prz) with its noise-free simulated density map and its noise-added maps at a resolution of 10 Å. Visual inspection reveals that maps with noise at *CCC* value of 0.8 look comparable to the experimental map of adenovirus. The simulated maps and the corresponding atomic coordinates served as input for the BCL::EM-Fit geometric hashing and MCM optimization routines.

For the geometric hashing step the density maps were converted into feature clouds with between 22 to 232 points. These point number totals are intended to represent the structural features in a particular density map, which depends on the Voxel size, the size of the protein, and the minimum distance between two resolvable features (Equation (7)). Ten top scoring placements from the initial geometric hashing step were selected for each atomic model fit into each of its simulated density maps (the noise-free map and the four noise-added maps) at each of the 15 resolution test points. These initial hits were subjected to MCM refinement in real space.

Specific parameters used for penton base, hexon and GroEL

For the penton base fit, 709 and 631 features were extracted from the density segments at 6.8 and 9.0 Å resolution, respectively. The hexon capsid protein density segments were represented by 2890 and 3699 features for the 6.8 and 9.0 Å density maps, respectively.

2884 features were used represent the entire 5.4 Å resolution density map of GroEL. The weight for intensity vs. gradient was the standard 1:1 ratio for all experiments (a). The t_l and t_h values as described in Equation (2), the feature distances and the feature radii were derived from the radius of gyration. For all fitting procedures, a spherical coordinate system was used. The precision for the hash key quantization was set to 12 (Figure 2b).

For the fitting of the proteins in the benchmark set, C_α atoms in helices or strands were extracted as features depending on whether it was more predominantly an α-helical, α/β or β-strand protein. For the penton base, C_α atoms in α-helices and β-strands were selected for fitting, for the hexon C_α atoms in β-strands, for GroEL C_α atoms in α-helices were selected for fitting (Figure 2d). In all procedures 500 randomly chosen bases (Figure 2e) were selected to generate a list of transformations T_D ordered by the number of agreeing features representing the best possible initial fits (Figure 2f,g). For all MCM optimizations the specific parameters were derived as described in the Methods section “The initial fits have to be optimized".

In an effort to remove similar transformations $T_{fit} = T_P \times T_D^{-1}$ the list of initial fits for the penton base was filtered by removing solutions if their centers were within 5 Å and had a relative effective rotation angle smaller than 1 radian (~60°) using a previously described protocol [67]. The list of initial fits for the hexon was filtered by removing solutions that were closer than 60 Å and had a relative effective rotation angle of less than 2 radians (~120°). Two fits for the GroEL experiment were considered identical within a translational difference of 5 Å and rotational difference of 3 radians (~170°). This filtering was necessary to find symmetrically related copies (since the hexon and penton base proteins are multimers) and to find translationally independent copies (the hexon map density segment had density for at least 4 full hexon proteins, the GroEL density map contained density for all 14 subunits).

Fold recognition and construction of comparative models using bioinfo.pl and Modeller

To identify template folds and construct comparative models for the benchmark proteins their primary sequences were submitted to the bioinfo.pl metaserver. The output with the best aligned sequence, and with sequence similarity < 99% to the original sequence, was chosen as a homologous structure. This helps to ensure that the template protein and homologous structure will have some differences. It is appreciated that in real-word applications the template and target structures may be considerably more distinct. However, a more detailed analysis of usage of comparative models for fitting is beyond the scope of the present work. The homologous proteins were downloaded from the PDB [5] and used for cross-fitting experiments. Comparative models were acquired by submitting the bioinfo.pl alignment to the MODELLER server using the "model" link provided on the bioinfo.pl website. This approach was chosen to keep the protocol as straight-forward and unbiased as possible. A more elaborate construction yielding possibly more accurate comparative models for fitting into cryoEM density maps remains to be pursued in future studies.

Conclusion

The intensities in a cryoEM density map represent structural features of rigid and dense parts of the structure, in particular secondary structure elements at resolutions better than ~10 Å. The position of these features can be pre-encoded in a geometric hash map. Using the C_α atom positions in α-helices and β-strands, atomic models can be fit into density maps by enumerating features in common between the density map and the atomic model. In BCL::EM-Fit tests presented here with both simulated and experimental density, initial fits that led to correct positions during refinement were distinguishable by their CCC. The accuracy of the final fit is dependent on the resolution of the density map, the Voxel size within the density map, and the resolution that is used to quantize the features within the hash map. MCM optimization with rigid body perturbation quickly and reliably refines the initial fit to a fit with the maximum CCC between the experimental and the simulated density map created from the atomic model. The BCL::EM-Fit algorithm provides an alternative method for docking of atomic models within cryoEM density maps.

CHAPTER III

BCL::SCORE - KNOWLEDGE BASED ENERGY POTENTIALS FOR RANKING PROTEIN MODELS REPRESENTED BY IDEALIZED SECONDARY STRUCTURE ELEMENTS

This chapter is a preproduction of the similarly titled co-first-author manuscript which will be submitted to “PLoS Computational Biology” co-authored by Mert Karakaş, Rene Staritzbichler, Ralf Müller and Jens Meiler.

Introduction

Many protein structures have been determined using experimental techniques like X-ray crystallography and Nuclear Magnetic Resonance NMR spectroscopy. Of the approximately 69,000 protein structures deposited in the Protein Data Bank (PDB) as of August 2011, X-ray crystallography [4] contributed 88% and nuclear magnetic resonance (NMR) [6] contributed almost all of the remaining 12% [5]. Although the number of experimentally determined protein structures grows, challenges still exist. Membrane proteins are hard to express, crystallize and are usually too large to be studied by NMR [68]. Some proteins evade atomic detail structure determination in isolation and adopt their biologically relevant structure only in the context of a complete biomolecular assembly, e.g. a virus or macromolecular machine [69].

The biological importance of these proteins justifies large efforts to collect limited experimental datasets that describe their structure. Often these data restrain the topology of the protein, i.e. the relative placement of secondary structure elements (SSEs). For example, electron density maps of medium resolution (4-10Å) obtained by X-ray crystallography or cryo-Electron Microscopy (cryo-EM) [2], [32], [35], [36] display the location of secondary structure elements but omit loop regions and side chains. Small-Angle X-ray Scattering (SAXS) and Small-Angle Neutron Scattering (SANS) display the overall shape of the protein topology [69], [70]. NMR spectroscopy of large and/or membrane proteins often yield distance and orientation restraints for atoms in the backbone of SSEs which are easier to label, assign, and interpret. Site-Directed Spin Labeling Electron Paramagnetic Resonance (SDSL-EPR) spectroscopy is applied to interrogate the relative positioning of SSEs relating the information from the tip of the non-natural and flexible spin label back onto the protein backbone [25], [71]. Lastly, cross-linking experiments interpreted with mass spectrometry yield typically distance restraints that again focus on the relative position of SSEs [26]. To facilitate construction and evaluation of protein structural models from such limited datasets a tailored energy function that only evaluates the relative positioning of SSEs in topologies would be of great value. Ideally, this energy function should predict the free energy of all states an amino acids sequence can access and the lowest free energy should be associated with the native structure [72]. In principle the free energy of a protein structure and its native conformation can then be derived with sufficient sampling of the potential energy surface using molecular mechanics force fields (e.g. CHARMM [29] or AMBER [73]). This approach is often computationally prohibitive and sometimes suffers from inaccuracies in the potential energy function. It has been shown that these potentials not always distinguish native-like from incorrect structures [74].

An alternative approach constructs scoring functions whose global energy minimum coincides with the native conformation for a database of experimentally determined protein structures of different sequence. Early versions of such knowledge-based or statistical potentials are based on contact frequencies [75] and likely exposure states of amino acid types [76]. Since then, a large variety of such potentials have been developed (for a review see [77]) and their applicability to fold recognition (threading) [76] and protein folding has been demonstrated [20], [21]. The underlying assumption that the knowledge based distribution of features is a BOLTZMANN-like distribution can be challenged e.g. for amino acid pair distances [30]. This is particularly true in protein structure prediction, where the reference state is dependent on the type and density of sampling used [78].

Knowledge based energy functions employ probability theory and in particular BAYES' theorem to circumvent the assumption of a Boltzmann distribution [30]. Shen and Sali derive a Discrete Optimized Protein Energy (DOPE) from a sample of native structures based entirely on probability theory [77]. The potential achieves enrichments between 3 and 9 for the identification of native structures in a set of models. Protein structure prediction with ROSETTA uses a low resolution knowledge-based scoring function consisting of an amino acid environment term defined by the burial of an amino acid and an amino acid pair interaction potential defined by all amino acid pair distances [20].]. It further includes a secondary structure packing potential for α-helix packing and β-strand pairing in β-sheets. A dot product captures hydrogen bonding in β-strand pairing. This potential uses the loop length connecting two SSEs as an additional dependent variable [21].

The energy function developed herein works off the hypothesis that interactions between SSEs define the core of the protein structure and are the major contributor to the stability of the protein fold, at least for a large fraction of folded proteins. In turn, the majority of stabilizing interactions in the protein structure is present in SSE-only models. Further, it is hypothesized that this part of the stabilizing interactions can be most accurately predicted as flexibility is reduced in the backbone of SSEs when compared to loop regions or amino acid side chains. The expected higher accuracy in placing the SSEs will result in a higher accuracy of the energetic evaluation. In result a smoothened energy landscape is expected that can be searched more readily as it is devoid of noise introduced by inaccurately placed of loop regions and side chains. The advantages of reduced conformational search space and smoothened energy landscape pair nicely with above-mentioned settings with limited experimental data as most experimental restraints relate to SSEs and can thus still be employed in protein folding. It is expected that models constructed and evaluated with this energy function can be readily completed through established protocols for the construction of loops and side chains. For example, loops can be modeled using fragment replacement [79], cyclic coordinate descent [80] or kinematic loop closure [81]. Side chains are added using dead end elimination or Monte Carlo sampling of rotamer libraries as implemented for example in SCWLR [82] and Rosetta [83].

The present manuscript introduces a comprehensive knowledge-based energy potential for proteins which is based on a simplified representation of the protein including only SSEs, i.e. α-helices and β-strands. The hypothesis is that for the majority of well-structured domains the assembly of the SSEs in three-dimensional space defines the domain topology, i.e. fold. Based on the amino acid C_β atom coordinates within the SSEs ($H_{\alpha 2}$ atom for Glycine) an amino acid pair potential, an amino acid environment potential, a secondary structure element packing potential, a β-strand pairing potential, a loop length potential, a radius of gyration potential, a contact order potential, and a secondary structure formation potential. Separate penalty functions forbid amino acid clashes, SSE clashes and loop distances that cannot be bridged. The overall energy potential is a linearly weighted consensus scoring function. These weights balance the individual terms to evaluate the native-likeliness of the SSE arrangement and the three dimensional placement of the amino acids in the context of the fold. While the scoring function is specialized to evaluate the loop less protein topology as defined by the SSEs, it can be applied to full chain protein models as well.

Results and Discussion

Bayes' theorem is applied to derive a comprehensive knowledge-based potential

In deriving the present knowledge-based potential we use BAYES' theorem to estimate the probability of a structure given the sequence. This strategy follows previously described approaches [20], [21] in expanding this probability into a series of terms that desribe certain aspect of the protein structure. This strategy avoids the requirement of BOLTZMANN-like distribution of states in the databank:

$$P(struct|seq) = P(struct) \times P(seq|struct) \times \frac{1}{P(seq)} \tag{8}$$

where seq is the amino acid sequence and $struct$ the protein's three dimensional structure. This approach separates the probability for a given sequence to fold into a certain structure into two terms. The probability of the structure, $P(struct)$, describes the relative arrangement of SSEs in space independent of their sequence. The probability of the sequence given this SSE arrangement, $P(seq|struct)$, evaluates placement of specific amino acids into these SSEs. For the protein folding problem the probability of the sequence $P(seq)$ is a constant. The terms $P(struct)$ and $P(seq|struct)$ will each be expressed as a product of multiple contributing terms $P_i(X)$.

The inverse Boltzmann relation converts probabilities into an approximation of energy

The collected probabilities $P_i(X)$ are converted into a free energy approximation using:

$$E_i(X) = -RT \times \ln\left(\frac{p_{i,observed}(X)}{p_{i,background}(X)}\right) \tag{9}$$

Where $E_i(X)$ is the energy function for X – being the feature observed, R – the gas constant, T – temperature, $P_{i,observed}(X)$ – the probability with which that feature was observed and $P_{i,background}(X)$ – the probability to observe that feature by chance. The normalization with $P_{i,background}(X)$ ensures that favorable states receive a negative energy, unfavorable states a positive energy. The energy unit RT is arbitrarily defined as 1 BCL energy unit (BCLEU).

The most direct approach computes the total energy as sum of all individual contributions. One disadvantage of this strategy is that double-counting of contributions through several energy terms is difficult to entirely prevent. Other features of protein folds will be ignored as they are not or only incompletely captured by the geometric features observed. To account for part of these inaccuracies, each energy term is scaled by an individual weight. This weight will be optimized to distinguish native-like from non-native models for a database of proteins.

$$E = \sum_i w_i \times E_i(X)$$

Another disadvantage of knowledge based potentials is the difficulty to assign an energy penalty to states not observed in protein structures. Typically small pseudo-counts are added which result in a positive energy. However, if a state is not observed at all, the energy assigned through a pseudo-count is arbitrary. To address this shortcoming, penalties for forbidden geometries are split into separate energy terms. Thereby the weight optimization procedure can assign a weight for these penalties independent from other contributions to the energy function.

While this approach is inherently imperfect it proved effective in the past. The resolution of protein models evaluated with the present energy function is too low to unambiguously distinguish native-like from non-native models based on energy alone. The objective of the energy function is to enrich for native-like topologies which can be done effectively in the presence of its inherent inaccuracies.

Ensure continuous differentiability of all geometric parameters and energy potentials

Traditionally some geometric parameters observed contain step functions. An example is the number of neighbors within a given distance cutoff which is often used as a measure of solvent exposure [21], [84]. To avoid discontinuities at the cutoff, a continuously differentiable transition function is often introduced into the definition of a feature:

$$trans^{+}(x_0,x_1,x) = \begin{cases} x \leq x_0, 0 \\ x\epsilon(x_0,x_1), \frac{1}{2}\left(cos\left(\frac{x-x_1}{x_1-x_0} * \pi\right)+1\right) \\ x \geq x_1, 1 \end{cases} \tag{10}$$

$$trans^{-}(x_0,x_1,x) = \begin{cases} x \leq x_0, 1 \\ x\epsilon(x_0,x_1), \frac{1}{2}\left(cos\left(\frac{x_0-x}{x_1-x_0} * \pi\right)+1\right) \\ x \geq x_1, 0 \end{cases} \tag{11}$$

In Figure 10A an example of $trans^{-}(x_0,x_1,x) = trans^{-}(4{,}11.4,x)$ is shown, which is used to smooth the neighbor count (described below). The different between $trans^{-}$ and $trans^{+}$ is that the first is a step-up, the latter is a step-down as a function of x. We demonstrated in the past that such a transition function allows for a neighbor count measure that is not only continuously differentiable but also more accurately approximates solvent accessible surface area [84].

Amino acid environment potential

This energy potential captures the preference of an amino acid to be buried and engage in hydrophobic interaction in the protein core or exposed and interacts with the solvent.

$$P(seq|struct) \cong \prod_{i} P(aa_i|e_i) \tag{12}$$

In order to measure burial a function that counts the neighbors of an amino acid was used (Figure 9A):

$$e_i = NC(aa_i) = \sum_{|i-j|>3} trans^{-}(r_{low}, r_{high}, r_{ij}) \tag{13}$$

Weighing the actual neighbor count between r_{low} and r_{high} smoothens the potential and enables gradient based minimizations. The thresholds have been optimized for a high correlation of the neighbor count value with the MSMS solvent accessible surface area (SASA) approximation implemented in the molecular visualization package VMD [85]. The lower threshold is set to 4.0 Å, the upper threshold to 11.4 Å [84]. A minimal sequence separation of three residues reduces the bias introduced by sequence proximity. This step is particularly necessary to accurately determine exposure at the end of SSEs. In SSE only protein models amino acids at the end of SSEs would otherwise have an artificially low neighbor count. The background probability distribution is the normalized sum of all normalized amino acid exposure distributions. Neighbor count bins that were empty or had one raw count were assigned a constant repulsive energy value of 18 BCLEU (Figure 9B).

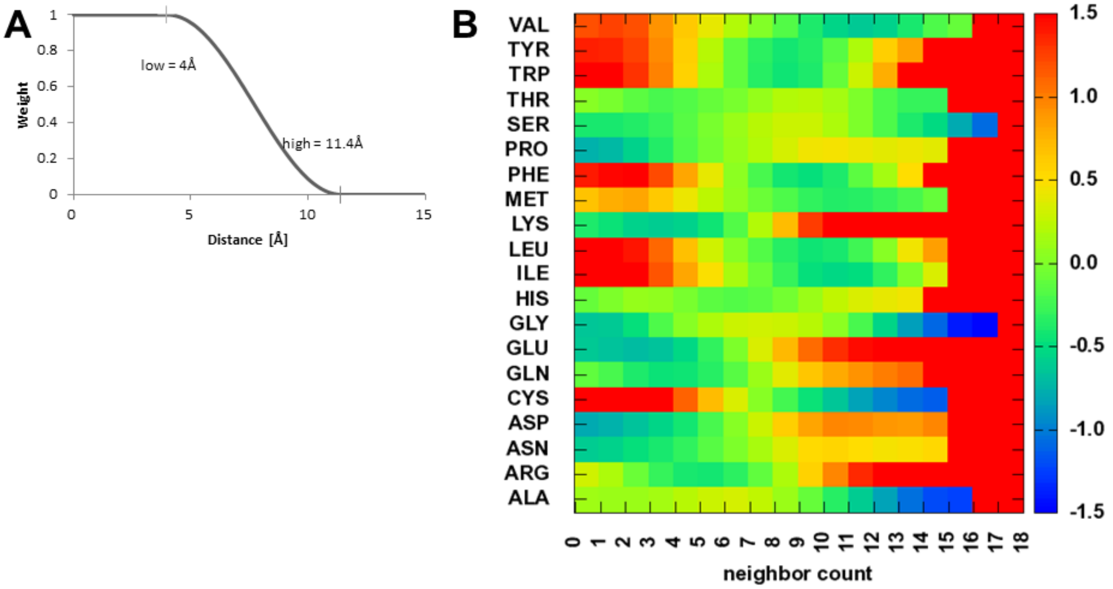


Figure 9 Amino acid neighbor count environment potential

A shows the transition function that is used between the lower and upper threshold in which the weight for the neighbor of considerations drops from 1 (4Å) to 0 (11.4Å) using half of a cosine function on the left. **B** shows the neighbor count energy potential for all 20 amino acids with their three letter code.

Amino acid pair distance potential

$P(seq|struct)$ is proportional to the amino acid pairs observed for a given distance.

$$P(seq|struct) \cong \prod_{i+12<j} P\left(aa_i, aa_j \middle| r_{ij}\right) \tag{14}$$

In order to define the interactions, statistics for the C_{β}-atom distance between pairs of amino acids $\left(aa_i, aa_j\right)$ have been collected. For Glycine, the $H_{\alpha 2}$ hydrogen position was used (Figure 10A). Distances have been collected between 0 and 20 Å in bins of size 1 Å. Amino acid pairs have been considered if they had a sequence separation of 12 residues $(i + 12 < j)$ in order to reduce the bias introduced by sequence proximity. For each bin the energy was approximated using the inverse BOLTZMANN relation. The expected background probability is estimated through the frequency of seeing aa_i or aa_j with any other amino acid at distancer_{ij}. Distance bins that had fewer than five raw counts were assigned a constant repulsive energy value of 18 BCLEU (Figure 10). Note that a separate penalty will forbid very close distances not observed in protein structures – i.e. van der Waals repulsion (read below).

The potentials obtained follow the expected trends (Figure 10B). For example, Leucine and Isoleucine are expected to interact favorably due to van der Waals (vdW) attraction, which is reflected in the negative energies for short distances. Arginine and Lysine with positively charged side chains are expected to experience Coulomb repulsion when approaching each other which is reflected in the positive energy for short C_{β}-atom distances. Tryptophan pairs may engage in π-stacking interactions, which are reflected in a preferred C_{β}-atom distance around 4 Å (β-strand pairing) and 8 Å (SSE packing). Arginine and Lysine are both positively charged and repel each other at close proximity as reflected in the positive energies until 10 Å. Note that a separate penalty controls very close distances not observed in protein structures – i.e. van der Waals repulsion (read below).

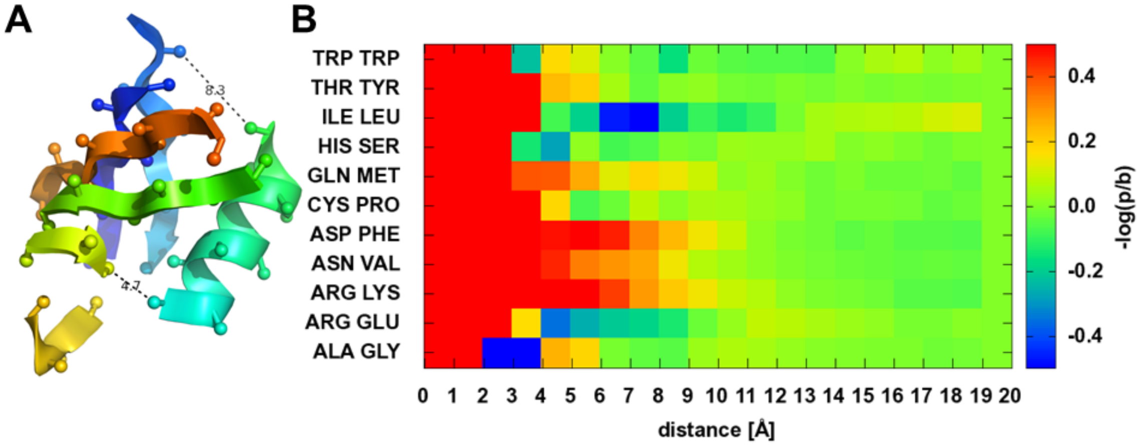


Figure 10 Amino acid pair distance potentials

In **A** the idealized structure of 1ubi with C_β and $H_\alpha2$ 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** shows selected amino acid pair distance potentials for Trp-Trp as an example for π-stacking interaction, ILE-LEU as an example for VDW apolar interaction, ARG-GLU as an example for Coulomb attraction, and Arg-Lys as an example for Coulomb repulsion.

Loop length potential

SSEs are connected by loop or coil regions whose coordinates are not explicitly considered in the present approach to score protein folds. However, there are preferences for loops of a certain length d_S to bridge a certain EUCLIDEAN distance d_E (Figure 11A). This is a sequence-independent score contributing to $P(struct)$. Note that the requirement that two SSEs can be physically linked with a fully extended loop is controlled by a separate loop closure penalty (read below).

$$P(struct) \cong \prod_{i<j} P\left(d_E(SSE_i, SSE_j)\middle|d_S(SSE_i, SSE_j)\right) \tag{15}$$

$d_S(SSE_i, SSE_j)$ Sequence distance between last residue of SSE_i and first residues of SSE_j

$d_E(SSE_i, SSE_j)$ EUCLIDEAN distance between end of main axis of last fragment of SSE_i and beginning of main axis first fragment of SSE_j (Figure 12E)

The background probability is set to $P\left(d_E(SSE_i, SSE_j)\right) = d_E^2$ (Figure 11B). For short sequence distances it is favorable that the EUCLIDEAN distance is short. Long EUCLIDEAN distances are forbidden by a constantly increasing positive energy which is a result of the pseudo count divided by the square of the EUCLIDEAN distance. EUCLIDEAN distances below 4 Å are generally possible but are only preferred for loops of length 0 and 1 which occur in the database for bent and kinked SSEs. There is a nearly linear dependency between the sequence separation and the EUCLIDEAN distance for up to 7 residues in the loop. The maximally possible EUCLIDEAN distance increases linearly to a distance of approximately 32Å at 10 residues. EUCLIDEAN distances longer than 32Å are rarely observed in this database of globular proteins. As loops get longer, the range of EUCLIDEAN distance they bridge becomes wider.

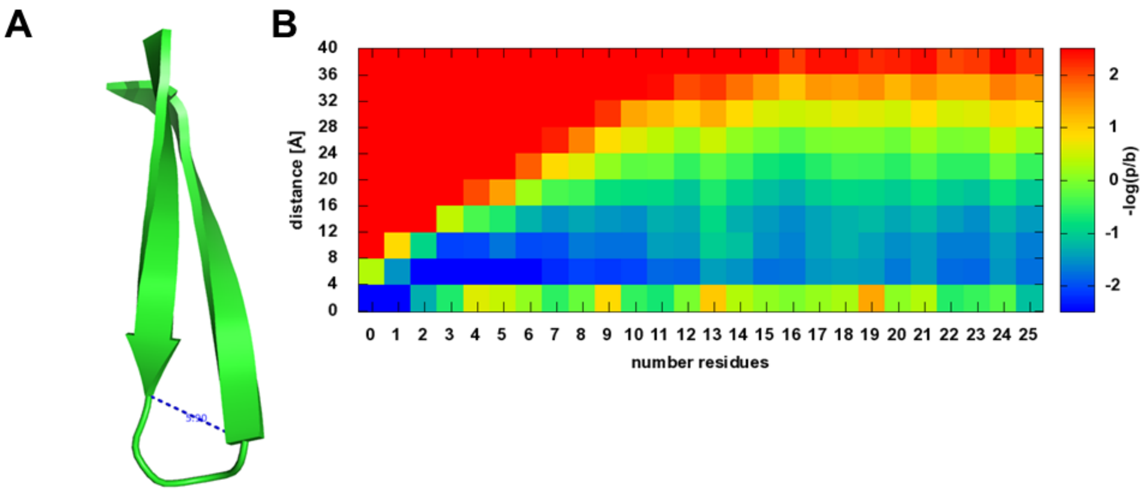


Figure 11 Loop length potential

A describes 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** describes 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.

β-Strand pairing potential

This potential evaluates the pairing of two β-strand SSEs to form a β-sheet contact.

$$P(struct) \cong \prod_{i<j} P\left(pair\left(SSE_i, SSE_j\right) \middle| SSE_i, SSE_j\right) \tag{16}$$

To compute $pair\left(SSE_i, SSE_j\right)$ both strands are decomposed into overlapping fragments of three amino acids (Figure 12E). A β-sheet contact then is defined as a series of l pairs of aligned fragments. The distance d and torsion angle θ between each pair of fragments is evaluated (Figure 12A). Further, a weight $w_{\beta\beta-pair}$ is used to distinguish a planar arrangement of two β-strands (β-strand pairing) from an opposing arrangement (β-sheet packing, Figure 12D, details see Methods). l is limited to the number of fragments in the shorter SSE:

$$pair\left(SSE_i, SSE_j\right) = \prod_{1<k<l} P\left(d_k, \theta_k, w_{k,\beta\beta-pair}\right)$$

d_k shortest, orthogonal distance in fragment pair k

θ_k torsion angle at shortest, orthogonal distance in fragment pair k

$w_{k,\beta\beta-pair}$ weight that decreases as the arrangement deviates from planar β-strand pairing

The potentials represents the likelihood of observing a given distance between the center of two β-strand fragments and a given twist of two β-strand fragments (Figure 13A) with respect to each other. Note that the potential omits explicit evaluation of backbone hydrogen bonds to keep the energy landscape smooth. The background probability is assumed to be proportional to d since the chance to find a second β-strand by chance in a parallel arrangements grows approximately linearly with the distance of the object, similar to the girth of a circle.

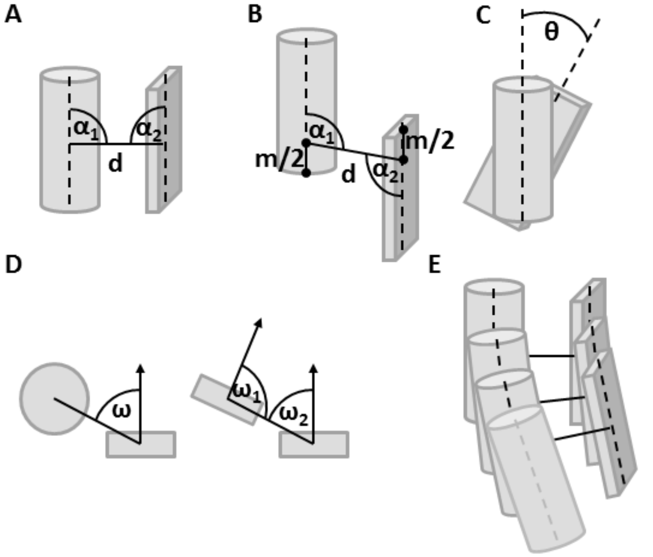


Figure 12 SSE Fragment packing

SSE fragments are shown with their geometric packing descriptors. **A** α_1 and α_2 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. ω_1 and ω_2 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).

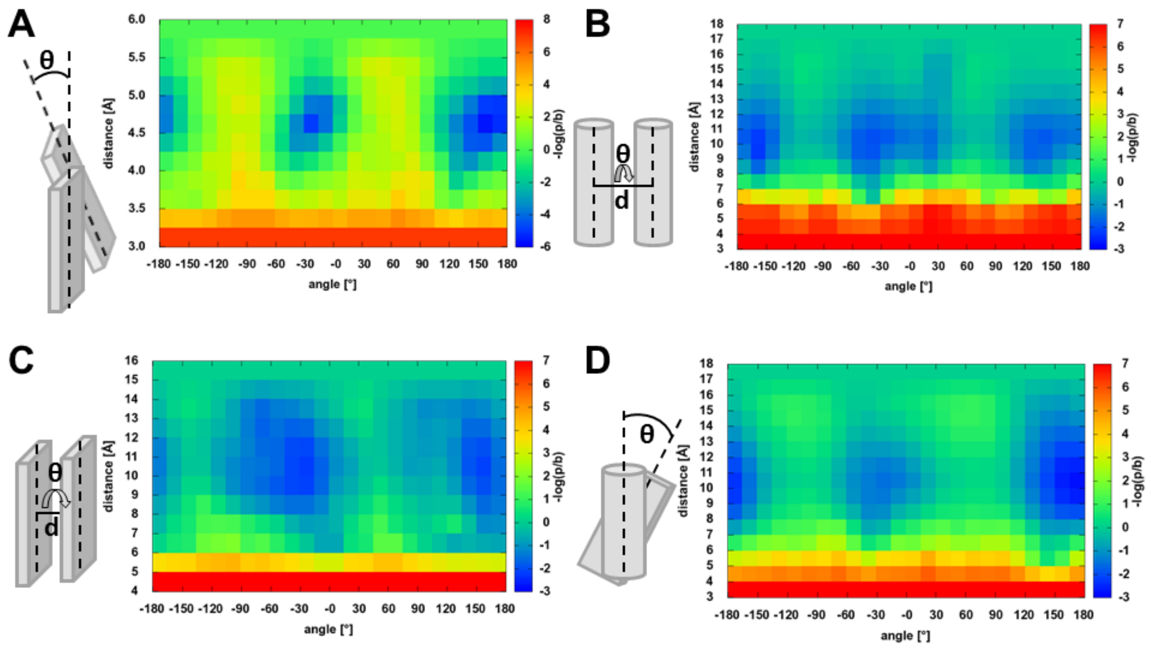


Figure 13 Strand pairing and SSE packing potential

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

Secondary structure element packing potential

While β-strand pairing is defined by backbone hydrogen bonds, SSE packing is driven through side chain interaction. In result distance and torsion angles are less tightly controlled which is why we treat both potentials separately. Other than that, SSE packing potentials have been derived in a fashion similar to the β-strand pairing potential.

$$P(struct) \cong \prod_{i<j} P\left(pack\left(SSE_i, SSE_j\right) \middle| SSE_i, SSE_j\right) \tag{17}$$

To compute $pair(SSE_i, SSE_j)$ both SSEs are decomposed into overlapping fragments of three amino acids (β-strands) and five amino acids (α-helices, Figure 12E). A contact then is defined as a series of l pairs of aligned fragments. The distance d and torsion angle θ between each pair of fragments is evaluated (Figure 12, Figure 13A).

$$pack(SSE_i, SSE_j) = \prod_{1<k<l} P(d_k, \theta_k, w_{k,pack}) \tag{18}$$

d_k shortest, orthogonal distance in fragment pair k

θ_k torsion angle at shortest, orthogonal distance in fragment pair k

$w_{k,pack}$ weight that decreases if β-sheets in the packing interact via their edge

The term $w_{k,pack}$ is dependent of the types of SSEs in the packing. For the helix-helix interaction $w_{\alpha\alpha-pack} = 1$. For helix-strand interactions $w_{\alpha\beta-pack}$ decreases from 1 if the face of the β-strand points away from the α-helix. For β-sheet packing $w_{\beta\beta-pack}$ decreases from 1 if the β-strands don't face each other (Figure 12D, details in Methods). The background probability is assumed to be proportional to d. The resulting potentials plot energy with respect to distance and twist angle.

Contact order score

Using the assembly of SSEs to describe the topology of a protein enables the optimization protocol to sample topologies with many non-local contacts. One measure for the complexity of the topology is the contact order. Contact order CO is defined as the average sequence separation of all amino acids in contact, conventionally identified by the closest heavy atom distance between two amino acids <= 8Å [86]. In this score, the

C_β-C_β distance is used. A larger contact order constitutes a more complex topology. The contact order score is added to restrain the models constructed to a likely contact order range. To ensure comparability we normalize the square of the contact order with the sequence length to compute $NCO = CO^2/length$. For native proteins, NCO is largely independent of sequence length being in the range of 0.25 to 0.60 (Figure 14). An energy term (Figure 15A) was added based on the hypothesis:

$$P(struct) \cong P(NCO) \qquad (19)$$

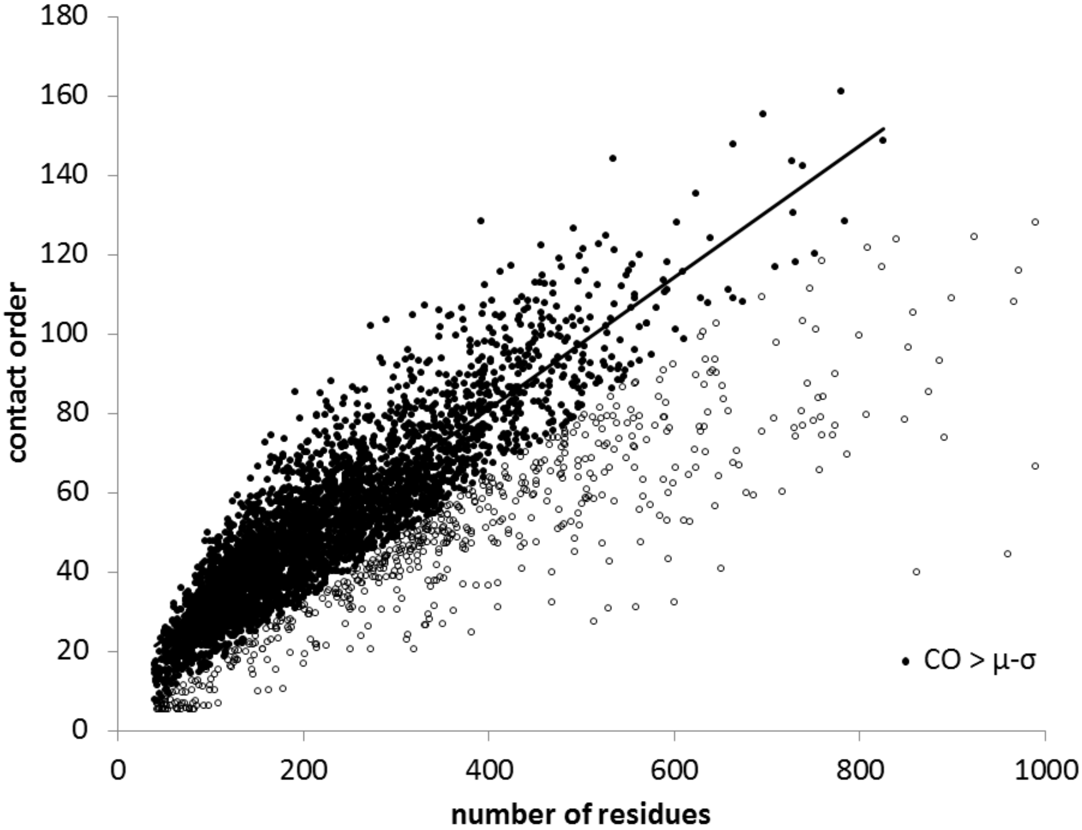


Figure 14 Contact order vs. chain length

Plotted is the amino acid chain contact order of 4303 protein chains. Empty circles have a ratio below the 86% statistical confidence interval and are not considered for the potential (475 chains). The filled circles with the linear fit line are CO/length ratios that are considered for the potential.

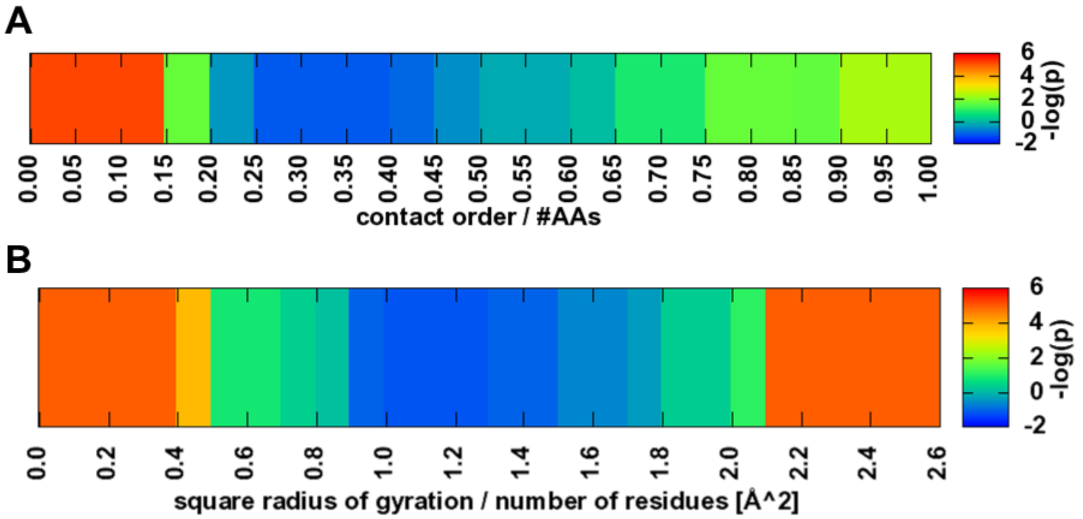


Figure 15 Contact order and Square radius of gyration potential

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 [20]. It is computed as the mean square distance of all C$_{\beta}$ atom coordinates (H$_{\alpha2}$ for Glycine) to their mean position:

$$R_{gyr}^{2} = \frac{1}{n}\sum_{i=1}^{n}(r_i - r_{mean})^2 \tag{20}$$

The term $e^{-R_{gyr}^{2}}$ can directly be used to estimate $P(struct)$ if sequence length is constant [87]. To enable our energy function to compare proteins of variable length e.g. during the assembly from SSEs, we introduce a normalized radius of gyration $NR_{gyr} = R_{gyr}^{2}/length$. For native proteins, NR_{gyr} is largely independent of sequence length being in the range of 0.8 to 2.0 (Figure 16). An energy term (Figure 15B) was added based on the hypothesis:

$$P(struct) \cong P(NR_{gyr}) \tag{21}$$

Extended α-helical coil-coiled structures as well as protomers that form obligate oligomers were removed prior to obtaining this statistics.

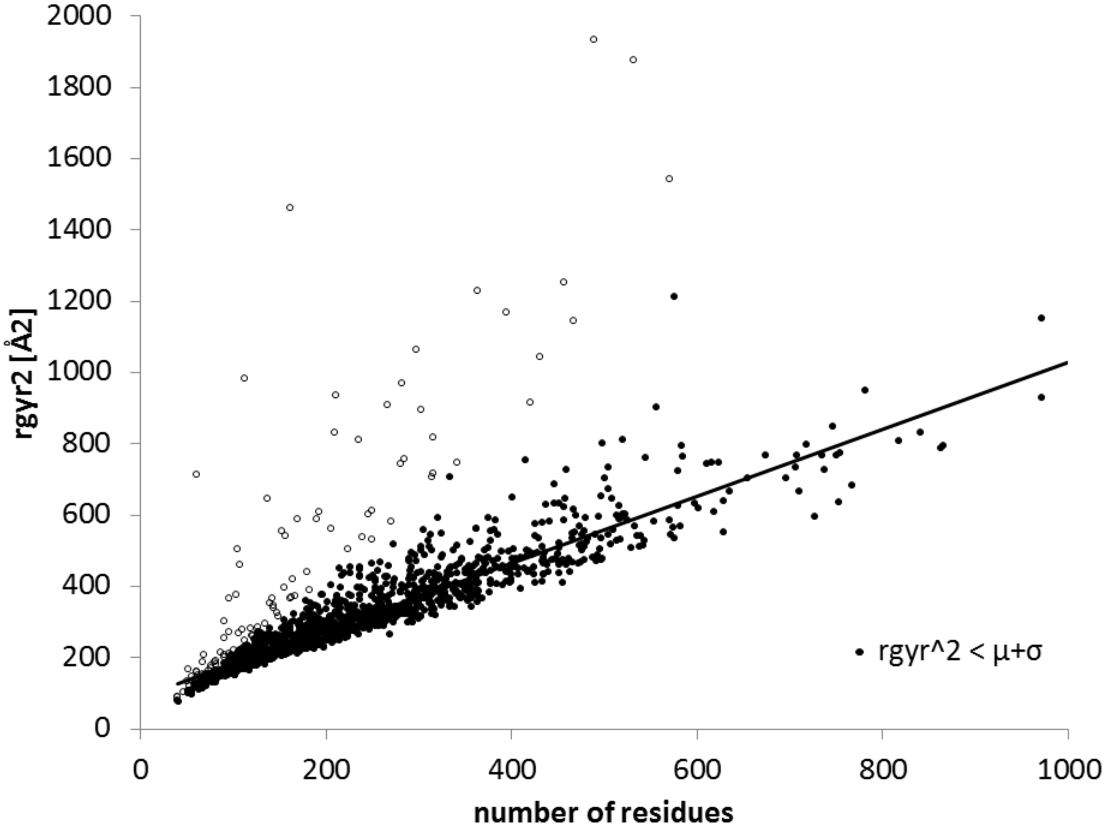


Figure 16 Square radius of gyration vs. chain length

Plotted are the amino acid chain square radius of gyration of 1342 single chain proteins. Empty circles have ratios below the 86% statistical confidence interval and are not considered for the potential (96 proteins). The filled circles with the linear fit line are rgyr^2/length ratios that are considered for the potential.

Secondary structure prediction agreement

Given an amino acid sequence, JUFO [14] and PSIPRED [15] calculate probabilities for each amino acid to be part of an α-helical, β-strand or a coil secondary structure element. Those prediction methods average a per-residue accuracy of up to 80%. This fact can be used, to evaluate the per-residue assigned secondary structure for a given protein model.

$$P(seq|struct) \cong \prod_{i} P(aa_i|SS_i) \tag{22}$$

SS_i secondary structure of amino acid i in the structure

Due to the inaccuracies in the secondary structure predictions, a mean probability and standard deviation for the probability for actual secondary structures are derived, and the error function of the standard score is defined as the potential used:

$$E_{SSPred} = \sum_{i} -erf\left(\frac{p_{SS,i} - \mu_{SS}}{\sigma_{SS}}\right) \tag{23}$$

$p_{SS,i}$ probability of the assigned Secondary structure in the model

μ_{SS} mean probability for accurately predicted secondary structure

σ_{SS} standard deviation for accurately predicted secondary structure

The use of the standard score makes it possible to use different secondary structure prediction methods, of different sensitivity and dynamic range of probabilities. The error function projects the standard score in a less sensitive range if probabilities strongly disagree with the average. The following parameters have been found for JUFO and PSIPRED:

	μ_{SS} helix	σ_{SS} helix	μ_{SS} strand	σ_{SS} strand	μ_{SS} coil	σ_{SS} coil
JUFO	0.67	0.21	0.58	0.24	0.59	0.18
PSIPRED	0.76	0.20	0.71	0.27	0.73	0.21

Amino acid clash, SSE clash and Loop closure potentials

A difficulty with knowledge based potentials is that a BOLTZMANN-like distribution is assumed for the dataset used to derive the potentials from. Although all potentials described above are based on probabilistic theory, they are ambiguous to geometries absent in native structures. Since no counts are observed for these geometries the associated energies would be infinitely high. The precise penalty for such non-native features remains difficult to determine. However, while the energy will be elevated it will not be infinite. Often one pseudo count for every observation is added (according to the rule of succession, "LAPLACE rule") giving all non-observed events an equally high penalty. To enable fine-tuning of the energy penalties in regions of non-observed events separate energy components are introduced. This procedure allows independent choice of a weight changing the penalty amplitude in "structural forbidden" regions. The procedure has a second advantage: vdW repulsion, is affiliated with steeply rising energies over a small change in distance. A separate potential allows for a finer binning of these penalty potentials when compared to the attractive counter-parts.

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 counts > 1, when iterating from shorter distances to larger distance, was determined to be the minimum permitted distance. Using this threshold, a “penalty” function is defined:

$$P(struct|seq) \cong \prod_{i<j} P\left(aa_i, aa_j \middle| r_{ij}\right) \tag{24}$$

$$E_{AAclash} = \sum_{i<j} trans^{-}\left(m\left(aa_i, aa_j\right) - 1\text{Å}, m\left(aa_i, aa_j\right), r_{ij}\right) \tag{25}$$

$m\left(aa_i, aa_j\right)$ Shortest allowed distance for amino acid type pair

r_{ij} Distance between amino acid pair

This term is complementary to the amino acid pair distance potential. If the distance between two amino acids is below the allowed distance for this pair of amino acid types, a positive, penalty energy is ramping reaching its maximum at 1Å below the allowed distance. A matrix of minimal distances for all amino acids types is depicted in the Figure 17.

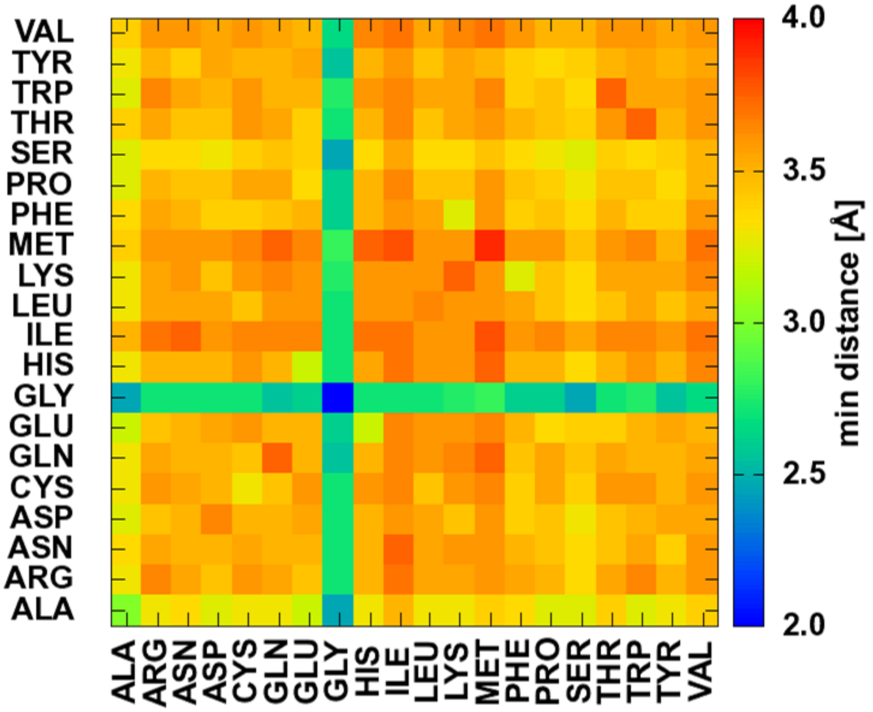


Figure 17 Minimal distances between amino acid pairs

The minimal distances determined by C_β atom distance or HA2 for GLY. The distances are color coded. Shorter distances like for Glycine are blue and green, little longer distances like for Alanine are yellow, while long distances go up to red.

SSE clash potential

Although the amino acid clash potentials suffices in “detecting” clashes of side chains in the packing of SSE, it does not penalize special cases of overlapping SSEs. An example for these kinds of topologies is when one β-strand is positioned on top of another β-strand but offset by one amino acid. C_β atoms point in opposite directions avoiding any clash while backbone atoms are not explicitly modeled. To prevent such situations a clash term that is based on the packing SSE fragments was derived. From unoccupied bins in the SSE packing and pairing potentials (Figure 12) minimal distances between two SSE fragments have been defined as α-helix/α-helix 4Å, α-helix/β-strand 4Å, β-strand/β-strand 3Å:

$$P(struct) \cong \prod_{i,j,k,l} P\big(d\big(F_{i,k,}F_{j,l}\big)|F_{i,k,}F_{j,l}\big) \tag{26}$$

$$E_{SSEclash} = \sum_{i<j,k} trans^{-}\big(m\big(SSE_{i,k,}SSE_{j,k}\big) - 1\text{Å}, m\big(SSE_{i,k}\big), d_{i,j,k}\big) \tag{27}$$

$m\big(SSE_{i,k,}SSE_{j,l}\big)$	Minimal allowed distance for aligned fragment pair k of SSEs i and j
$d_{i,j,k}$	Length of shortest connection between the two SSE fragments

This term is complimentary to the SSE packing and β-strand pairing potential. If the distance between two SSE fragments is smaller than $m\big(SSE_{i,k,}SSE_{j,l}\big)$, a positive energy is the result. The full positive energy is reached if the distance is 1Å below the allowed distance for that pair of SSE types.

Loop closure potential

In order to guarantee the possibility to close loops it proved necessary to add steep penalty if the EUCLIDEAN distance becomes too long. In contrast to the loop length potential, the loop closure constraint only considers SSEs adjacent in sequence. The EUCLIDEAN distance between the terminal C atom and the starting N atom of the following SSEs d_{CN} is evaluated.

In native proteins d_{CN} is generally shorter than $d_{CN}^{limit} = 2.11 + 2.56 \times loop\_length$. This relation was obtained by selecting the EUCLIDEAN distance for a loo length, which is the 5th percent of the longest distances. For length between one and twenty amino acids in the databank, a linear regression was fitted (Figure 18). We evaluate therefore $\Delta d_{CN} = d_{CN} - d_{CN}^{limit}$:

$$P(struct) \cong \prod_{i}^{n-1} P\left(\Delta d_{CN,i} | (SSE_i, SSE_{i+1})\right) \quad (28)$$

$$E_{LoopClosure} = \sum_{i}^{n-1} trans^{+}(0\text{Å}, 1\text{Å}, \Delta d_{CN,i}) \quad (29)$$

This potential is complimentary to the loop length potential. It forbids loops that cannot be closed because of too large EUCLIDEAN distance. Additionally, it measures the distance between the two atoms, that are the bases for the loop, while the loop length potentials is using a more crude estimation for the ends of the SSEs using only the tips of the fragment main axes.

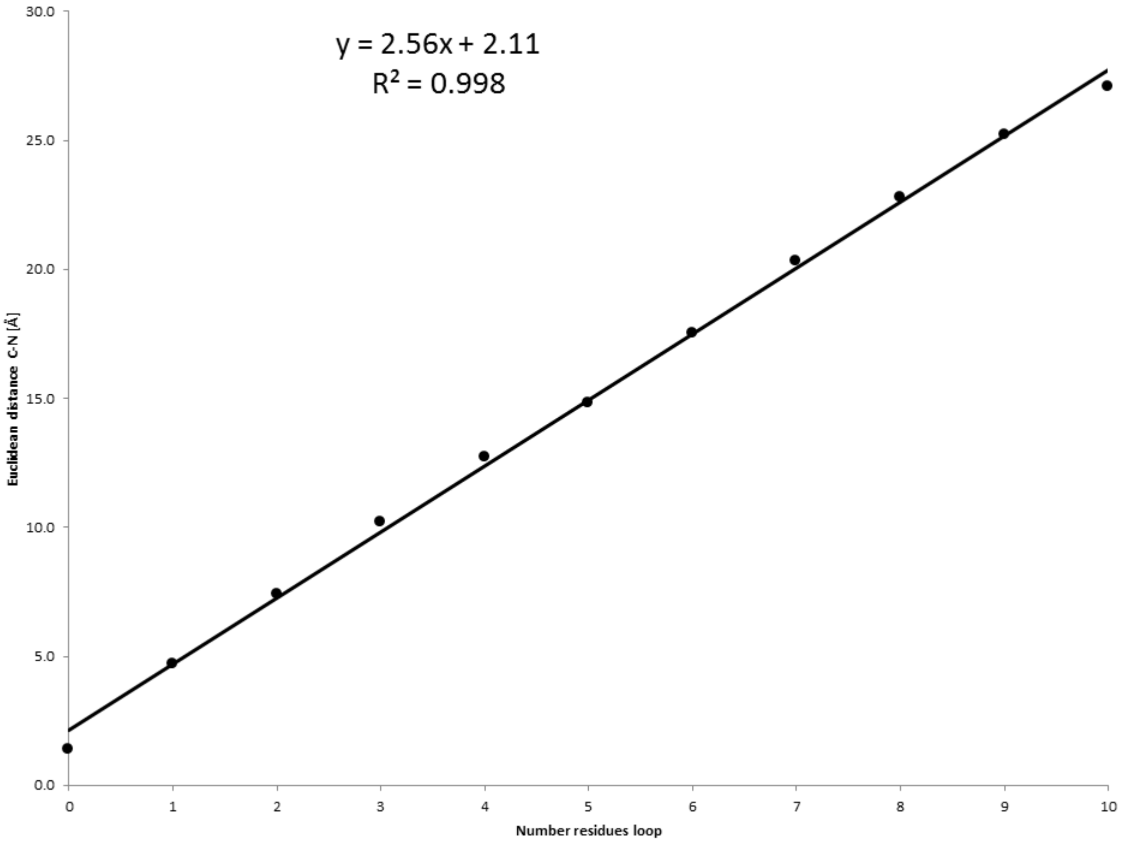


Figure 18 Maximal loop length extension

95% of the longest loop extensions as distance between the backbone carbon and nitrogen atoms vs. the number of residues in the loop. A linear fit shows the trend and can be used to estimate possible loop bridging distances.

53 protein model sets have been generated using ROSETTA, a BCL Perturbation protocol and a BCL Folding protocol

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) Re-assembling the SSEs leading to protein models of various arrangements and topologies. Details on the protocols are described in the Methods section.

The rational for usage of three separate sets of protein models was to maximize diversity in the models thereby maximizing generalizability of the scoring function. The identification of native-like structures was based on two measures: (1) GDT_TS < 25% [35] and (2) RMSD100 < 8Å [36]. The percentage of native-like models varies between 0 and 99.5% for the protein model sets. Only model sets with percentage of native-like models between 1% and 99% have been used for the analysis in a ten-fold cross validation calculation of enrichments. The cross validation subsets have been generated by randomly removing models so that each subset contained 10% correctly folded models and 90% incorrect models.

Enrichment is a good measure to evaluate the performance of an energy potential

Figure 19 shows a representative RMSD100-energy plot of a set of protein models that was prepared to contain 10% of native-like models below an 8 Å RMSD100 cutoff. The 8 Å cutoff is based on the observation, that two protein models typically share the same topology below that measure. The horizontal line denotes the best 10% of the models with respect to the scoring function used. Models that are below the RMSD100 cutoff are positives (P), and if they are below the energy of the best 10% by energy, 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 native-like models by RMSD100. The enrichment is now defined by the ratio of true positives within the 10% native-like models $(TP + FN)$ divided by the initial ratio of native-like models by RMSD100 cutoff to the total number of models $(TP + FN + FP + TN)$.

$$enrichment = \frac{TP}{TP + FN} * \frac{P + N}{P} \tag{30}$$

In this manuscript is adjusted to be $(P + N)/P = 10$ limiting the maximal enrichment to 10. An enrichment of 1 corresponds to no improvement. Enrichment values smaller than 1 suggest that the score deselects native-like arrangements.

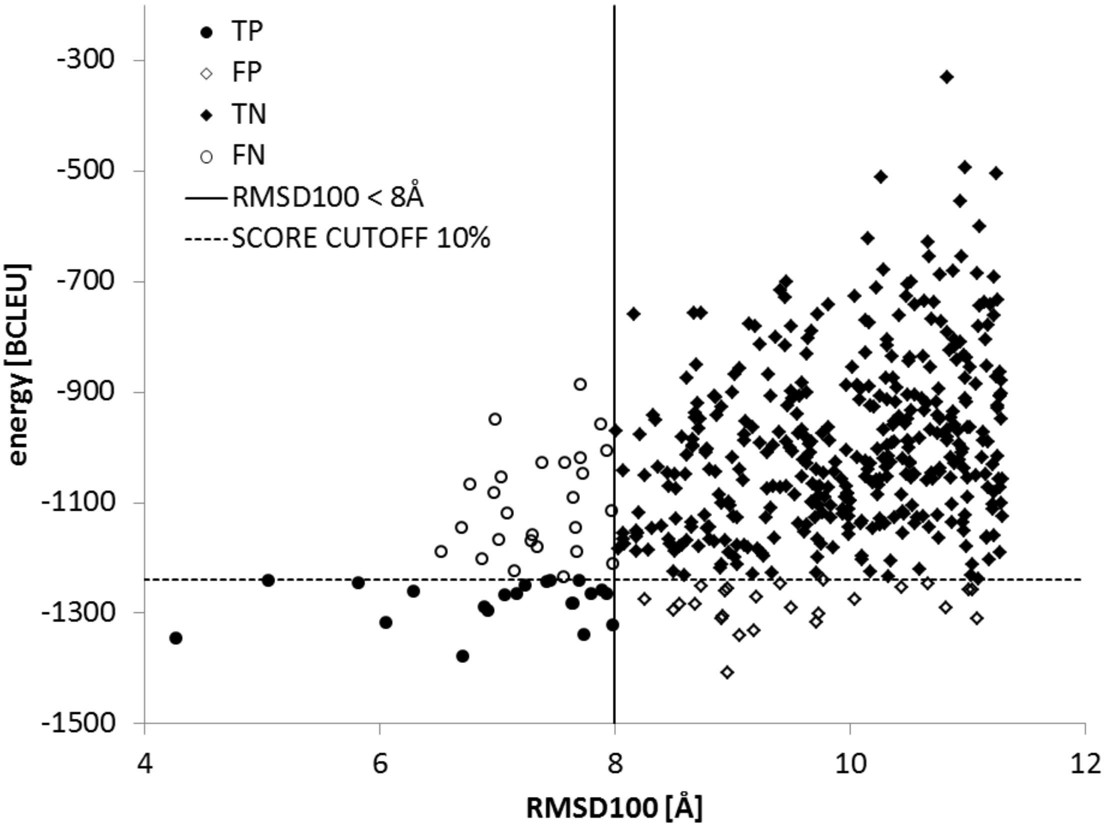


Figure 19 Schematic RMSD vs energy plot reprenseting classififcation for enrichment

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 native-like models with RMSD100 cutoff around 8Å.

Benchmark enrichment of native like structures through potentials

Table 10 contains enrichments for the 53 protein sets from three different methods each, and the various scores. Note that the number of proteins considered can be smaller than 53 if an insufficitient number of native-like models was in the dataset (read above). Statistical significance was established by computing the average enrichment over 10 cross-validations, subtracting 1.0 (baseline), and deviding the result with the standard deviation of the enrichment (Z-score). The percent of models sets that could be enriched by a statistical significant factor are reported (Z-score > 1.0, **Error! Reference source not found.**). Enrichments for the three penalty functions are also reported in Table 10. Individual components of the scoring function generally discriminate well against random models for the BCL folded and perturbed structures but do perform worse for ROSETTA folded models. We attribute this observation to the fact that ROSETTA folded models will generally score well in the present energy function due to the similarity of the two scoring functions. The amino acid pair distance, amino acid neighbor count and the SSE packing potentials achieve enrichments for nearly all the protein sets. The secondary structure prediction program potentials using PSIPRED secondary structure probabilities help for ROSETTA and perturbation model sets, which have varying SSE content. BCL folded models cannot be discriminated, since the secondary structure is fixed and the predictions are used to define the secondary structure that was assembled into models. The consensus scoring function enriches significantly (67% of ROSETTA, 77% of perturbation model sets for RMSD < 8Å). No statistically significant improvement for BCL folded models is observed. We attribute this to the fact that these models were subject to energy evaluation with the scoring function with non-optimized weights creating a circular dependence. Considering the performance in respect to GDT_TS > 25%, for the three different models sets, 80%, 94% and 83% have a significant enriched model sets for ROSETTA as well as BCL perturbed and folded model sets.

BCL::Score C_β-centered potentials resemble first principles of physics and chemistry of amino acid interaction

The scoring function was developed for protein models consisting out of disconnected idealized SSEs. The absence of atomic-detail in the SSE-only protein models inherently prevents unambiguous identification of the native conformation in a set of models. Nevertheless, the amino acid pair potential and the amino acid environment potential both resemble native-like arrangements of amino acids. The environment potential follows the expected trend preferring around three neighbors for the negatively charged Glutamate residue but around eleven neighbors for the apolor Valine. For Glycine two minima are observed – very few and very many neighbors. This is somewhat counter-intuitive as Glycine prefers exposed positions in loop regions. However, the present potential maps $P(aa_i|e_i)$ – i.e. given a certain exposure value, which amino acid is likely. In densely packed positions with an extremely high number of neighbors only Glycine will fit giving it the high probability for such positions. Positions with neighbor counts above twelve are rare in folded proteins and should therefore be disfavored when predicting protein structures. However, this fact will be represented by $P(struct)$ and is correctly omitted in $P(seq|struct)$. Leucine and Isoleucine are expected to interact favorably in the pair potential due to van der Waals (vdW) attraction, which is reflected in the negative energies for short distances (Figure 10B). Arginine and Lysine with positively charged side chains are expected to experience COULOMB repulsion when approaching each other which is reflected in the positive energy for short C_β-atom distances. Tryptophan pairs may engage in π-stacking interactions, which are reflected in a preferred C_β-atom distance around 4 Å (β-strand pairing) and 8 Å (SSE packing).

Arginine and Lysine are both positively charged and repel each other at close proximity as reflected in the positive energies until 10 Å. These findings imply that for reduced SSE-only protein models a C_β atom side chain representation ($H_{\alpha2}$ for Glycine) is sufficient to estimate $P(seq|struct)$.

Secondary structure element arrangement determines the domain topology

The preferential arrangement of SSEs in a protein domain results from the sum of many atom-atom interactions. In the absence of atomic-detail in SSE-only protein models, BCL::Score knowledge-based potentials derived from $P(struct)$ discriminate native-like SSE arrangements. An optimal β-strand distance between 4.25 and 5.00 Å is observed. The optimal twist angle is around -15° (parallel β-strand contact) and 165° (anti-parallel β-strand contact). A twist angle of 165° is more pronounced as anti-parallel β-strand contacts are slightly overrepresented in the database. 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 9-12 Å (Figure 13B). For α-helix-β-sheet packing, the anti-parallel case with angles between 150° and 180° is most common as seen in the TIM-barrel fold or other "ROSSMAN-Folds" [88] (Figure 13D). As in the α-helix-α-helix packing, the optimal distance is around 9-12 Å. β-sandwiches pack with a distance of 9-12 Å and twist angles of -30° or 150° (Figure 13C). Twist angles lead in general to an improved packing as the interacting side chains can reach into gaps left by the side chains of the opposite SSE [89]. Ridges and grooves are formed on the surface of helices. These ridges are formed by residues usually separated by four in sequence. This model explains the predominant packing angle of around 50°.

Enrichments are reduced due to the incomplete, reduced representation of protein structure

There are two major explanations as to why maximum enrichment for any of the score for any set is never above five. Firstly, the protein models used are incomplete. Contributions of loop and coil regions to the overall energy are neglected resulting in inherent inaccuracies. Secondly, amino acids are represented by their C_β-atom only. This procedure introduces additional inaccuracies in the energetic evaluation. As discussed in the introduction, these inaccuracies are taken into account to enable a more rapid sampling of domain topology specifically in a limited experimental data setting. Nevertheless, BCL::Score knowledge-based potentials enrich a divers set of decoys with enrichments up to 7 for individual proteins. This is a respectable achievement when keeping in mind that the protein models are created using an energy function that necessarily covers some or even most aspects of the BCL::Score knowledge-based potential, i.e. most models created with these methods are expected to generally score well with BCL::Score.

Enrichment was achieved for a diverse set of protein models regardless of the sampling algorithm

Although ROSETTA generates low resolution models, they have a complete and defined backbone conformation. All BCL::Score potentials except for the loop length and contact order score can enrich ROSETTA models for native like conformations. It is expected that the loop length potential will not enrich ROSETTA models as they have a continuous amino acid chain. The loop length potential enriches BCL perturbed and folded structures with a discontinued amino acid chain. Due to the unrestrained sampling of the secondary structure elements, loops are violated and the potential is penalizing this arrangement. 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 unlikely in ROSETTA models which are biased towards lower contact orders. As expected, the β-strand pairing score contributes only for β-strand containing proteins. The radius of gyration score performs well for proteins < 150 residues, but seems to degrade for larger proteins. It can be observed that for GDT_TS and RMSD100 classification, the percentage drops under 50% for the BCL perturbed structures. This is expected as this decoy set was created to preserve protein size and relative positioning of SSEs that is native-like but create non-native topologies. For this decoys set we also observe the best discrimination for native like models. The weighted sum of individual terms performs comparable over all benchmark sets and shows that a linear combination can overcome some weaknesses of the individual terms.

Conclusions

A knowledge-based scoring function is presented optimized for SSE-only models. It enriches native-like topologies in diverse sets of protein models. We expect this scoring to be beneficial for certain settings in *de novo* protein structure determination: (1) When folding large proteins with complex topology simultaneous sampling of SSE arrangements and loop conformations creating a size limit for *de novo* protein structure determination. The BCL::Score potential for SSE-only models allows sampling of SSE arrangement independent and prior to the sampling of loop conformations. This approach has the potential to increase the size limit in *de novo* protein structure determination. (2) Limited experimental dataset often restrain the position of SSEs, for example density maps obtained form cryo-Electron Microscopy [90] or EPR distance restraints [91]. We expect that the present potential can be applied to assemble the topology of large proteins from such datasets. In fact, an early version of BCL::Score has been successfully applied to medium resolution density maps form cryo-Electron Microscopy [2].

Table 10 Enrichment of sets of protein models

RMSD100 < 8Å		total	amino acid clash		amino acid distance		amino acid neighbor count		contact order		loop length		loop closure		radius of gyration		SSE clash		SSE packing		strand pairing		SSPred JUFO		SSPred PSIPRED		sum	
Enrichment change			↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓	↑	↓
all	rosetta	18	44	17	**72**	17	**56**	17	44	39	22	**72**	44	50	**61**	33	50	44	**100**	0	33	44	**56**	28	**78**	17	**67**	22
	perturbation	53	**100**	0	**98**	2	**96**	2	21	**74**	**94**	4	**98**	0	49	45	**96**	4	**89**	9	**57**	38	47	45	**60**	36	**77**	23
	fold	14	**64**	29	**57**	29	29	**57**	29	**64**	**64**	21	**79**	14	29	50	36	43	29	**57**	0	**86**	29	**71**	29	**71**	43	50
α-helical	rosetta	12	**58**	17	**83**	8	**58**	17	42	42	25	**75**	50	50	**58**	42	**67**	33	**100**	0	17	50	50	33	**67**	25	**58**	25
	perturbation	24	**100**	0	**96**	4	**92**	4	17	**79**	**92**	8	**100**	0	**58**	33	**92**	8	**75**	21	4	**83**	42	50	46	50	**63**	38
	fold	10	**60**	30	**70**	20	30	**60**	30	**60**	50	30	**80**	20	20	**60**	30	40	40	40	0	**100**	40	**60**	40	**60**	50	50
β-sheet	rosetta	3	0	0	**67**	33	33	33	**100**	0	0	**67**	33	**67**	33	33	33	33	**100**	0	**67**	33	33	33	**100**	0	**67**	33
	perturbation	8	**100**	0	**100**	0	**100**	0	38	50	**100**	0	**100**	0	50	38	**100**	0	**100**	0	**100**	0	25	**63**	25	**63**	**75**	25
	fold	3	**67**	33	33	**67**	33	33	33	**67**	**100**	0	**67**	0	**67**	33	**67**	33	0	**100**	0	**67**	0	**100**	0	**100**	33	**67**
α/β	rosetta	3	33	33	33	33	**67**	0	0	**67**	33	**67**	33	33	**100**	0	0	**100**	**100**	0	**67**	33	**100**	0	**100**	0	**100**	0
	perturbation	21	**100**	0	**100**	0	**100**	0	19	**76**	**95**	0	**95**	0	38	**62**	**100**	0	**100**	0	**100**	0	**62**	33	**90**	10	**95**	5
	fold	1	**100**	0	0	0	0	**100**	0	**100**	**100**	0	**100**	0	0	0	0	**100**	0	**100**	0	0	0	**100**	0	**100**	0	0
≤150 AA	rosetta	12	**58**	0	**92**	0	**58**	0	50	0	17	0	33	0	**58**	0	50	0	**100**	0	25	0	42	0	**75**	0	**67**	0
	perturbation	17	**100**	0	**94**	0	**100**	0	29	0	**100**	0	**100**	0	**76**	0	**94**	0	**82**	0	47	0	41	0	41	0	**88**	0
	fold	9	**67**	0	44	0	22	0	22	0	**78**	0	**89**	0	22	0	22	0	11	0	0	0	22	0	22	0	22	0
>150 AA	rosetta	6	17	0	33	0	50	0	33	0	33	0	**67**	0	**67**	0	50	0	**100**	0	50	0	**83**	0	**83**	0	**67**	0
	perturbation	36	**100**	0	**100**	0	**94**	0	17	0	**92**	0	**97**	0	36	0	**97**	0	**92**	0	**61**	0	50	0	**69**	0	**72**	0
	fold	5	**60**	0	**80**	0	40	0	40	0	40	0	**60**	0	40	0	**60**	0	**60**	0	0	0	40	0	40	0	**80**	0
GDT_TS > 25%																																
all	rosetta	30	23	20	**53**	13	**70**	10	7	**73**	33	47	13	40	**67**	13	47	20	**83**	0	47	33	**63**	10	**80**	7	**80**	10
	perturbation	52	**71**	23	**75**	15	**94**	0	35	50	**87**	0	**79**	8	40	50	**62**	19	**98**	0	**60**	38	**71**	17	**87**	6	**94**	4
	fold	18	39	11	**61**	6	44	17	33	39	**61**	17	50	28	33	33	22	39	**56**	11	11	**72**	39	50	**56**	33	**83**	11
α-helical	rosetta	12	**58**	17	**83**	8	**58**	17	42	42	25	**75**	50	50	**58**	42	**67**	33	**100**	0	17	50	50	33	**67**	25	**58**	25
	perturbation	24	**100**	0	**96**	4	**92**	4	17	**79**	**92**	8	**100**	0	**58**	33	**92**	8	**75**	21	4	**83**	42	50	46	50	**63**	38
	fold	12	**100**	0	**100**	0	**100**	0	25	**75**	**100**	0	**100**	0	**83**	0	**83**	17	**92**	0	0	**75**	50	42	**58**	42	**100**	0
β-sheet	rosetta	3	0	0	**67**	33	33	33	**100**	0	0	**67**	33	**67**	33	33	33	33	**100**	0	**67**	33	33	33	**100**	0	**67**	33
	perturbation	8	**100**	0	**100**	0	**100**	0	38	50	**100**	0	**100**	0	50	38	**100**	0	**100**	0	**100**	0	25	**63**	25	**63**	**75**	25
	fold	5	**100**	0	**100**	0	**100**	0	40	**60**	**100**	0	**100**	0	**80**	0	**100**	0	**100**	0	**100**	0	20	**80**	20	**80**	**100**	0
α/β	rosetta	3	33	33	33	33	**67**	0	0	**67**	33	**67**	33	33	**100**	0	0	**100**	**100**	0	**67**	33	**100**	0	**100**	0	**100**	0

	perturbation	20	**100**	*0*	**100**	*0*	**100**	*0*	15	***80***	**100**	*0*	**100**	*0*	40	***60***	**100**	*0*	**100**	*0*	**100**	*0*	**60**	*35*	**90**	*10*	**95**	*5*
	fold	1	**100**	*0*	**100**	*0*	**100**	*0*	0	***100***	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*	**100**	*0*
≤150 AA	**rosetta**	12	**58**	*0*	**92**	*8*	**58**	*17*	50	*42*	17	***75***	33	***67***	**58**	*33*	50	*42*	**100**	*0*	25	***58***	42	*33*	**75**	*17*	**67**	*33*
	perturbation	17	**100**	*0*	**94**	*6*	**100**	*0*	29	***65***	**100**	*0*	**100**	*0*	**76**	*12*	**94**	*6*	**82**	*12*	47	*41*	41	***59***	41	***59***	**88**	*12*
	fold	11	**100**	*0*	**100**	*0*	**100**	*0*	27	***73***	**100**	*0*	**100**	*0*	**82**	*0*	**91**	*9*	**91**	*0*	45	*36*	45	***55***	36	***64***	**100**	*0*
>150 AA	**rosetta**	6	17	*50*	33	*33*	50	*17*	33	*33*	33	***67***	**67**	*17*	**67**	*33*	50	*50*	**100**	*0*	50	*17*	**83**	*17*	**83**	*17*	**67**	*0*
	perturbation	35	**100**	*0*	**100**	*0*	**94**	*3*	14	***80***	**94**	*6*	**100**	*0*	37	***60***	**97**	*3*	**91**	*9*	**60**	*37*	49	*40*	**69**	*26*	**71**	*29*
	fold	7	**100**	*0*	**100**	*0*	**100**	*0*	29	***71***	**100**	*0*	**100**	*0*	**86**	*0*	**86**	*14*	**100**	*0*	14	***71***	43	*43*	**71**	*29*	**100**	*0*

For each score and benchmark set, the percentage of protein model sets that had significant improvement in enrichment (Z-score > 1.0) and significant decline (Z-score < -1.0, second row in italic) are displayed. Two classifications for native-like 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. Proteins were also classified as small when having ≤ 150 amino acids. Cells with bold percentages highlight the cases where for more than 50% of the protein model sets a significant change in enrichment was achieved.

Enrichment can be achieved regardless of the sampling algorithm

Although ROSETTA generates low resolution models, they have complete chain and defined backbone conformation. 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 a partial arrangement of SSEs optimal. This yields not only 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 score 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 for GDT_TS and RMSD100 classification, the percentage drops 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 comparable over the benchmark set and shows that on optimal linear combination can overcome the weaknesses of the individual terms.

Methods and Materials

Divergent databank of high resolution crystal structures

Statistics have been derived from a divergent high resolution subset of the protein databank (PDB) which was generated using the protein sequence culling server “PISCES” [92]. With a sequence identity limit of 25%, resolutions up to 2.0 Å, a maximum R-value of 0.3, sequence lengths of 40 residues minimum only X-ray structures have been culled from the PDB. This guarantees that similar sequence are not over represented introducing a bias to proteins that are easier to experiment on or are of higher interest in the scientific fields. All membrane proteins have been excluded. The resulting databank has 4,379 chains in 3,409 PDB entries.

Secondary structure element packing

In order to determine the packing between two secondary structure elements, secondary structure elements have been read from their PDB-file. α-helices with a length <7 residues and β-strands <5 residues have been ignored, and α-helices or β-strands have been described as overlapping sets of fragments of the length of 5 residues for α-helices and 3 residues for β-strands (Figure 12A). 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 have been 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 12B). 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.

$$w_I = \frac{\cos 2\alpha_1 + 1}{2} \frac{\cos 2\alpha_2 + 1}{2} \tag{31}$$

The twist between the SSE fragments is defined by the dihedral angle θ between the SSE main axes (Figure 12C). 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 12D). 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:

$$w_O = \frac{\cos 2\omega + 1}{2} \tag{32}$$

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 omegas 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:

$$w_{sandwhich} = \frac{\cos 2\omega_1 + 1}{2}\frac{\cos 2\omega_2 + 1}{2} \tag{33}$$

$$w_{pairing} = \left(1 - \frac{\cos 2\omega_1 + 1}{2}\right)\left(1 - \frac{\cos 2\omega_2 + 1}{2}\right) \tag{34}$$

The actual packing between two SSEs is a list of fragment interactions (Figure 12E). 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: α-helical and β-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.

Three benchmark sets were created:

a) Using ROSETTA [23] 10,000 models have been folded *de novo* for each sequence. Since ROSETTA does not assign secondary structure, DSSP [93] was used to add definitions to the models.

b) 10,000 models each have been folded using the BCL::Fold program. For these simulations a scoring function with weights set to 1 was used. Further details on the folding simulations can be cleaned from Chapter IV.

c) Additionally, 12,000 perturbed structures have been generated using the BCL::Fold program by starting with the native SSE arrangement and applying randomly the following perturbations to the starting structure: (1) SSE rotation and translation; (2) SSE flip; (3) swapping two SSEs and (4) SSE removal.

Native-like models or postives were defined using two quality metrics: RMSD100 cutoff of 8Å to as well as a GDT_TS cutoff of 25%. The remaining models in each set were considered negatives or non-native-like. If there were less than 1% or more than 99% native-like models, that set was ignored for further analysis, since it indicates that the sampling algorithm is not suitable for that protein's structure, either creating too many or two few native-like models. The ratio native-like/non-native-like is dependent on the performance of each protocol. As this ratio also determines maximum enrichment we compensate by creating 10 sets with 10% native-like models each. Models were randomly selected from the set that is underrepresented in the native-like/non-native-like ratio. These models were added to overrepresented classified models. Enrichments were calculated over all 10 sets and a mean and standard deviation is reported in Table 10. The sum was calculated as a linear combination of the potentials with a weight set:

AA distance	AA neighbor	loop length	Radius of gyration	SSE clash	SSE packing	Strand pairing	Contact Score
0.35	50	10	5	500	8	20	0.5

CHAPTER IV

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

This chapter is a preproduction of the similarly titled co-first-author manuscript which will be submitted to “PLoS Computational Biology” co-authored by Mert Karakaş*, Rene Staritzbichler, Nathan Alexander and Jens Meiler.

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 [5], [94], [95]. 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 gains importance [96].

Despite impressive advancements in the combination of experimental protein structure determination techniques [97], [98] with comparative modeling [99], entire classes of proteins remain underrepresented in the PDB as they evade crystallization or are unsuitable for NMR studies; e.g. membrane proteins [100] and proteins that only fold as part of a large macromolecular assembly [69], [101]. 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 [102]. 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 [16], [103-105] and other properties of a given sequence such as β-hairpins [106], disorder [107], [108], non-local contacts [109], domain boundaries [110-112], and domain interactions [113], [114]. 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 [18], [19].

This preparatory step is followed by the actual folding simulation. Rosetta, one of the best performing *de novo* methods, follows a fragment assembly approach [20], [31], [115]. 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 [24].

Chunk-Tasser is another fragment assembly method for *de novo* structure prediction that was one of the top groups in CASP8 [116]. 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 [117].

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. In such cases, X-Ray crystallography and cryo-Electron Microscopy 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 [59], [118], [119]. NMR and EPR spectroscopy yield sparse datasets due to technological or resource limitations [91], [120]. 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 [121]. *De novo* methods have also been applied for rapid fold determination from unassigned NMR data [27] and structure determination for larger proteins from NMR restraints [122]. In addition, *de novo* structure prediction have also been coupled with EPR restraints [25] as well as cryoEM [2].

Objective of the present work is to introduce an algorithm for protein folding with a novel approach of assembly of secondary structure elements 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 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.

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

During the folding simulation, Rosetta and most *de novo* methods use a reduced protein representation that excludes side chain degrees of freedom to simplify the conformational search space and complexity of the energy 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 [123-125]. At this stage, the backbone of flexible loop regions can be further refined, in Rosetta by a combination of fragment insertions and gradient minimization. In the CASP6 experiment, Rosetta was able to predict *de novo* the structure of a small α-helical protein to a resolution of 1.59Å [115]. 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 [8]. Note that the refinement of backbone conformations and construction of side chain coordinates aligns with most comparative modeling protocols [58]. These algorithms model gaps and insertions using loop closure algorithms that use analytical geometry [80], molecular mechanics [126], or loop libraries from the PDB [79] 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.

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 to sample increases rapidly as the protein gets larger. Despite simplified representation of proteins using just backbone atoms, 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 more 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 acids require coordinated adjustment of multiple phi, psi angles or will 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 ($H_{\alpha 2}$ for Glycine) within 8Å [127], [128]. As the RCO increases above 0.25, the success rate of *de novo* prediction drops drastically [129]. 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 before *de novo* methods can be successfully applied to larger proteins. Interestingly, contact order correlates also with protein folding rates suggesting that the sampling of non-local contacts is the rate-limiting step in protein folding [86].

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. In such cases, X-Ray crystallography and cryo-Electron Microscopy 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 [59], [118], [119]. NMR and EPR spectroscopy yield sparse datasets due to technological or resource limitations [91], [120]. 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 [121]. *De novo* methods have also been applied for rapid fold determination from unassigned NMR data [27] and structure determination for larger proteins from NMR restraints [122]. In addition, *de novo* structure prediction have also been coupled with EPR restraints [25] as well as cryoEM [2].

Objective of the present work is to introduce an algorithm for protein folding with a novel approach of assembly of secondary structure elements 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 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.

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 [129]. 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 pairings is often sampled insufficiently frequent 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.

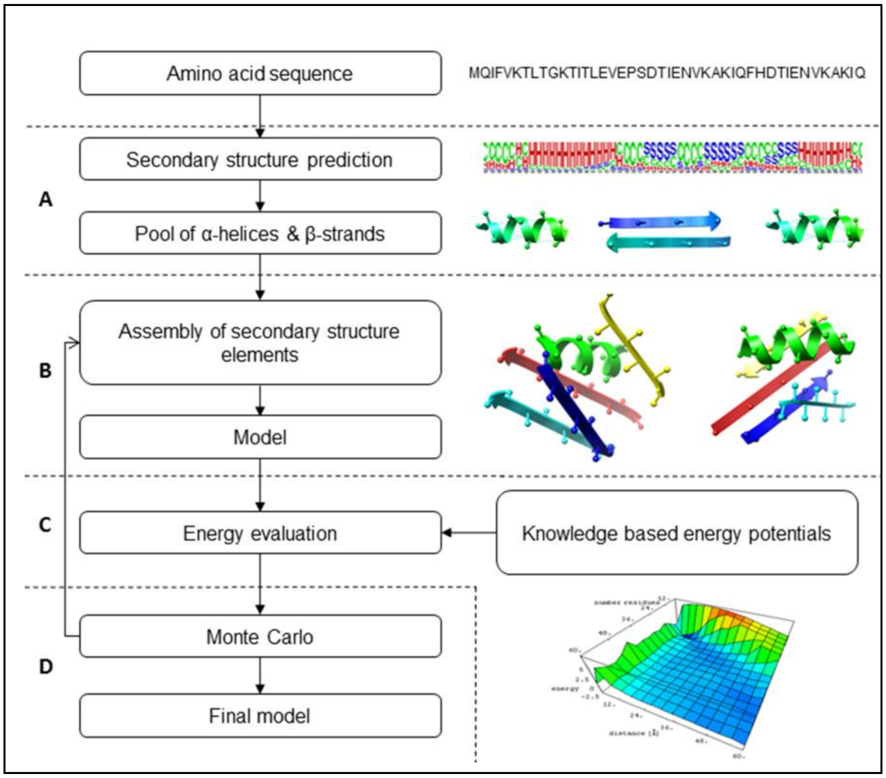

Amino acid sequence
MQIFVKTLTGKTITLEVEPSDTIENVKAKIQFHDTIENVKAKIQ
A
Secondary structure prediction
Pool of α-helices & β-strands
B
Assembly of secondary structure elements
Model
C
Energy evaluation
Knowledge based energy potentials
D
Monte Carlo
Final model
number residues
energy
distance (Å)

Figure 20 BCL::Fold protocol

(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 20B). Individual residues are represented by their backbone and Cβ atoms only ($H_{\alpha2}$ 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 11) and evaluated by knowledge-based energy potentials (Table 12). 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.

BCL::Fold was evaluated using a benchmark set of proteins collected using PISCES culling server. The set includes 64 proteins of lengths ranging from 83 to 293 residues with <30% sequence similarity. The set contains different topologies including 29 all α-helical, 16 all β-strand, and 19 mixed αβ folds (Table 13). The selected proteins have RCOs in the range of 0.13 to 0.46 with an average of 0.29 ± 0.07. It should be noted that as proteins get larger, RCO values start decreasing (compare Figure 21**)**.

Table 11 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 12 Weight set for the energy function in BCL::Fold

energy function	weight
aa_clash	500.0
aa_dist	0.3
aa_neigh	83.0
sse_clash	500.0
sse_pack	5.0
strand_pair	36.0
loop	14.5
loop_closure	500.0
rgyr	12.5
co	2.5
sse_prediction_JUFO*	1.0
sse_prediction_PSIPRED*	1.0
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 (aa_neigh), 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 definition agreement score using secondary structure predictions from JUFO (sse_prediction_JUFO) and PSIPRED (sse_prediction_PSIPRED), entropy score (entropy).

* sse_prediction_JUFO and sse_prediction_PSIPRED scores were not used for BCL::Fold benchmark runs that used native secondary structure definitions.

Table 13 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
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
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	58	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 64 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.

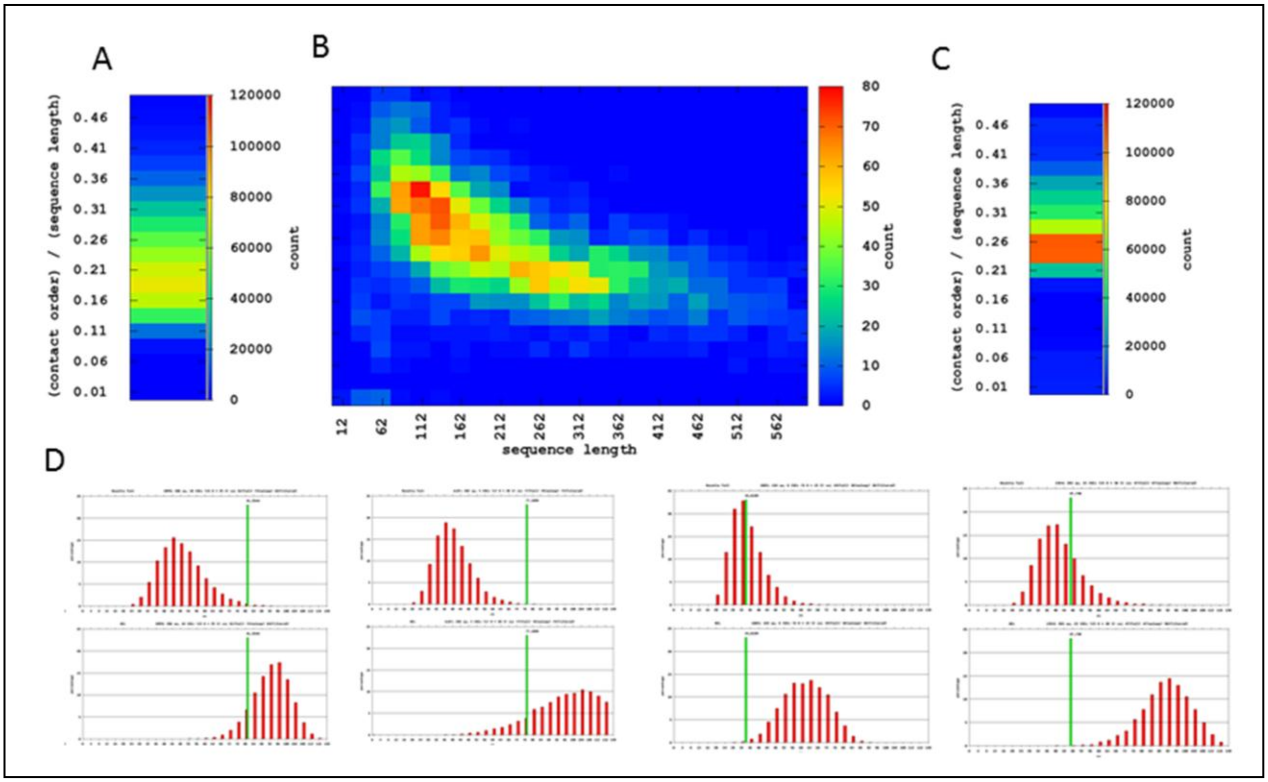


Figure 21 Contact order distributions for BCL before contact order score

Panels A-C show RCO distribution histograms with use of heat maps, for **(A)** Rosetta generated models for a benchmark of 54 proteins **(B)** Pisces culled non-redundant protein set, proteins are distributed along the x axis by sequence length. **(C)** Heat map for BCL::Fold generated models for the same benchmark set used in (A). **(D)** Representative set of RCO distribution histograms for Rosetta (top row) and BCL (bottom row). Native contact order values are indicated with the green bar.

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

The secondary structure prediction programs JUFO [14] and PSIPRED [15] 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 [104]. 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 separate MCM minimization is run to create a pool of more likely SSEs. The scoring 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 and b) the limited contribution to fold stability expected from short SSEs.

Table 14 Secondary structure pool statistics for the benchmark proteins

pdb id	Pool agreement score				Q3			
	H_{JUFO}	MC_{JUFO}	$H_{PSIPRED}$	$MC_{PSIPRED}$	H_{JUFO}	MC_{JUFO}	$H_{PSIPRED}$	$MC_{PSIPRED}$
1EYHA	47.62	47.85	28.88	28.30	71.79	71.55	87.72	88.60
1FQIA	36.79	34.01	21.67	20.28	73.79	74.51	82.18	83.00
1GAKA	48.01	42.04	44.33	41.80	75.24	75.96	87.50	85.58
1GYUA	31.90	24.59	11.33	10.75	71.01	67.65	86.76	88.41
1IAPA	52.84	50.75	43.11	42.87	78.83	76.69	80.88	81.48
1ICXA	28.07	31.08	22.38	22.38	81.25	73.45	83.04	83.04
1J27A	27.13	22.54	1.39	1.39	72.84	76.25	96.15	96.15
1JL1A	53.94	52.67	40.71	39.32	66.67	64.55	76.70	75.96
1LMIA	48.24	46.31	38.13	37.19	43.66	45.83	54.55	56.06
1OXJA	30.03	41.40	36.29	36.69	78.74	76.56	84.30	83.61
1OZ9A	35.58	41.05	11.63	11.63	78.85	69.16	90.91	90.91
1PBVA	21.48	17.08	12.71	12.71	88.64	90.08	93.89	93.89
1PKOA	34.88	25.60	20.09	20.09	63.29	71.05	77.14	77.46
1Q5ZA	41.58	39.37	20.86	19.47	64.13	59.77	75.53	76.92
1RJ1A	44.06	43.85	28.79	26.83	85.83	85.71	90.24	90.16
1T3YA	33.36	32.07	28.42	25.20	75.82	70.33	73.03	75.28
1TP6A	53.44	45.80	35.14	29.09	51.49	56.44	73.47	74.49
1TQGA	12.45	12.95	4.16	4.16	87.78	86.67	96.63	96.63
1TZVA	45.25	45.25	34.44	36.40	79.05	79.05	86.67	85.85
1UAIA	46.00	38.02	43.62	49.46	66.40	70.40	68.85	68.85
1ULRA	19.64	20.90	8.55	9.94	70.59	69.57	90.16	88.52
1VINA	53.14	57.45	33.21	39.20	78.41	74.14	83.52	80.75
1X91A	31.63	29.46	14.33	14.33	80.17	82.61	89.43	89.43
1XAKA	33.72	34.53	38.61	24.66	21.74	25.53	36.00	43.18
1XKRA	34.51	42.74	30.62	26.08	80.00	79.61	89.33	87.25
1XQOA	74.65	77.58	79.73	70.53	67.05	60.89	74.47	71.12
1Z3XA	59.20	64.38	30.45	30.45	75.69	78.17	82.64	82.64
2AP3A	72.96	65.29	28.92	29.30	76.88	75.16	84.18	83.71
2BK8A	13.29	16.06	4.97	4.97	71.19	71.67	94.00	94.00
2CWRA	18.19	19.00	25.33	25.23	75.81	74.19	79.03	80.65
2EJXA	78.32	68.78	40.88	36.95	50.45	52.68	71.43	70.54
2F1SA	41.02	51.54	28.03	26.64	75.00	74.81	84.13	84.80
2FC3A	41.76	39.56	20.19	18.80	70.00	70.00	84.44	85.39
2FM9A	27.93	30.31	46.14	45.78	84.62	84.52	84.97	84.39
2FRGP	28.87	25.86	25.75	25.75	68.83	71.05	68.12	68.12
2GKGA	21.68	23.06	8.76	14.98	76.47	75.58	92.50	86.42
2HUJA	29.34	37.01	6.93	6.36	84.85	83.00	95.15	95.15
2IU1A	67.87	69.52	39.78	39.78	76.43	75.71	81.43	81.43
2JLIA	34.52	36.72	23.92	26.12	65.93	65.93	63.33	63.74
2LISA	36.94	40.01	17.32	17.32	80.65	82.80	88.30	88.30
2OF3A	77.35	80.24	69.61	71.97	78.87	78.30	86.43	87.27
2OSAA	53.57	53.22	42.82	26.40	69.57	65.94	78.79	80.92
2QZQA	47.78	48.94	55.33	40.65	43.08	40.30	49.38	56.34
2R0SA	50.66	55.08	59.56	53.63	64.16	62.72	68.91	65.57
2RB8A	6.93	6.93	9.13	6.93	80.39	80.77	80.77	82.69
2RCIA	84.05	74.19	82.75	73.79	55.00	54.86	59.48	60.93
2V75A	27.16	29.09	24.57	24.57	69.86	70.83	75.64	75.64
2VQ4A	17.68	17.19	19.62	25.22	69.35	70.77	75.38	72.06
2WJ5A	13.73	18.21	8.99	8.99	71.43	70.83	85.11	85.11
2WWEA	23.29	21.67	25.17	26.43	70.83	69.74	79.49	78.48
2YV8A	31.38	29.18	9.53	13.69	72.94	70.93	82.14	78.57
2YXFA	34.53	34.53	19.28	17.32	52.46	52.46	71.15	72.55
2YYOA	43.80	37.68	33.15	33.15	63.44	68.48	68.29	68.29

2ZCOA	90.31	101.98	76.24	77.63	79.66	77.22	82.67	82.30
3B5OA	83.24	80.93	81.40	69.62	59.70	61.22	73.21	74.06
3CTGA	33.42	33.05	17.76	17.76	66.67	69.05	82.89	82.89
3CX2A	20.76	15.34	21.12	16.97	67.19	73.44	75.38	79.03
3FH2A	18.88	20.26	5.55	18.95	90.38	89.52	96.04	95.10
3FHFA	84.71	90.50	70.31	66.15	61.82	60.37	73.65	73.65
3FRRA	34.95	36.57	26.03	28.43	86.99	86.21	93.01	91.61
3HVWA	67.32	61.99	55.41	55.41	53.24	59.12	60.87	60.00
3IV4A	22.50	23.88	21.56	21.56	82.05	81.01	82.89	82.89
3NE3B	39.43	31.44	24.91	24.91	68.75	76.67	79.35	79.35
3OIZA	28.47	29.86	36.65	35.84	64.94	65.38	66.67	67.57
average	41.26	41.05	30.80	29.67	70.85	70.79	79.74	79.80
stdev	19.86	20.08	19.87	18.06	12.13	11.48	11.49	10.62

The table depicts pool agreement score and Q3 score for the pools generated using secondary structure prediction methods Jufo and PSIPRED for all of the 64 proteins in the benchmark set. H_{JUFO} and $H_{PSIPRED}$ refer to the pools generated by simply using the highest probability for each residue for secondary structure assignment, while MC_{JUFO} and $MC_{PSIPRED}$ refer to pools that were generated using Monte Carlo based minimization on the previous pools. The last three rows show the average and the standard deviation for pool agreement score and Q3 measure.

Table 14 depicts Q3 [130] accuracies and the BCL::SSE pool agreement scores (see Methods) for the SSE pools of the 64 benchmark proteins using PSIPRED and JUFO secondary structure prediction. BCL::SSE generated SSE pools exhibit Q3 values comparable to the highest probability assignments with 80% and 71% accuracy respectively for PSIPRED and JUFO. The BCL::SSE pool agreement scores decreased from 41.26 to 41.05 for Jufo and 30.80 to 29.67 for PSIPRED. BCL::SSE is a separate application executed prior to BCL::Fold. Thereby secondary structure prediction methods used can be adjusted by the user. Further the user can manually define SSEs he wants considered by BCL::Fold.

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-strand 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. Moves enabled only in this phase include SSE bending or small rotations and translations. Currently both stages utilize the same energy function (compare Chapter III).

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 or 400 consecutive steps that did not improve the score. In general a move can result in one of four outcomes: “improved” in score, “accepted” through Metropolis criterion, “rejected” as score worsened, or “skipped” as SSE elements required for move are not present in the model.

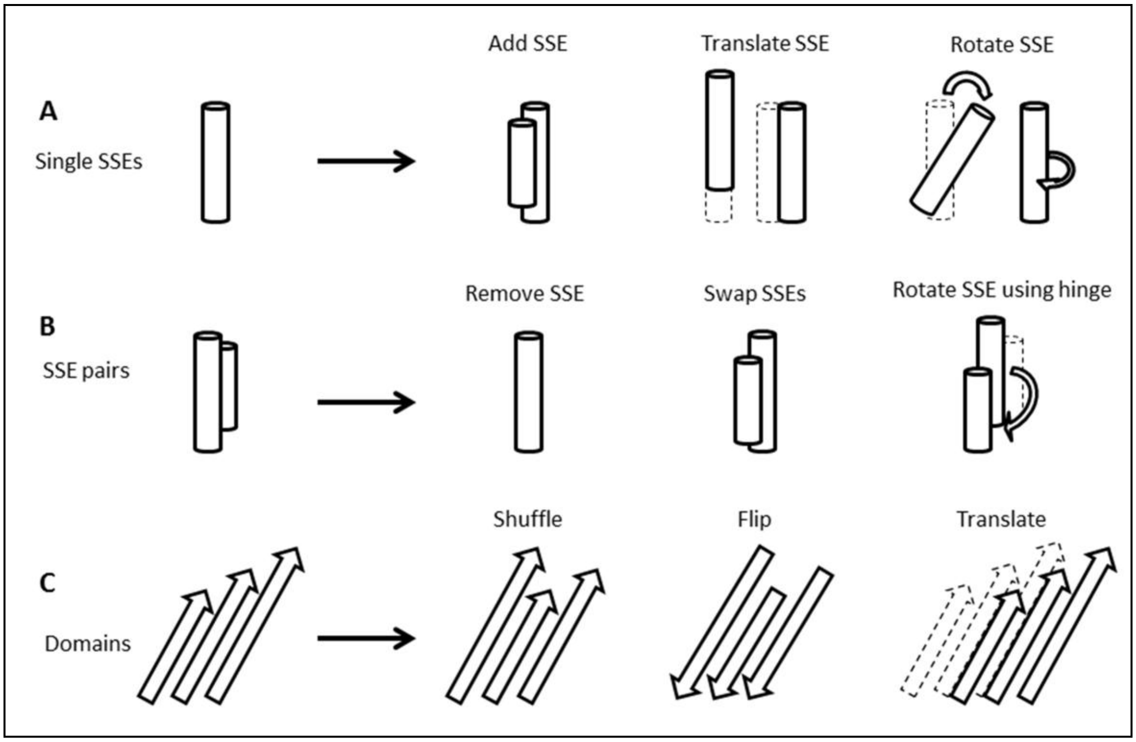


Figure 22 SSE-based moves allow rapid sampling in conformational search space

The types 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.

A comprehensive list of all moves used in BCL::Fold is given in Table 11 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 22. 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 107 moves, 74 are used in assembly and 34 are used in refinement. Resizing SSEs ("sse_resize") is the only move used in both stages. Table 15 also 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 15).

Table 15 Statistics for the moves used in BCL::Fold protocol

Move	Type	Stage	$\%_{\text{improved}}$	$\%_{\text{accepted}}$	$\%_{\text{rejected}}$	$\%_{\text{skipped}}$	Δ_{mean}
add_sse_next_to_sse	add	A	1.7	4.3	8.8	85.2	-392.8
add_sse_short_loop	add	A	2.4	4.5	7.1	85.9	-401.8
add_strand_next_to_sheet	add	A	2.0	2.0	2.0	94.0	-458.4
remove_random	remove	A	0.2	16.5	82.8	0.5	-236.9
remove_unpaired_strand	remove	A	0.1	6.9	11.4	81.6	-220.6
swap_sse_with_pool	swap	A	1.0	4.3	5.1	89.6	-241.8
swap_sse_with_pool_overlap	swap	A	4.0	37.1	58.0	0.9	-126.5
swap_sses	swap	A	0.8	18.3	78.6	2.3	-208.5
sse_bend_ramachandran	SSE	R	7.8	19.4	72.8	0.0	-21.8
sse_bend_random_large	SSE	R	7.8	23.0	69.2	0.0	-27.7
sse_bend_random_small	SSE	R	20.3	36.9	42.8	0.0	-20.1
sse_furthest_move_next	SSE	A	1.1	15.0	84.0	0.0	-289.2
sse_move_next	SSE	A	0.5	11.0	88.5	0.0	-264.6
sse_move_short_loop	SSE	A	0.8	11.5	76.1	11.7	-276.9
sse_resize	SSE	A + R	14.7	28.2	45.2	11.9	-106.5
sse_rotate_large	SSE	A	1.4	20.2	78.5	0.0	-98.2
sse_rotate_x_large	SSE	A	2.4	23.1	74.4	0.0	-79.7
sse_rotate_y_large	SSE	A	4.0	28.5	67.5	0.0	-126.5
sse_rotate_z_large	SSE	A	9.1	47.3	43.6	0.0	-40.1
sse_rotate_small	SSE	R	3.3	17.0	79.7	0.0	-33.2
sse_rotate_x_small	SSE	R	7.5	23.5	69.0	0.0	-20.9
sse_rotate_y_small	SSE	R	10.2	26.6	63.1	0.0	-27.4
sse_rotate_z_small	SSE	R	17.9	42.3	39.8	0.0	-11.2
sse_split_JUFO	SSE	A	1.8	24.2	69.3	4.7	-88.8
sse_split_PSIPRED	SSE	A	2.1	24.6	68.5	4.8	-84.7
sse_translate_large	SSE	A	0.6	16.6	82.7	0.0	-148.3
sse_translate_x_large	SSE	A	2.1	27.1	70.9	0.0	-106.5
sse_translate_y_large	SSE	A	1.7	21.5	76.8	0.0	-110.6
sse_translate_z_large	SSE	A	7.4	45.6	47.0	0.0	-59.3
sse_transform_large	SSE	A	0.4	14.1	85.5	0.0	-136.6
sse_translate_small	SSE	R	3.0	18.1	78.9	0.0	-50.0
sse_translate_x_small	SSE	R	12.7	31.9	55.4	0.0	-18.1
sse_translate_y_small	SSE	R	9.2	27.0	63.8	0.0	-30.0
sse_translate_z_small	SSE	R	15.0	41.8	43.2	0.0	-7.6
sse_transform_small	SSE	R	1.1	11.8	87.1	0.0	-45.2
helix_flip_xy	α-helix	A	2.8	32.9	64.0	0.3	-132.1
helix_flip_z	α-helix	A	3.7	40.8	55.2	0.3	-109.6
helix_furthest_move_next	α-helix	A	1.1	15.5	83.2	0.3	-295.1
helix_move_next	α-helix	A	0.6	12.6	86.6	0.3	-274.8
helix_move_short_loop	α-helix	A	0.9	13.4	73.4	12.3	-278.5
helix_translate_xy_large	α-helix	A	1.6	26.7	71.4	0.3	-128.3
helix_translate_z_large	α-helix	A	8.4	46.9	44.5	0.3	-59.7
helix_rotate_xy_large	α-helix	A	2.0	26.4	71.3	0.3	-91.1
helix_rotate_z_large	α-helix	A	13.7	53.1	33.0	0.3	-40.9
helix_transform_xy_large	α-helix	A	0.9	21.0	77.8	0.3	-123.6
helix_transform_z_large	α-helix	A	4.5	38.4	56.9	0.3	-88.3
helix_translate_xy_small	α-helix	R	4.8	30.3	64.8	0.1	-17.5
helix_translate_z_small	α-helix	R	16.1	46.6	37.2	0.1	-8.0
helix_rotate_xy_small	α-helix	R	5.0	26.2	68.8	0.1	-23.1
helix_rotate_z_small	α-helix	R	18.4	51.0	30.5	0.1	-7.0
helix_transform_xy_small	α-helix	R	2.3	20.3	77.3	0.1	-31.7
helix_transform_z_small	α-helix	R	9.4	40.4	50.1	0.1	-12.4
strand_flip_x	β-strand	A	1.6	26.6	69.8	1.9	-180.7
strand_flip_y	β-strand	A	1.5	26.1	70.5	2.0	-188.0
strand_flip_z	β-strand	A	8.8	53.7	35.5	2.0	-34.2
strand_furthest_move_next	β-strand	A	0.7	11.7	85.7	1.9	-237.3
strand_furthest_move_sheet	β-strand	A	1.5	17.6	66.6	14.3	-310.5
strand_move_next	β-strand	A	0.4	8.7	89.0	1.9	-232.0
strand_move_sheet	β-strand	A	0.7	12.7	72.3	14.3	-257.5
strand_translate_z_large	β-strand	A	9.7	47.4	41.0	2.0	-48.7
strand_translate_z_small	β-strand	R	13.1	35.1	50.7	1.1	-7.8
ssepair_translate_large	SSE pair	A	1.1	12.3	25.7	60.9	-101.6
ssepair_translate_no_hinge_large	SSE pair	A	0.3	7.2	31.7	60.8	-155.5
ssepair_rotate_large	SSE pair	A	1.3	10.3	27.4	61.0	-91.3
ssepair_transform_large	SSE pair	A	0.4	7.8	30.8	61.0	-132.5

ssepair_translate_small	SSE pair	R	4.3	14.6	22.9	58.2	-19.0
ssepair_translate_no_hinge_small	SSE pair	R	0.9	7.7	33.3	58.1	-33.9
ssepair_rotate_small	SSE pair	R	2.9	10.7	28.2	58.1	-17.2
ssepair_transform_small	SSE pair	R	1.8	10.2	29.9	58.2	-24.5
helixpair_rotate_z_large_hinge	α-pair	A	1.1	19.6	66.4	12.9	-146.7
helixpair_rotate_z_large_no_hinge	α-pair	A	1.2	19.9	66.0	12.9	-143.4
helixpair_rotate_z_small_hinge	α-pair	R	4.2	26.0	60.4	9.4	-11.8
helixpair_rotate_z_small_no_hinge	α-pair	R	4.5	26.3	59.7	9.4	-11.5
helixdomain_flip_ext	α-domain	A	0.1	3.8	18.9	77.2	-192.2
helixdomain_flip_int	α-domain	A	0.2	5.6	16.7	77.5	-137.4
helixdomain_shuffle	α-domain	A	0.4	16.4	82.0	1.2	-259.2
helixdomain_translate_large	α-domain	A	0.3	13.5	85.1	1.2	-186.5
helixdomain_rotate_large	α-domain	A	0.2	9.9	88.8	1.1	-140.0
helixdomain_transform_large	α-domain	A	0.1	8.6	90.1	1.2	-156.3
helixdomain_translate_small	α-domain	R	1.0	17.6	81.4	0.1	-30.9
helixdomain_rotate_small	α-domain	R	0.5	9.2	90.3	0.1	-37.2
helixdomain_transform_small	α-domain	R	0.0	3.2	96.7	0.1	-59.1
sheet_shuffle	β-sheet	A	1.0	17.1	75.8	6.1	-192.6
sheet_switch_strand	β-sheet	A	0.9	7.5	27.4	64.1	-380.8
sheet_cycle	β-sheet	A	0.5	12.3	68.5	18.7	-256.5
sheet_cycle_intact	β-sheet	A	0.5	11.8	69.2	18.5	-225.1
sheet_cycle_subset	β-sheet	A	0.7	28.3	52.6	18.4	-182.8
sheet_cycle_subset_intact	β-sheet	A	0.7	27.8	52.7	18.7	-175.2
sheet_divide	β-sheet	A	0.7	8.6	54.5	36.2	-154.3
sheet_divide_sandwich	β-sheet	A	0.2	3.3	60.1	36.5	-371.3
sheet_flip_ext	β-sheet	A	0.7	41.6	51.7	6.1	-147.1
sheet_flip_int	β-sheet	A	1.4	24.5	67.9	6.2	-102.4
sheet_flip_int_sub	β-sheet	A	2.1	25.4	66.4	6.2	-90.1
sheet_flip_int_sub_diff	β-sheet	A	1.4	20.0	72.6	6.1	-128.5
sheet_pair_strands	β-sheet	A	0.8	2.7	4.6	91.9	-457.8
sheet_register_fix	β-sheet	R	1.0	13.0	66.6	19.4	-23.2
sheet_register_shift	β-sheet	A	1.7	25.7	53.9	18.7	-83.7
sheet_register_shift_flip	β-sheet	A	3.4	33.6	44.4	18.5	-71.4
sheet_translate_large	β-sheet	A	1.0	42.7	55.9	0.5	-139.4
sheet_rotate_large	β-sheet	A	0.6	38.5	60.4	0.5	-99.9
sheet_transform_large	β-sheet	A	0.4	37.7	61.4	0.6	-109.1
sheet_twist_large	β-sheet	A	7.5	26.9	47.0	18.6	-128.7
sheet_translate_small	β-sheet	R	1.6	43.9	54.4	0.1	-33.5
sheet_rotate_small	β-sheet	R	0.9	38.3	60.7	0.1	-53.2
sheet_transform_small	β-sheet	R	0.4	36.2	63.4	0.1	-71.9
sheet_twist_small	β-sheet	R	10.4	21.9	48.3	19.4	-27.8
total	TOTAL		2.7	19.6	59.1	18.6	-73.7

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(PI), accepted steps (PA), rejected steps (PR), skipped steps(PS). This is followed by ΔMEAN, 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.

The five individual moves with largest score improvements are mostly add and strand moves, 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 dangerous 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.

BCL::Fold samples native-like topologies for 72% 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 [93]. 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 (see Methods) [80]. However, in the present analysis we focus on placement of SSEs to form the topology. The average and standard deviations of RMSD100 [131] and GDT [11] values of the best models generated by these runs can be found in Table 16**.** RMSD100 and GDT measurements were calculated using Cα atoms of all amino acids in the model, which is lower for SSE-only models. BCL::Fold using the correct secondary structure RMSD100-values of 5.4 ± 1.5Å (SSE only models) and 6.9 ± 1.6Å (complete models) were achieved. For simulations with predicted SSEs RMSD100 values of 6.1 ± 1.5Å (SSE only models) and 7.0 ± 1.7Å (complete models) were obtained. For comparison, ROSETTA [23] generated models with RMSD100-values of 6.3 ± 2.1Å.

BCL::Fold improved the RMSD100 when compared with Rosetta in 20 cases (31%) with correct SSE definitions and in 17 cases (27%) using an predicted SSE pool.

When GDT_TS values are compared, Rosetta has a significant advantage over BCL. This is expected due to the nature of GDT_TS measure accompanied with the differences between the methods. Rosetta utilizes local sequence bias in its sampling including fragments not only for SSEs but also for loop regions and thus allowing better superimposition of super secondary structures. In BCL::Fold, any extensive backbone sampling for SSEs are currently not implemented, so even when two SSEs are packed correctly, frequently the curvature of the SSEs are not correctly captured. This issue, accompanied with Rosetta's successful backbone fragment replacement strategy allows Rosetta to produce model with significantly higher GDT_TS, especially for 1Å and 2Å cutoffs.

Table 16 Best RMSD100 and GDT_TS values for models generated by BCL and Rosetta

	RMSD100					**GDT_TS**				
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**
1EYHA	4.74	5.74	6.44	7.21	4.30	42.53	46.88	41.15	41.32	60.24
1FQIA	7.32	8.57	8.06	8.80	5.22	29.42	34.01	38.95	39.97	50.17
1GAKA	5.85	7.62	6.45	6.80	4.55	40.07	45.57	40.43	44.15	60.64
1GYUA	3.76	6.10	3.74	4.29	5.56	27.14	37.68	35.89	42.32	44.29
1IAPA	7.01	8.64	6.97	8.01	5.43	23.58	25.47	25.59	27.61	38.39
1ICXA	4.39	5.37	5.43	6.04	5.59	32.26	40.97	40.65	40.65	46.13
1J27A	3.97	4.57	4.24	4.53	4.40	48.04	57.11	50.49	54.17	62.25
1JL1A	6.81	8.53	6.59	7.49	8.19	28.55	34.19	34.84	38.06	39.84
1LMIA	5.70	7.87	6.95	9.05	9.49	24.24	32.82	22.52	28.44	33.21
1OXJA	6.06	8.06	6.46	6.83	6.70	31.79	35.26	33.81	34.39	48.99
1OZ9A	5.41	6.40	5.20	6.40	5.25	33.67	39.50	35.50	40.83	51.17
1PBVA	7.95	8.98	7.61	7.99	6.47	24.87	30.90	29.23	31.54	51.15
1PKOA	5.94	7.60	7.84	8.40	8.01	22.12	31.83	26.08	31.83	41.73
1Q5ZA	3.83	7.20	6.00	7.22	8.23	22.18	31.78	26.41	29.52	35.73
1RJ1A	4.44	5.34	4.25	4.34	3.30	47.85	56.29	56.79	60.43	72.02
1T3YA	4.93	5.60	5.16	5.78	6.07	30.32	37.77	34.93	39.54	45.04
1TP6A	4.24	4.78	5.87	6.22	5.21	36.13	43.75	34.96	42.97	50.98
1TQGA	1.54	2.55	1.98	2.23	1.41	73.81	77.38	74.52	78.10	96.67
1TZVA	4.43	5.77	4.60	5.12	3.19	39.61	51.41	44.01	46.83	63.03
1UAIA	6.39	8.94	6.99	9.00	9.61	17.86	24.44	18.19	22.77	27.57
1ULRA	3.69	5.64	4.34	5.03	4.16	50	65.06	58.24	66.76	78.12
1VINA	7.99	8.86	7.91	8.57	8.31	20.24	23.41	21.74	24.72	28.36
1X91A	2.47	3.99	4.18	4.41	2.46	61.44	67.65	61.11	66.34	77.78
1XAKA	6.03	6.28	5.30	8.36	7.72	27.11	39.16	31.93	33.13	43.07
1XKRA	6.48	7.74	7.79	8.54	8.78	26.58	30.46	28.76	30.58	34.83
1XQOA	7.94	9.37	8.19	9.42	9.13	19.04	22.46	21.09	22.85	26.37
1Z3XA	7.28	9.49	7.57	9.16	8.41	20.38	24.05	21.85	25.74	29.41
2AP3A	3.75	5.28	5.62	5.95	4.11	53.77	55.03	53.77	56.66	61.68
2BK8A	5.24	7.74	6.99	7.51	4.27	31.19	46.13	39.95	48.45	76.03
2CWRA	4.85	5.41	6.12	7.17	7.24	32.77	43.93	38.59	41.50	40.53
2EJXA	5.14	5.79	6.62	7.35	5.09	39.57	46.58	36.69	39.93	51.8
2F1SA	7.24	7.57	7.03	7.87	7.20	25.4	27.02	26.88	27.55	37.5
2FC3A	4.93	7.78	5.94	7.91	5.75	33.06	39.92	42.94	45.97	48.59
2FM9A	6.30	6.82	6.14	6.51	6.22	33.26	34.42	35.93	38.49	42.09
2FRGP	4.53	5.48	6.54	6.91	6.53	35.14	47.41	35.14	42.92	51.18
2GKGA	3.02	4.31	3.20	4.86	3.39	43.7	52.17	45.87	52.17	63.78
2HUJA	2.35	3.47	2.60	2.74	3.47	52.86	57.68	59.11	61.43	57.5
2IU1A	6.45	7.46	6.01	7.55	6.76	27.76	31.25	29.21	29.33	38.7
2JLIA	5.30	6.60	6.13	6.69	5.86	33.74	39.84	32.32	35.77	43.5
2LISA	6.01	7.01	6.91	7.24	5.61	38.79	45.04	45.40	48.53	59.38
2OF3A	8.55	8.89	8.72	9.42	8.30	24.91	27.91	21.43	24.34	33.18
2OSAA	6.36	7.39	6.83	7.72	7.96	24.63	29.21	24.50	28.71	38.99
2QZQA	5.15	6.88	5.68	8.04	9.89	23.03	34.05	21.05	28.29	25.82
2R0SA	6.19	9.72	7.19	10.12	10.27	19.82	21.14	20.18	21.23	25.09
2RB8A	3.72	5.17	4.02	4.78	4.64	29.57	48.32	40.87	52.16	58.89
2RCIA	5.44	6.98	9.07	10.64	9.98	27.7	31.50	20.47	22.67	26.35
2V75A	3.79	4.42	3.33	3.50	2.11	46.88	54.09	52.16	55.53	74.28
2VQ4A	4.82	6.94	6.31	7.28	9.18	27.83	42.92	35.85	41.98	43.16
2WJ5A	5.55	9.44	7.31	8.21	7.66	29.21	39.11	39.36	45.54	65.84
2WWEA	4.55	6.85	4.91	6.26	5.61	29.13	40.16	37.20	41.73	50.2
2YV8A	5.82	7.51	5.17	7.49	8.25	24.85	34.45	26.68	31.40	34.15
2YXFA	5.42	7.31	5.57	6.32	4.36	29.5	41.75	37.25	43.00	57.5
2YYOA	5.75	8.71	6.48	7.46	8.89	17.84	21.64	23.68	25.29	34.21
2ZCOA	7.60	8.41	7.65	8.32	8.12	19.45	21.50	22.95	25.34	27.47
3B5OA	5.96	7.01	8.66	9.15	8.92	31.56	35.96	22.64	24.59	28.89
3CTGA	5.17	6.85	5.85	6.42	3.75	28.68	37.60	32.75	36.82	50.78
3CX2A	5.43	7.63	7.32	7.77	8.16	30.56	43.52	37.73	43.75	57.18
3FH2A	6.44	7.40	6.82	7.53	4.73	32.19	36.64	34.08	34.93	53.08
3FHFA	7.83	8.76	7.90	8.72	7.54	21.96	26.99	26.40	30.02	38.08
3FRRA	6.48	7.67	6.45	7.59	5.41	40.84	45.16	42.41	44.76	59.69
3HVWA	6.99	8.77	6.14	6.30	6.43	25.14	27.41	34.09	36.51	46.31
3IV4A	4.20	4.66	4.83	5.64	3.98	39.29	51.79	40.85	47.77	64.73
3NE3B	4.83	7.14	6.65	7.18	5.58	35.58	43.08	39.81	44.04	53.46

3OIZA	4.13	4.95	5.01	5.46	4.05	42.93	55.05	49.75	55.81	63.64
Average	5.44	6.90	6.12	7.01	6.26	32.58	39.76	35.87	39.69	48.76
stdev	1.47	1.63	1.50	1.72	2.16	10.89	11.92	11.63	12.28	15.31

The table lists for all proteins, best RMSD100 and best GDT_TS observed for models generated by BCL and Rosetta. BCL results are presented in 4 columns: SSE only models using native SSE definitions ($\text{BCL}_{\text{N-SSE}}$), complete models using native SSE definitions (BCL_{N}), SSE only models using predicted SSE definitions ($\text{BCL}_{\text{P-SSE}}$), 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

Table 17 lists for all BCL::Fold runs and Rosetta runs, the percentage of benchmark proteins in which the best RMSD100 as well as 0.1th, 1st and 5th percentile (when sorted by RMSD100) are below 6Å, 8Å, 10Å and 12Å. Out of 64 proteins, BCL::Fold was able to generate a best RMSD100 complete model below 8Å for 48 proteins (75%) when using DSSP-derived SSEs and for 46 proteins (72%) when using predicted SSE pools whereas Rosetta generated native-like models for 45 proteins (70%). BCL::Fold RMSD100 values deteriorate for strongly bent β-sheets as SSE backbone conformational sampling in BCL::Fold is limited. Even if the topology is correctly predicted, the RMSD100 values remain high. Figure 23 and Figure 24 show the best RMSD100 SSE-only and complete structural models generated by BCL using predicted SSE pools for a selection of benchmark proteins.

Table 17 Number of best, 0.1th, 1st and 5th percentile RMSD100 models below 6, 8, 10 and 12 Å for BCL and Rosetta

Percentile	Threshold	BCL$_{\text{N-SSE}}$	BCL$_{\text{N}}$	BCL$_{\text{P-SSE}}$	BCL$_{\text{P}}$	Rosetta
Best	<6	41	20	25	15	32
	<8	63	48	59	46	45
	<10	64	64	64	62	63
	<12	64	64	64	64	64
0.1	<6	23	7	16	7	18
	<8	57	27	47	24	38
	<10	64	61	64	56	57
	<12	64	64	64	64	64
1	<6	9	3	5	2	6
	<8	40	12	21	12	26
	<10	64	41	58	43	44
	<12	64	63	64	62	62
5	<6	3	2	2	2	3
	<8	17	5	8	4	11
	<10	55	23	48	22	28
	<12	64	61	64	56	54

The table lists for BCL and Rosetta generated models, the number of proteins (out of 64 proteins) where the best RMSD100, 0.1th percentile, 1st percentile and 5th percentile model when sorted by RMSD100, is below 6, 8, 10 and 12 Å. BCL results are presented in 4 columns: SSE only models using native SSE definitions (BCLN-SSE), complete models using native SSE definitions (BCLN), SSE only models using predicted SSE definitions (BCLP-SSE), complete models using predicted SSE definitions (BCLP). The 5th columns under RMSD100 and GDT_TS are for Rosetta models.

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

Comparison of BCL::Fold runs with predicted and correct SSEs (Table 16) reveals that using native SSE definitions provides an improvement of 0.7 ± 0.9Å in RMSD100 for SSE only models and only 0.1 ± 1.0 Å RMSD100 for complete models after loop construction. As described in Table 11, 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. These moves were not utilized for simulations with correct SSEs.

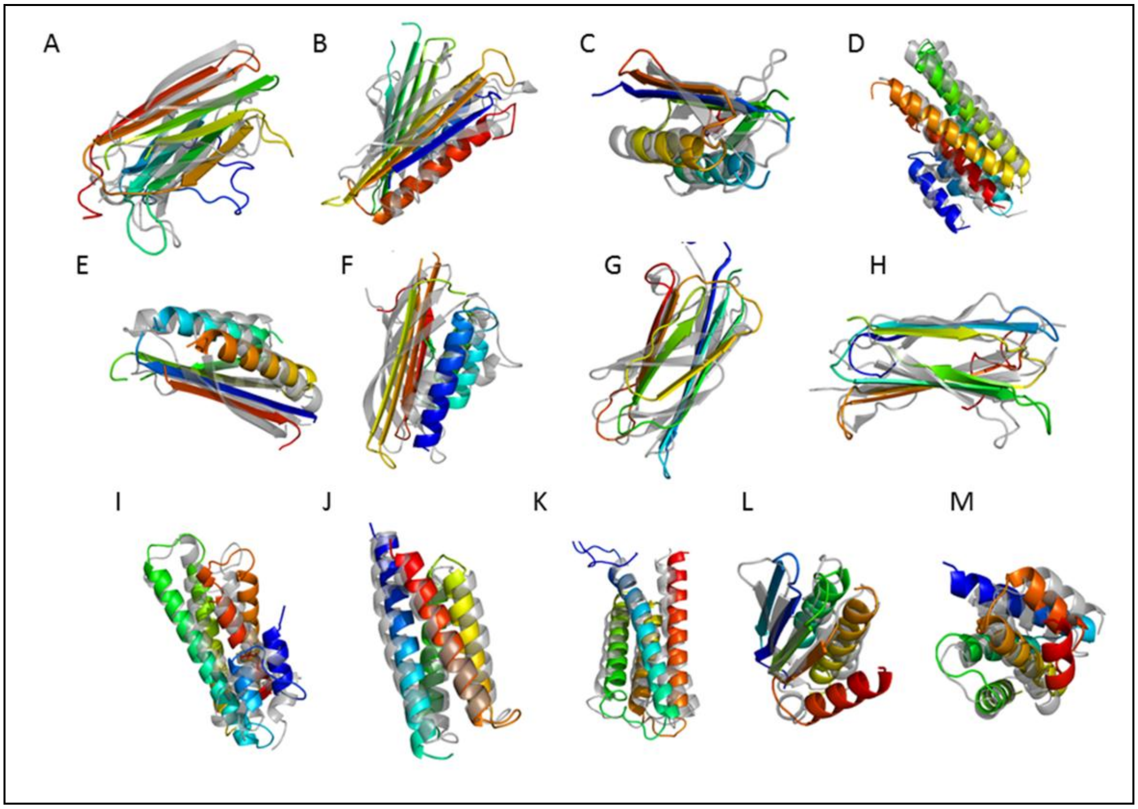


Figure 23 Structures for a selection of best RMSD100 SSE-only models generated by BCL::Fold

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 – 3.74Å **(B)** 1ICXA – 5.43Å **(C)** 1ULRA – 4.34Å **(D)** 1X91A –4.18Å **(E)** 1J27A – 4.27Å **(F)** 1TP6A –5.87Å **(G)** 2CWRA -6.12 Å **(H)** 2RB8A –4.02Å **(I)** 1RJ1A –4.25Å **(J)** 1TQGA – 1.98Å **(K)** 2HUJA –2.60Å **(L)** 3OIZA –5.01Å **(M)** 2V75A –3.33Å

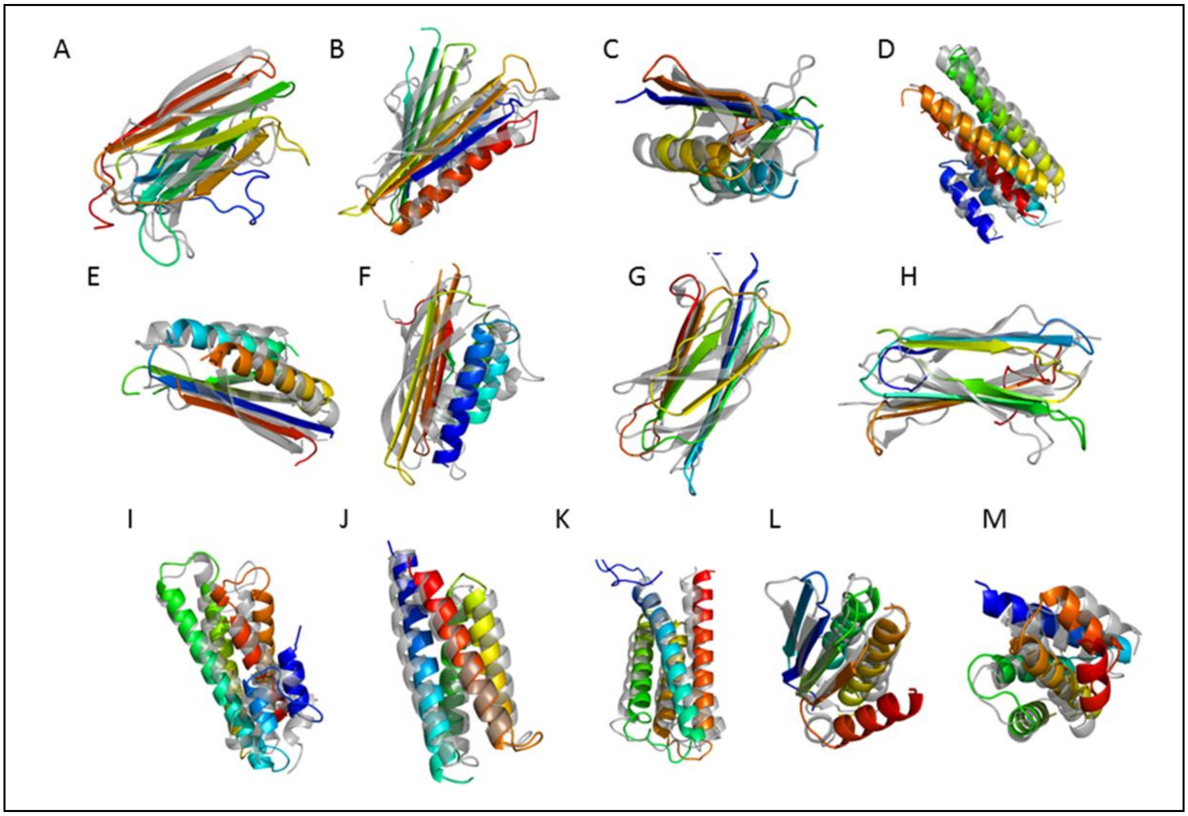


Figure 24 Structures for a selection of best RMSD100 complete models generated by BCL::Fold

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 - 4.29Å **(B)** 1ICXA – 6.04Å **(C)** 1ULRA – 4.71Å **(D)** 1X91A – 4.41Å **(E)** 1J27A – 4.53Å **(F)** 1TP6A – 6.22Å **(G)** 2CWRA 7.71Å **(H)** 2RB8 – 4.78Å **(I)** 1RJ1A – 4.34Å **(J)** 1TQGA 2.23Å **(K)** 2HUJA – 2.74Å **(L)** 3OIZA – 5.46Å **(M)** 2V75A – 3.50Å

BCL::Fold samples local and non-local contacts at rates similar to the distribution in experimental protein structures

Improving structure prediction for large proteins with complex topologies requires sampling more non-local contacts. In analysis of the benchmark set (heat maps and representative set shown in Figure 21), BCL was observed to produce high contact order models. RCO values were calculated over SSEs for both BCL and Rosetta pdbs. The capability to easily sample high RCO topologies by BCL::Fold arises from the fact that local-contacts are not strictly enforced due to the lack of loop residues. Local-contact formation is favored by only one of the energy components used, mostly the loop score and an add move that only applies to SSEs separated by short loops (<8 residues). Especially in formation of β-sheets, moves that shuffle locations or cycle the locations of individual β-strands, as well as moves which switch locations of two β-strands each from a different β-sheet, allows rapid sampling of a variety of possible topologies. This would not be possible so easily in methods that are based on fragment assembly approach for full length sequences.

However, it was also observed that the ranges of RCO sampled were significantly higher than native RCO values for a subset of benchmark proteins. Although this proves the sampling capability of BCL::Fold, it leads to a decrease in the overall accuracy since there is a certain range of RCOs observed for proteins of certain length in nature. In order to improve the accuracy, a new score for evaluating the contact order of models with respect to an expected contact order value for a protein of similar length was developed. For all the benchmark results reported for BCL::Fold, the contact order score was utilized with a weight of 2.5.

For the proteins within the benchmark set, RCO distributions for the 10,000 models produced by BCL::Fold and Rosetta were examined. Table 18 shows the percentage of models with RCO values within the range of native RCO value for cutoffs of 0.010, 0.025, 0.050, 0.075, 0.100, 0.125, 0.175 and 0.200. Complete models generated by BCL::Fold using predicted pools and native SSE definitions do provide similar percentages as Rosetta. For BCL::Fold using predicted SSE pools the following percentage of models have native-like RCO values, 8.49% for ± 0.01, 21.04% for ±0.025 and up to 40.68% for ±0.050. The contact order score was able to move the average of sampled RCO values down to native-like ranges. In most sequence assembly methods, the contact order score is used for the inverse purpose in order to push for higher RCO values.

Table 18 Contact order distributions of BCL and Rosetta generated model with respect to native contact orders

Method	**0.010**	**0.025**	**0.050**	**0.075**	**0.100**	**0.125**	**0.175**	**0.200**
$\text{BCL}_{\text{N-SSE}}$	6.98	18.18	35.45	50.94	64.27	75.70	88.94	91.26
BCL_{N}	8.22	20.61	40.98	59.01	73.73	84.47	95.96	98.69
$\text{BCL}_{\text{P-SSE}}$	9.96	24.09	44.87	61.46	74.90	84.72	92.81	93.69
BCL_{P}	8.49	21.04	40.68	58.20	73.06	84.04	95.65	98.30
Rosetta	9.03	22.18	42.24	59.59	74.09	85.28	97.37	99.13

The table shows contact order distributions for models generated from BCL; : SSE only models using native SSE definitions (BCLN-SSE), complete models using native SSE definitions (BCLN), SSE only models using predicted SSE definitions (BCLP-SSE), complete models using predicted SSE definitions (BCLP) and Rosetta models. For each method, the percentage of models within 0.01, 0.025, 0.05, 0.075, 0.100, 0.125, 0.175 and 0.200 range to the native relative contact order (RCO) are displayed.

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 [9], [132]. 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, top 10,000 by BCL score was then picked and then underwent clustering analysis. The top five best scoring models as well as the best scoring models in each of the larger clusters (~20) then underwent loop construction and side chain packing protocol 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 [7]. This approach was chosen to test the BCL::Fold sampling independent from BCL::Score (compare Chapter III).

Targets in CASP9 were biased towards proteins of known fold. In fact, for ~40 out of the 52 targets submitted for BCL::Fold BETA a template was available. 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. 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 25). 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).

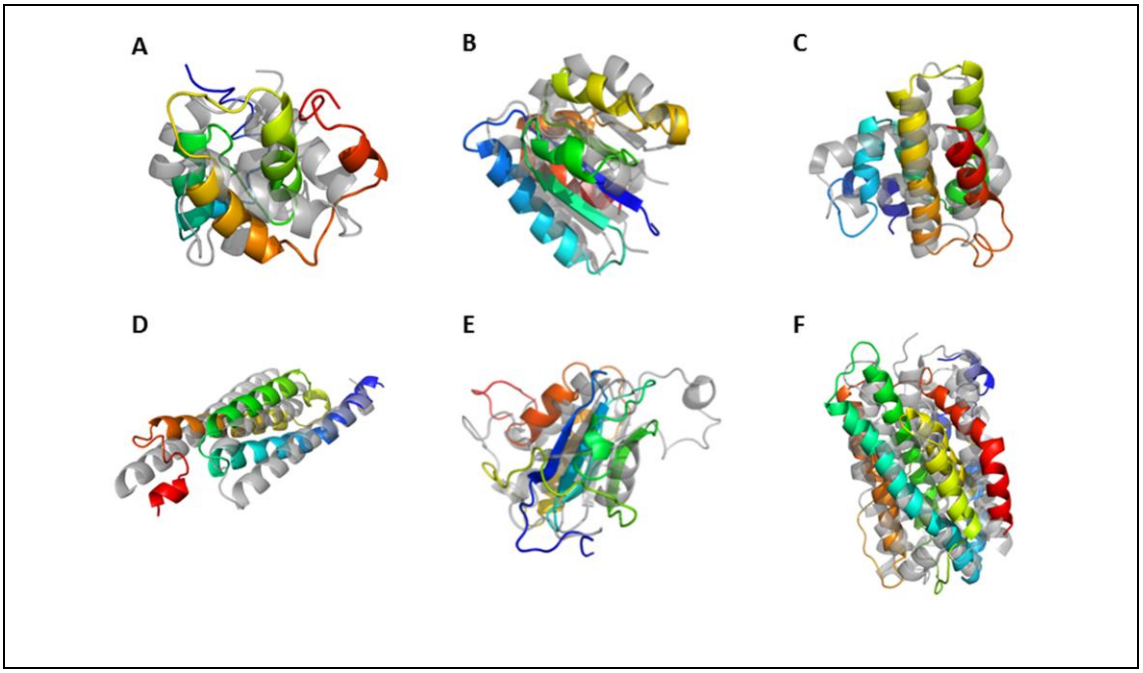


Figure 25 BCL::Fold results from CASP9

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

Assembly of SSEs is a viable tool to predicting protein structures de novo

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 4 proteins with sequence lengths ranging from 88 residues to 293 residues and 4 to 15 SSEs. 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 was not correctly reproduced. With development of more efficient SSE backbone flexibility sampling strategies BCL::Fold can overcome these limitations.

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 cryoEM was previously shown to predict the correct topology for α-helical proteins [2]. 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 20. The algorithm uses the given fasta amino acid sequence and associated secondary structure predictions to generate a pool of secondary structures (Figure 20A). The secondary structure pool is likely to have multiple copies with varying lengths for the same SSEs. The algorithm then picks one SSE randomly from the pool and places it in the origin before starting the minimization. The minimization protocol is composed of a Monte Carlo-based sampling algorithm (Figure 20B) coupled with knowledge-based energy potentials (Figure 20C). 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 1D).

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 [92] culling server and includes 64 proteins of lengths ranging from 83 to 293 residues with <30% sequence similarity. The set contains different topologies including all α-helical, all β-strand, and mixed αβ folds (Table 13). The original PDB entries and FASTA sequences of the selected proteins were downloaded from the PDB [5]. The secondary structure definitions were regenerated using DSSP [93], since the native SSE definitions found in some PDB files had inconsistencies.

Secondary structure prediction and preparation of secondary structure pool

JUFO [14] and PSIPRED [15] were obtained from the authors of the methods and installed locally. In addition the sequence alignment tool BLAST [133], [134] was installed locally to create the required position specific scoring matrices. These alignment files along with FASTA files for each protein are used as input to the secondary structure prediction methods. An initial version of the pool named "highest pool" is prepared by taking the highest probability for each residue and assigning it the corresponding secondary structure type. However, this was shown to cause problems with over-prediction of secondary structures as well as missing short breaks. In order to overcome this problem, a new Monte-Carlo based minimization method was developed to optimize this initial set of secondary structure assignments. For both the initial "highest pool" as well as the minimized pool definitions, α-helices shorter than 5 residues and β-strands shorter than 3 residues are excluded.

Pool agreement score for measuring deviation between two sets of secondary structure assignments

Q3 is the most commonly used method for evaluating secondary structure assignments [130]. Q3 evaluates the percentage of residues with correct secondary structure assignments. However, since the actual identification of an SSE is more crucial for BCL::Fold than the exact length of the SSE, a difference measure named "pool agreement score” was developed which penalizes deviations between two sets of secondary structure elements, considering per SSE under- and over-prediction, SSE length deviation, missed or additional secondary structure. An asymmetric function $f(A, B)$ evaluates two sets of secondary structure elements A and B.

Pseudo code:

```
deviation = 0
foreach ssea in A:
        if ssea is coil: next
        overlap_sses = all sses in B that overlaps with ssea and have same type as ssea
        if overlap_sses is empty
                deviation += 3 * log(length(ssea) + 1)
                next
        endif
        foreach sseo in overlap_sses
                overlap_left    = first_seq_id(ssea) – first_seq_id(sseo)
                overlap_right  = last_seq_id(sseo) – last_seq_id(ssea)
                length_difference = overlap_left + overlap_right
                deviation += log(max(0,abs(overlap_left)-nr_tolerated_residues)+1)
                deviation += log(max(0,abs(overlap_rigth)- nr_tolerated_residues)+1)
                deviation += log(abs(length_difference)+1)
        end
end
```

The deviation between two sets of secondary structure elements is defined as

$$d = f(A, B) + f(B, A)$$

The *nr_tolerated_residues* of a single residue is added as tolerance measure to compensate for the few residue differences in the lengths of SSEs that can be observed when comparing secondary structure element assignments by experimentalists. A missing SSE is penalized with a factor of three, since there are three terms contributing to the deviation if an overlapping SSE was found. Instead of using absolute values, the logarithm is used, so a missing SSE weighs more than the actual length of the SSE that was not found. For the overlaps, it also favors a balanced overlap on either end rather than an overlap where many more residues are missing from just one end.

Scoring terms for secondary structure pool evaluation

All terms are error functions of the Standard Score (z-score). Each z-score is defined by:

$$z_{SS}(p_{SS}) = \frac{p_{SS}-\mu_{SS}}{\sigma_{SS}}$$

With: p_{SS} probability of for the secondary structure assigned

μ_{SS} mean for a specific secondary structure

σ_{SS} standard deviation for a specific secondary structure

The scoring term is the error function with a confidence threshold:

$$S_{SS}(p_{SS}) = -erf(z(p_{SS}) - c)$$

With: p_{SS} probability of residue i for a specific secondary structure

c confidence threshold

The confidence threshold defines the z-score, above which the scoring term turns negative. This term can be used to adjust the “sensitivity” of the scoring function – permitting more than what is statistically expected (smaller c) or being more strict to what is allowed (larger c).

Single residue confidence: This score evaluates the probability of a single residue for the current secondary structure assignment as the error function of the z-score. This score is derived from the databank of proteins.

Multiple residue average confidence: This score evaluates the average probability over n residues of the same secondary structure as the error function of the score. The z-score is derived from the databank of proteins. The average probability at position k $p_{k,SS}$ is defined by:

$$p_{SS} = \frac{1}{n}\sum_{i=k}^{k+n-1} p_{i,SS}$$

With: $p_{i,SS}$ single residue probability for the secondary structure assigned

Confidence Deviation: This score evaluates the probability of a residue with the lowest probability within a secondary structure element as the error function of the z-score. This z-score is defined by the mean and standard deviation calculated from the probabilities within this secondary structure element. If this probabilities z-score is within the confidence interval, it is 0. If the probability is outside, it is positive according to the z-score. The mean and standard deviation of an SSE is derived by:

$$\mu_{SSE} = \frac{1}{l}\sum_{i=k+s}^{k+l-1} p_{i,SS}$$

$$\sigma_{SSE} = \sqrt[2]{\frac{1}{l}\sum_{i=k+s}^{k+l-1-s}\left(p_{i,SS} - \mu_{SSE}\right)^2}$$

$$z(SSE) = \frac{p_{min,SS} - \mu_{SSE}}{\sigma_{SSE}}$$

With: l length of the SSE

k first residue in SSE

s number of residues to ignore on edges

Prediction slope: This score evaluates the least square regression over n residues at the beginning and end of a secondary structure element. The resulting slopes are evaluated as the error function of the z-score. The z-score is derived from the databank of proteins.

$$\hat{\beta}_{left,SSE} = \frac{\sum_{i=k}^{k+n-1} i p_{i,SS} - \frac{1}{n}\sum_{i=k}^{k+n-1} i \sum_{i=k}^{k+n-1} p_{i,SS}}{\sum_{i=k}^{k+n-1} i^2 - \frac{1}{n}\left(\sum_{i=k}^{k+n-1} i\right)^2}$$

$$\hat{\beta}_{right,SSE} = -\frac{\sum_{i=k+l-n+1}^{k+l} i p_{i,SS} - \frac{1}{n}\sum_{i=k+l-n+1}^{k+l} i \sum_{i=k+l-n+1}^{k+l} p_{i,SS}}{\sum_{i=k+l-n+1}^{k+l} i^2 - \frac{1}{n}\left(\sum_{i=k+l-n+1}^{k+l} i\right)^2}$$

With: l length of the SSE

k first residue in SSE

n number of residues considered on each side

Monte Carlo-based sampling algorithm and temperature control

Unless a starting structural model is specified, 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 picked 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[63],

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

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 10th step to allow a linear decrease of acceptance ratio from 0.5 to 0.2.

This evaluation can lead to four different results; (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 an “rejected” or “skipped” state, then 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 replaced with this one.

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 11 and Table 15. 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.

Loop building

Missing loop residues were built on to the model predicted by BCL::Fold using an in-house CCD based loop building protocol [80]. 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 typical 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) contact order and lastly (11) an entropy term that evaluates all the residues not represented in the model, using the previous ten potentials. All scoring functions are implemented within the BCL. In BCL::Fold runs with predicted SSE pools, two additional terms specialized on sse predictions (one for Jufo, one for PSIPRED) was added, making it a total of thirteen terms.

All knowledge based potentials have been derived from a databank that contained 3409 high resolution x-ray crystallography protein structures compiled using the PISCES server [92]. 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 [23], 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 12.

Benchmark analysis

The models produced by BCL::Fold benchmarks are evaluated by looking at following quality measures root-mean-square-deviation (RMSD), RMSD100 [131] and GDT_TS [12]. These measures are calculated over Cα atoms of all the residues in α–helices and β–strands in the models. In addition, contact order [129] values were calculated by looking at average sequence separation of contacts defined as having Cβ ($H_{\alpha 2}$ for Glycine) atoms within 8Å distance. Relative contact order (RCO) values were calculated by normalizing contact order values by the length of the sequence.

For each BCL::Fold run of 10,000 models for each of the 64 proteins in the benchmark set, an initial filtering is done to remove any incomplete models. It is possible that certain topologies constructed by BCL::Fold can make it impossible to complete the model due to loop restraints and the minimization can terminate early. In addition, models with significant clashes between amino acids or SSEs are also filtered out.

Protein structure prediction using Rosetta

Rosetta [20], [21], [23] protein structure prediction program was used to generate 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 then underwent the same analysis as the models produced by BCL::Fold. Since Rosetta models have full chain and BCL::Fold models do not have loop residues, the secondary structures in Rosetta models were determined using DSSP [93] and the quality calculations were completed considering Cα 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

The focus of the presented work was twofold. Firstly, a rapid fitting method for atomic detail protein structures into electron density maps was presented. It employed geometric hashing, known from robotics, were rapid pattern matching is required. Secondly, a knowledge based scoring potential was presented, that focuses on the evaluation of the assembly of secondary structure elements to define the topology and stability of a protein. This potential was employed in the BCL::Fold method that predicts protein structure *de novo* from the primary amino acid sequence. The following discussion will elaborate on the development of these methods, their current achievements and their future potential.

BCL::EM-Fit

The objective for developing BCL::EM-Fit was to find a fast method to fit atomic protein structures into medium resolution electron density maps. This enables faster analysis of cryoEM maps and offers the possibility for screening the maps against many structural models.

The geometric hashing protocol in conjunction with the Monte Carlo/Metropolis algorithm has proven to be able to fit proteins of different secondary structure content into electron density maps of resolutions up to12 Å. It could be applied to density map segments of large macromolecular assemblies like GroEL and Adenovirus. It was able to solve the problem of identifying the handedness of a density map, which is a general problem for three dimensional reconstructions from two dimensional projections.

The method could show that it lives up to all expected standards, which comprise completeness in identifying all possible positions for a given protein structure into the electron density map and the accuracy of the fitting required to identify symmetry, definition of protein-protein interfaces and the identification of unoccupied electron density that accounts for proteins of unknown structure.

It remains to be shown that the algorithm can be used to screen a density map against a dataset of possible structures to identify the most likely structure that would fit. Although some initial homology model and cross fitting experiments succeeded in identifying the correct structure for a given density map, the experiments where designed to show limitations of the procedure. Further speed-up of the minimization was already achieved with a general purpose graphical processing unit (GPGPU) implementation of the minimization algorithm [40].

BCL::Score

The knowledge based energy terms were developed focusing on evaluating the native-likeness of a proteins structure based on the topology defined by the arrangement of secondary structure elements. For this, novel terms were defined, that evaluate the packing of secondary structure elements represented by fragments. Additionally, a loop length potential was introduced, that evaluates the omitted loops. A contact order potential warrants, that proteins do not show too high of a complexity – nothing seen in natural proteins. Special focus was paid to define proper background probabilities to leverage the features that the energy potential can take advantage of, when evaluating models for their native-likeness.

Each individual term was able to enrich at least one set the BCL folded protein structures, indicating that they are orthogonal to the scoring terms that were used in the generation of the models. It also indicates the ability to identify non-native structure like in the case of perturbed structural models, starting from the native protein.

The loop length and loop closure potential was not able to enrich ROSETTA generate model sets, which is expected since they are continuous amino acid sequences and should fulfill any constraints given by nature. Their backbone still resembles a native backbone trace, increasing the chance of having a proper sequence length to Euclidean distance ratio.

The clash potentials exhibited good performance for the randomly perturbed structures, as they were generated with no consideration against special overlap. For the BCL::Fold and ROSETTA model set, the sampling and present scoring algorithms prevented sever clashes.

Since almost all scores are pairwise decomposable, the time efficiency of the energy evaluation could be leveraged above the advantage of their simplicity by the ability to reuse pairwise evaluations of the score, if relative arrangements of two features did not change from one evaluated model to another.

The BCL::Fold benchmark has shown that the scoring terms can be used in *de novo* structure prediction. One of the challenges that are still needed to be overcome is the relative weighing of the scoring terms. There might be an optimal weight set to achieve optimal enrichment for a set of native-like and random models like shown in the BCL::Score benchmark. Together with a sampling algorithm, the consensus scoring function has to be able to drive the assembly to the native like structure, which might require going through non-native-like states of the protein model.

Scoring terms that evaluate the model given experimental restraints are being developed, but also need to be integrated carefully into the consensus scoring function.

BCL::Fold

BCL::Fold was benchmarked using 64 proteins with diverse topologies, SSE contents, varying sequence lengths in the range of 88 to 293 amino acids and a RCO range of 0.13 to 0.46 with an average of 0.29 ± 0.07. 10,000 SSE-only models were generated for each of the 64 proteins using 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 48 proteins (75%) when using native SSE definitions and for 46 proteins (72%) when using predicted SSE pools. When SSE only models are considered, the correct topology was found for 63 proteins (98%) using native SSE definitions, 59 proteins (92%) using predicted SSE pools. Further detailed analysis of results could be found in Chapter IV.

The results show BCL::Fold's novel approach to *de novo* protein structure prediction is promising and can overcome current limitations. A more detailed analysis on the problems that the algorithm has with certain classes (α-helical vs. β-strand or mixed) of proteins will be required. A few approaches are worked on to address the sampling efficiency of BCL::Fold. Dr. Brian Weiner introduces fold full and partial fold templates, so that the assembly can start with native-like topologies. Those fold templates are implemented sequence and order independent, so that they follow the idea of BCL::Fold to enable any SSE arrangement unrestricted by folding pathways restricted by loop connections. There has been significant progress in this project and this method is currently being benchmarked to be published before the end of 2011.

In order to evaluate the performance of a *de novo* protein structure prediction method one has to define and objective. Classically, a model is considered good if the RMSD to the native is optimal. For large proteins this is rarely achievable. Defining the success based on capturing the correct topology, or just the core of the protein correctly, is a project that is worked on. Ultimately, the measure should be defined depending on the context the generated models are used in. If they are used for small molecule docking, a highly accurate model with low RMSD is necessary. If the model is used to define the arrangement of components in an electron density map, a secondary structure arrangement/topology close to the correct structure is sufficient. Details might be easier elucidated, once the protein is seen in its biological context and the interface between proteins can give additional information to restrain the protein structure problem further.

BCL::Fold was introduced as an alternative to current protein structure prediction methods, trying to overcome size limitations by incorporating experimental restraints. The algorithm is implemented modularly in the sampling as well as the scoring part. This enables plug’n’play extensions of the protocol. It is currently serving as a framework for many projects in the Meiler Laboratory incorporating new scoring terms and sampling moves using cryoEM, NMR and EPR restraints. Being developed in merely six years, BCL::Fold did not play out its full potential and can be successful when consequently developed and tested for and on the systems of recent scientific interest.

APPENDIX

Contained in the appendix is the usage of all BCL programs as they were used to generate the data for the different chapters. All command lines are based of the BCL trunk revision 3966 at https://gforge.accre.vanderbilt.edu/svn/bcl.

A. BCL::EMFIT APPLICATIONS

The fitting algorithm is implemented as an application with the BioChemistry Library. The required inputs are a protein (PDB) file with the atomic coordinates of the protein to be fitted and an MRC density map, containing the electron density map.

FitInDensity

The most basic commandline contains the location of the input files as well as the desired output location for the results, comprising the result table with the file names of the fitted structure pdb files, cross correlation coefficient (CCC) after intial fit and after minimization as well the RMSD relative to the input pdb.

Command line:

```
bcl.exe FitInDensity 1ubi.pdb 1ubi_res_6.6voxelsize_2.200Gaussian.mrc -mrc_resolution 6.6
-hash_storage HashMap –prefix result_path/ -protein_storage File result_path/pdbs/ Create
–coordinatesystem Spherical -atoms N CA C O
```

Input files: 1ubi.pdb 1ubi_res_6.6voxelsize_2.200Gaussian.mrc

Output files: result_path/result.table; result_path/pdbs/transformed*.pdb; result_path/pdbs/transformedmin*.pdb

FitInDensityMinimize

An additional application aids in minimizing initial fits for the purpose of refinement. It requires the same inputs as the FitInDensity program. The output is a single pdb file, which is the location of the minimized fit.

Command line:

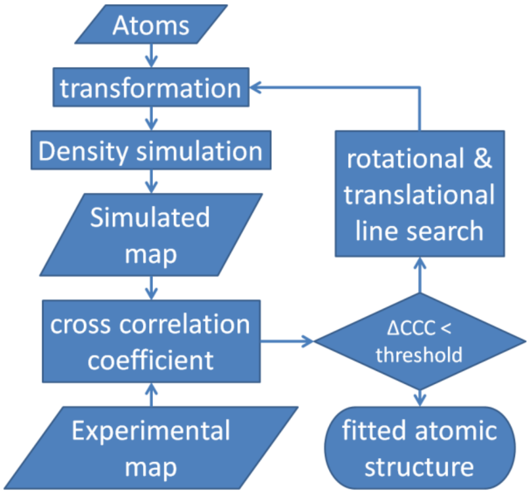
```
bcl.exe FitInDensityMinimize 1ubi.pdb 1ubi_res_6.6voxelsize_2.200Gaussian.mrc
-mrc_resolution 6.6 –approximator mc –prefix result_path/
```

Input files: 1ubi.pdb 1ubi_res_6.6voxelsize_2.200Gaussian.mrc

Output files: result_path/transformed_min.pdb

POWELL approximator with golden section line search

Besides the standard MonteCarlo/Metropolis approximator (“mc”), a Powell minimizer is implemented, with golden section line search. This minimizer is slower, due to more objective function evaluations, but will find the absolute local minimum reliably.



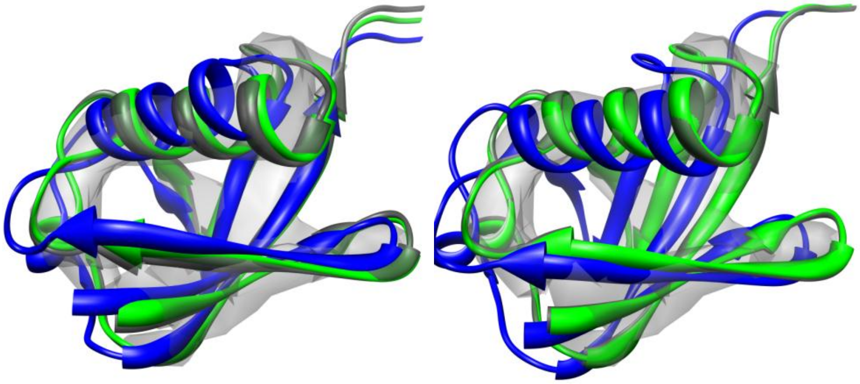
GPGPU accelerated approximator

Using the parallelizability of CCC calculation and electron density map simulation, it is possible to accelerate the fitting by implementing the Powell minimization an general purpose graphical processing units (GPGPU). This acceleration leads to significant speed-ups in the computation time:

fit	CCC		Intel(R) Xeon(R)	NVIDIA			ATI Radeon
	Start	Final	W3570@ 3.20GHz	Tesla C1060	Tesla C2050	GTX 470	HD 5970
1	0.892	0.989	38.486	7.260	1.423	2.398	2.486
2	0.862	0.989	35.579	6.944	1.468	2.138	2.443
3	0.698	0.842	23.888	7.151	1.476	2.184	2.518

Computation times in seconds for the Powell optimization for 3 different initial fits of 1ubi on CPU and 4 different GPU architectures in double floating point precision. Out of the three refined placement, only two were above 0.9 CCC (Fig. 2) while their backbone RMSD to the correct placement was below 0.1 Å. The third initial fit could not be refined using those parameters.

The fitting results are still of high quality:



Initial (blue) and refined (green) placements of 1ubi for initial fit 1 (left) and initial fit 2 (right). The optimal placement is shown in grey. A 6.9 Å resolution density map simulated using Colores is shown as transparent envelope.

PDBToDensity

For all benchmark experiments, it was necessary to simulate electron density maps from the atomic coordinate files (pdbs). A program within the BCL was created to do that. The input is a pdb file with atomic coordinates, the desired resolution for the density map, and the simulation algorithm.

Command line:

```
bcl.exe PDBToDensity 1ubi.pdb -resolution 6.6 -voxel_size 2.2 -kernel GaussianSphere -noise
0.8 –prefix result_path/
```

Input files: 1ubi.pdb

Output files: 1ubi_res_6.6voxelsize_2.200GaussianSphere_noiseccc_0.792.mrc

An mrc density map is generated with a resolution of 6.6 Å, 2.2 Å Voxel size using a Gaussian sphere to represent the electron density of a given atom and and random noise is added, so that the CCC to the starting map is just below 0.8.

B. BCL::SCOREPROTEIN APPLICATIONS

The knowledge based potentials are derived from a non-redundant set of protein x-ray structures of high resolution. This database of structures is derived using the PISCES sequence culling server [92]. All membrane proteins are excluded before culling using the PDBTM (PDB of trans-membrane proteins), since all potentials are initially derived only for soluble proteins. Different interactions play a role in membrane proteins and are derived separately.

The result is a list of pdb 4 letter codes and the chain id of the sequence. These pdbs are parsed with the BCL::PDBConvert application, to extract the individual chains as pdb files and the according fasta sequences. If a protein cannot be read, it is removed from the list to cull, and PISCES sequence culling will be restarted, until all pdbs are readable by the BCL.

The sequences are subject to three iterations of blast search resulting in position specific scoring matrices [134], which are input to protein structure prediction algorithms, JUFO [14] and PSIPRED [15].

StatisticProteins

This application derives histograms for all features that are used to derive the knowledge based potentials. The input is a list of protein structures (pdb 5 letter codes), and the output are histogram files of desired resolution (angular and distance) of the features observed in the protein structures.

Command line:

```
bcl.exe -pdblist /blue/meilerlab/apps/PISCES/data/ current_soluble_5.ls 1 -aadistance -
radiusofgyration -loops -loop_closure -ssepacking -neighbor_count_sasa -neighbor_vector_sasa
-ols_sasa -phi_psi -sse_count –sspred JUFO PSIPRED -contact_order -multimer 1 -
convert_to_natural_aa_type
```

Input files: current_soluble_5.ls which references pdb files in the folder /blue/meilerlab/apps/PISCES/data/??/ where ?? are the second and third character of the pdb 5 letter code.

Output files: *.histogram* for each of the requested potentials.

Examples visualize potentials

Within the BCL, each potential is calculated using the histogram of features as input. The examples demonstrate the usage of the scoring functions. If used with “-message_level Debug”, all potentials are written as gnuplot script files, which can be used to generate heatmaps for the potentials:

Command lines:

```
bcl.exe Examples –namespace Score –message_level Debug
gnuplot -f {potential_name}.gnuplot
```

Output files: *.gnuplot and *.png

The gnuplot files can be adjusted for more appropriate plot scaling or visualization options.

ScoreProtein

An individual pdb file or a set of proteins can be scored at once, with all scoring function introduced in this manuscript. Additionally, if a template structure is given. protein similarity measures can be calculated as well. The output is a table, with one row for each protein, and columns for all scores and qulity measures.

If template pdb is given, the terminal output contains the rank of the template structure for all of the scores. It is also possible to give any quality measure and a cutoff to calculate the enrichment for model below that threshold. This gives an indicator for how well the individual potential discriminates for native like protein structures in a set of models.

Command line scoring:

```
bcl.exe ScoreProtein -pdblist pdbs.ls -score_table_write scores.table -template template.pdb -
quality RMSD GDT_TS -atoms CA -convert_to_natural_aa_type -sspred JUFO PSIPRED
```

Input files: pdbs.ls a list of pdb files names

Output files: scores.table

Commandline enrichment:

```
bcl.exe ScoreProtein -score_table_read scores.table -rank template.pdb -weight_set assembly.scoreweights -sspred JUFO PSIPRED -enrichment 0.1 8.0 10 RMSD100 less
```

Input files: scores.table; assembly.scoreweights

Output files: none, all the enrichment and ranks are written to the terminal

The enrichment is calculated by balancing the set of scores models, so that the resulting table has a fraction of 0.1 with RMSD100 less than 8Å. 10 different tables are generated with different subsets from the input scores.table. The assembly.scoreweights is used to calculate the sum – the weighted consensus score from each of the scoring terms.

MinimizeScoreWeightSet

Using the score tables from scoring a set of proteins with calculated quality measures, an optimal weight set for the consensus scoring function can be derived. In a Monte Carlo minimization, the weight sets are randomly changed. If the consensus score enriches the protein data set better. For each table in a list of tables, where each tables contains the scores for a set or structural models for a protein, the enrichment is calculated. The enrichment is optimized for all protein sets.

Commandline:

```
bcl.exe MinimizeScoreWeightSet -list tables.ls -weight_set weights.table -weight_set_write optimized_ -enrichment RMSD100 8 0.1 10 100 -sort_order less -scheduler PThread 8 -mc_tot_unimproved 10000 500 -number_repeats 5 -keep_positive
```

Input files: tables.ls and the tables that are listed in that file

Output files: optimized_*.weights one for each repeat

BIBLIOGRAPHY

[1] N. Woetzel, S. Lindert, P. L. Stewart, and J. Meiler, "BCL::EM-Fit: Rigid body fitting of atomic structures into density maps using geometric hashing and real space refinement.," *Journal of structural biology*, May 2011.

[2] S. Lindert, R. Staritzbichler, N. Wötzel, M. Karakaş, P. L. Stewart, and J. Meiler, "EM-fold: De novo folding of alpha-helical proteins guided by intermediate-resolution electron microscopy density maps.," *Structure (London, England : 1993)*, vol. 17, no. 7, pp. 990-1003, Jul. 2009.

[3] F. H. CRICK, "On protein synthesis.," *Symposia of the Society for Experimental Biology*, vol. 12, pp. 138-63, Jan. 1958.

[4] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips, "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis.," *Nature*, vol. 181, no. 4610, pp. 662-6, Mar. 1958.

[5] H. M. Berman, "The Protein Data Bank: a historical perspective.," *Acta crystallographica. Section A, Foundations of crystallography*, vol. 64, no. 1, pp. 88-95, Jan. 2008.

[6] K. Wüthrich, "Protein structure determination in solution by NMR spectroscopy.," *The Journal of biological chemistry*, vol. 265, no. 36, pp. 22059-62, Dec. 1990.

[7] A. Fiser and A. Sali, "Modeller: generation and refinement of homology-based protein structure models.," *Methods in enzymology*, vol. 374, pp. 461-91, Jan. 2003.

[8] P. Bradley, K. M. S. Misura, and D. Baker, “Toward high-resolution de novo structure prediction for small proteins.,” *Science (New York, N.Y.)*, vol. 309, no. 5742, pp. 1868-71, Sep. 2005.

[9] A. Kryshtafovych, O. Krysko, P. Daniluk, Z. Dmytriv, and K. Fidelis, “Protein structure prediction center in CASP8.,” *Proteins*, vol. 77 Suppl 9, pp. 5-9, Jan. 2009.

[10] N. Alexander, N. Woetzel, and J. Meiler, *Bcl::Cluster: A method for clustering biological molecules coupled with visualization in the Pymol Molecular Graphics System*. IEEE, 2011, pp. 13-18.

[11] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer, “MaxSub: an automated measure for the assessment of protein structure prediction quality.,” *Bioinformatics (Oxford, England)*, vol. 16, no. 9, pp. 776-85, Sep. 2000.

[12] A. Zemla, Venclovas, J. Moult, and K. Fidelis, “Processing and evaluation of predictions in CASP4.,” *Proteins*, vol. 5, pp. 13-21, Jan. 2001.

[13] D. Cozzetto, A. Kryshtafovych, K. Fidelis, J. Moult, B. Rost, and A. Tramontano, “Evaluation of template-based models in CASP8 with standard measures.,” *Proteins*, vol. 77 Suppl 9, pp. 18-28, Jan. 2009.

[14] J. Meiler, A. Zeidler, F. Schmaeschke, and M. Mueller, “Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks,” *Journal of Molecular Modeling*, vol. 7, no. 9, pp. 360-369, Sep. 2001.

[15] D. T. Jones, “Protein secondary structure prediction based on position-specific scoring matrices.,” *Journal of molecular biology*, vol. 292, no. 2, pp. 195-202, 1999.

[16] B. Rost, “PHD: predicting one-dimensional protein structure by profile-based neural networks.,” *Methods in enzymology*, vol. 266, pp. 525-39, Jan. 1996.

[17] M. Karakaş, N. Woetzel, and J. Meiler, “BCL::contact-low confidence fold recognition hits boost protein contact prediction and de novo structure determination.,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 17, no. 2, pp. 153-68, Feb. 2010.

[18] B. Rost, “Review: protein secondary structure prediction continues to rise.,” *Journal of structural biology*, vol. 134, no. 2-3, pp. 204-18, 2001.

[19] B. Rost, “Prediction in 1D: secondary structure, membrane helices, and accessibility.,” *Methods of biochemical analysis*, vol. 44, pp. 559-87, Jan. 2003.

[20] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, “Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions.,” *Journal of molecular biology*, vol. 268, no. 1, pp. 209-25, Apr. 1997.

[21] K. T. Simons, B. A. Fox, I. Ruczinski, C. Kooperberg, C. Bystroff, and D. Baker, “Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins.,” *Proteins*, vol. 34, no. 1, pp. 82-95, Jan. 1999.

[22] R. Das et al., “Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home.,” *Proteins*, vol. 69 Suppl 8, no. May, pp. 118-28, Jan. 2007.

[23] A. Leaver-Fay et al., “ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules.,” *Methods in enzymology*, vol. 487, pp. 545-74, Jan. 2011.

[24] R. Bonneau et al., “De novo prediction of three-dimensional structures for major protein families.,” *Journal of molecular biology*, vol. 322, no. 1, pp. 65-78, Sep. 2002.

[25] N. Alexander, M. Bortolus, A. Al-Mestarihi, H. Mchaourab, and J. Meiler, “De novo high-resolution protein structure determination from sparse spin-labeling EPR data.,” *Structure (London, England : 1993)*, vol. 16, no. 2, pp. 181-95, Feb. 2008.

[26] S. Kalkhof, S. Haehn, M. Paulsson, N. Smyth, J. Meiler, and A. Sinz, “Computational modeling of laminin N-terminal domains using sparse distance constraints from disulfide bonds and chemical cross-linking.,” *Proteins*, vol. 78, no. 16, pp. 3409-27, Dec. 2010.

[27] J. Meiler and D. Baker, “Rapid protein fold determination using unassigned NMR data.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 26, pp. 15404-9, 2003.

[28] M. D. Tyka, F. DiMaio, M. L. Baker, W. Chiu, and D. Baker, “Refinement of protein structures into low-resolution density maps using rosetta.,” *Journal of molecular biology*, vol. 392, no. 1, pp. 181-90, 2009.

[29] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, “CHARMM - A PROGRAM FOR MACROMOLECULAR ENERGY, MINIMIZATION, AND DYNAMICS CALCULATIONS,” *Journal of Computational Chemistry*, vol. 4, no. 2, pp. 187-217, 1983.

[30] M. J. Sippl, “Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins.,” *Journal of molecular biology*, vol. 213, no. 4, pp. 859-83, Jun. 1990.

[31] P. Bradley et al., “Rosetta predictions in CASP5: successes, failures, and prospects for complete automation.,” *Proteins*, vol. 53 Suppl 6, pp. 457-68, Jan. 2003.

[32] J. Lepault, F. P. Booy, and J. Dubochet, “Electron microscopy of frozen biological suspensions.,” *Journal of microscopy*, vol. 129, no. 1, pp. 89-102, 1983.

[33] S. D. Saban, R. R. Nepomuceno, L. D. Gritton, G. R. Nemerow, and P. L. Stewart, “CryoEM structure at 9A resolution of an adenovirus vector targeted to hematopoietic cells.,” *Journal of molecular biology*, vol. 349, no. 3, pp. 526-37, Jun. 2005.

[34] L. Montesano-Roditis, D. G. Glitz, R. R. Traut, and P. L. Stewart, “Cryo-electron microscopic localization of protein L7/L12 within the Escherichia coli 70 S

ribosome by difference mapping and Nanogold labeling.," *The Journal of biological chemistry*, vol. 276, no. 17, pp. 14117-23, Apr. 2001.

[35] S. D. Saban, M. Silvestry, G. R. Nemerow, and P. L. Stewart, "Visualization of alpha-helices in a 6-angstrom resolution cryoelectron microscopy structure of adenovirus allows refinement of capsid protein assignments.," *Journal of virology*, vol. 80, no. 24, pp. 12049-59, Dec. 2006.

[36] S. Lindert, P. L. Stewart, and J. Meiler, "Hybrid approaches: applying computational methods in cryo-electron microscopy.," *Current opinion in structural biology*, vol. 19, no. 2, pp. 218-25, Apr. 2009.

[37] W. Wriggers and S. Birmanns, "Using situs for flexible and rigid-body fitting of multiresolution single-molecule data.," *Journal of structural biology*, vol. 133, no. 2-3, pp. 193-202, 2001.

[38] J. Meiler, W. Peti, and C. Griesinger, "DipoCoup: A versatile program for 3D-structure homology comparison based on residual dipolar couplings and pseudocontact shifts.," *Journal of biomolecular NMR*, vol. 17, no. 4, pp. 283-94, Aug. 2000.

[39] J. Meiler, R. Meusinger, and M. Will, "Fast determination of 13C NMR chemical shifts using artificial neural networks.," *Journal of chemical information and computer sciences*, vol. 40, no. 5, pp. 1169-76, Aug. 2000.

[40] N. Woetzel, E. W. Lowe, and J. Meiler, *Poster: GPU-accelerated rigid body fitting of atomic structures into electron density maps*. IEEE, 2011, pp. 265-265.

[41] F. Fabiola and M. S. Chapman, "Fitting of high-resolution structures into electron microscopy reconstruction images.," *Structure (London, England : 1993)*, vol. 13, no. 3, pp. 389-400, Mar. 2005.

[42] W. Wriggers and P. Chacón, "Modeling tricks and fitting techniques for multiresolution structures.," *Structure (London, England : 1993)*, vol. 9, no. 9, pp. 779-88, Sep. 2001.

[43] A. Korostelev, R. Bertram, and M. S. Chapman, "Simulated-annealing real-space refinement as a tool in model building," *Acta Crystallographica Section D Biological Crystallography*, vol. 58, no. 5, pp. 761-767, Apr. 2002.

[44] A. M. Roseman, "Docking structures of domains into maps from cryo-electron microscopy using local correlation.," *Acta crystallographica. Section D, Biological crystallography*, vol. 56, no. 10, pp. 1332-40, Oct. 2000.

[45] W. Wriggers, R. A. Milligan, and J. A. McCammon, "Situs: a package for docking crystal structures into low-resolution maps from electron microscopy," *Journal of Structural Biology*, vol. 125, no. 2-3, pp. 185–195, 1999.

[46] T. D. Goddard, C. C. Huang, and T. E. Ferrin, "Visualizing density maps with UCSF Chimera.," *Journal of structural biology*, vol. 157, no. 1, pp. 281-7, Jan. 2007.

[47] J. A. Velazquez-Muriel and J.-M. A. Carazo, "Flexible fitting in 3D-EM with incomplete data on superfamily variability.," *Journal of structural biology*, vol. 158, no. 2, pp. 165-81, 2007.

[48] F. Tama, O. Miyashita, and C. L. Brooks, “Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM.,” *Journal of structural biology*, vol. 147, no. 3, pp. 315-26, 2004.

[49] G. F. Schröder, A. T. Brunger, and M. Levitt, “Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution.,” *Structure (London, England : 1993)*, vol. 15, no. 12, pp. 1630-41, Dec. 2007.

[50] L. G. Trabuco, E. Villa, E. Schreiner, C. B. Harrison, and K. Schulten, “Molecular dynamics flexible fitting: a practical guide to combine cryo-electron microscopy and X-ray crystallography.,” *Methods (San Diego, Calif.)*, vol. 49, no. 2, pp. 174-80, Oct. 2009.

[51] M. Topf, K. Lasker, B. Webb, H. Wolfson, W. Chiu, and A. Sali, “Protein structure fitting and refinement guided by cryo-EM density.,” *Structure (London, England : 1993)*, vol. 16, no. 2, pp. 295-307, 2008.

[52] H. J. Wolfson and I. Rigoutsos, “Geometric hashing: an overview,” *IEEE Computational Science and Engineering*, vol. 4, no. 4, pp. 10-21, 1997.

[53] A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson, “Recognition of functional sites in protein structures.,” *Journal of molecular biology*, vol. 339, no. 3, pp. 607-33, 2004.

[54] N. Metropolis and S. Ulam, “The Monte Carlo method.,” *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335-41, Sep. 1949.

[55] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, p. 1087, 1953.

[56] S. M. Stagg et al., “A test-bed for optimizing high-resolution single particle reconstructions.,” *Journal of structural biology*, vol. 163, no. 1, pp. 29-39, Jul. 2008.

[57] K. Ginalski, A. Elofsson, D. Fischer, and L. Rychlewski, “3D-Jury: a simple approach to improve protein structure predictions.,” *Bioinformatics (Oxford, England)*, vol. 19, no. 8, pp. 1015-8, May 2003.

[58] R. Sánchez and A. Sali, “Comparative protein structure modeling. Introduction and practical examples with modeller.,” *Methods in molecular biology (Clifton, N.J.)*, vol. 143, pp. 97-129, Jan. 2000.

[59] S. Lindert, M. Silvestry, T.-M. Mullen, G. R. Nemerow, and P. L. Stewart, “Cryo-electron microscopy structure of an adenovirus-integrin complex indicates conformational changes in both penton base and integrin.,” *Journal of virology*, vol. 83, no. 22, pp. 11491-501, Nov. 2009.

[60] C. Zubieta, G. Schoehn, J. Chroboczek, and S. Cusack, “The structure of the human adenovirus 2 penton.,” *Molecular cell*, vol. 17, no. 1, pp. 121-35, 2005.

[61] J. J. Rux, P. R. Kuser, and R. M. Burnett, “Structural and Phylogenetic Analysis of Adenovirus Hexons by Use of High-Resolution X-Ray Crystallographic, Molecular Modeling, and Sequence-Based Methods,” *Journal of Virology*, vol. 77, no. 17, pp. 9553-9566, Aug. 2003.

[62] K. Braig, P. D. Adams, and A. T. Brünger, “Conformational variability in the refined structure of the chaperonin GroEL at 2.8 A resolution.,” *Nature structural biology*, vol. 2, no. 12, pp. 1083-94, Dec. 1995.

[63] C. L. Lawson et al., “EMDataBank.org: unified data resource for CryoEM.,” *Nucleic acids research*, vol. 39, no. Database issue, pp. D456-64, Jan. 2011.

[64] D. M. Belnap, N. H. Olson, and T. S. Baker, “A method for establishing the handedness of biological macromolecules.,” *Journal of structural biology*, vol. 120, no. 1, pp. 44-51, Oct. 1997.

[65] P. B. Rosenthal and R. Henderson, “Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy.,” *Journal of molecular biology*, vol. 333, no. 4, pp. 721-45, 2003.

[66] W. Jiang, M. L. Baker, S. J. Ludtke, and W. Chiu, “Bridging the information gap: computational tools for intermediate resolution structure interpretation.,” *Journal of molecular biology*, vol. 308, no. 5, pp. 1033-44, May 2001.

[67] L. Urzhumtseva and A. Urzhumtsev, “COMPANG: automated comparison of orientations,” *Journal of Applied Crystallography*, vol. 35, no. 5, pp. 644-647, 2002.

[68] P. Loll, “Membrane protein structural biology: the high throughput challenge,” *Journal of Structural Biology*, vol. 142, no. 1, pp. 144-153, 2003.

[69] F. Alber et al., “Determining the architectures of macromolecular assemblies.,” *Nature*, vol. 450, no. 7170, pp. 683-94, Dec. 2007.

[70] D. I. Svergun, “Small-angle X-ray and neutron scattering as a tool for structural systems biology.,” *Biological chemistry*, vol. 391, no. 7, pp. 737-43, Jul. 2010.

[71] N. Van Eps et al., “Interaction of a G protein with an activated receptor opens the interdomain interface in the alpha subunit.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 23, pp. 9420-4, Jul. 2011.

[72] C. B. Anfinsen, “The formation and stabilization of protein structure.,” *The Biochemical journal*, vol. 128, no. 4, pp. 737-49, Jul. 1972.

[73] J. W. Ponder and D. A. Case, “Force fields for protein simulations,” vol. 66, 2003, p. 27-+.

[74] J. Novotný, R. Bruccoleri, and M. Karplus, “An analysis of incorrectly folded protein models. Implications for structure predictions.,” *Journal of molecular biology*, vol. 177, no. 4, pp. 787-818, Aug. 1984.

[75] S. Miyazawa and R. L. Jernigan, “Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation,” *Macromolecules*, vol. 18, no. 3, pp. 534-552, May 1985.

[76] D. T. Jones, W. R. Taylor, and J. M. Thornton, “A new approach to protein fold recognition.,” *Nature*, vol. 358, no. 6381, pp. 86-9, Jul. 1992.

[77] M.-Y. Shen and A. Sali, “Statistical potential for assessment and prediction of protein structures.,” *Protein science : a publication of the Protein Society*, vol. 15, no. 11, pp. 2507-24, Nov. 2006.

[78] T. Hamelryck et al., “Potentials of mean force for protein structure prediction vindicated, formalized and generalized.,” *PloS one*, vol. 5, no. 11, p. e13714, Jan. 2010.

[79] C. A. Rohl, C. E. M. Strauss, D. Chivian, and D. Baker, “Modeling structurally variable regions in homologous proteins with rosetta.,” *Proteins*, vol. 55, no. 3, pp. 656-77, May 2004.

[80] A. A. Canutescu and R. L. Dunbrack, “Cyclic coordinate descent: A robotics algorithm for protein loop closure.,” *Protein science : a publication of the Protein Society*, vol. 12, no. 5, pp. 963-72, May 2003.

[81] D. J. Mandell, E. A. Coutsias, and T. Kortemme, “Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling.,” *Nature methods*, vol. 6, no. 8, pp. 551-2, Aug. 2009.

[82] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack, “Improved prediction of protein side-chain conformations with SCWRL4.,” *Proteins*, vol. 77, no. 4, pp. 778-95, Dec. 2009.

[83] K. W. Kaufmann, G. H. Lemmon, S. L. Deluca, J. H. Sheehan, and J. Meiler, “Practically useful: what the Rosetta protein modeling suite can do for you.,” *Biochemistry*, vol. 49, no. 14, pp. 2987-98, May 2010.

[84] E. Durham, B. Dorr, N. Woetzel, R. Staritzbichler, and J. Meiler, “Solvent accessible surface area approximations for rapid and accurate protein structure prediction.,” *Journal of molecular modeling*, vol. 15, no. 9, pp. 1093-108, Oct. 2009.

[85] J. Hsin, A. Arkhipov, Y. Yin, J. E. Stone, and K. Schulten, “Using VMD: an introductory tutorial.,” *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, vol. 5, p. Unit 5.7, Dec. 2008.

[86] D. N. Ivankov, S. O. Garbuzynskiy, E. Alm, K. W. Plaxco, D. Baker, and A. V. Finkelstein, “Contact order revisited: influence of protein size on the folding rate.,” *Protein science : a publication of the Protein Society*, vol. 12, no. 9, pp. 2057-62, Sep. 2003.

[87] P. Flory, *Principles of Polymer Chemistry*. Cornell University Press, 1953.

[88] S. T. Rao and M. G. Rossmann, “Comparison of super-secondary structures in proteins.,” *Journal of molecular biology*, vol. 76, no. 2, pp. 241-56, May 1973.

[89] C. Chothia, M. Levitt, and D. Richardson, “Helix to helix packing in proteins.,” *Journal of molecular biology*, vol. 145, no. 1, pp. 215-50, Jan. 1981.

[90] Z. H. Zhou, “Towards atomic resolution structural determination by single-particle cryo-electron microscopy.,” *Current opinion in structural biology*, vol. 18, no. 2, pp. 218-28, Apr. 2008.

[91] C. S. Klug and J. B. Feix, “Methods and applications of site-directed spin labeling EPR spectroscopy.,” *Methods in cell biology*, vol. 84, pp. 617-58, Jan. 2008.

[92] G. Wang and R. L. Dunbrack, “PISCES: recent improvements to a PDB sequence culling server.,” *Nucleic acids research*, vol. 33, no. Web Server issue, pp. W94-8, Jul. 2005.

[93] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.,” *Biopolymers*, vol. 22, no. 12, pp. 2577-637, Dec. 1983.

[94] H. M. Berman et al., “The Protein Data Bank.,” *Acta crystallographica. Section D, Biological crystallography*, vol. 58, no. Pt 6 No 1, pp. 899-907, Jun. 2002.

[95] S. Dutta and H. M. Berman, “Large macromolecular complexes in the Protein Data Bank: a status report.,” *Structure (London, England : 1993)*, vol. 13, no. 3, pp. 381-8, 2005.

[96] P. R. Daga, R. Y. Patel, and R. J. Doerksen, “Template-based protein modeling: recent methodological advances.,” *Current topics in medicinal chemistry*, vol. 10, no. 1, pp. 84-94, Jan. 2010.

[97] R. C. Stevens, S. Yokoyama, and I. A. Wilson, “Global efforts in structural genomics.,” *Science (New York, N.Y.)*, vol. 294, no. 5540, pp. 89-92, Oct. 2001.

[98] S. A. Lesley et al., “Structural genomics of the Thermotoga maritima proteome implemented in a high-throughput structure determination pipeline.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 18, pp. 11664-9, Sep. 2002.

[99] F. DiMaio et al., “Improved molecular replacement by density- and energy-guided protein structure optimization.,” *Nature*, vol. 473, no. 7348, pp. 540-3, May 2011.

[100] R. M. Bill et al., “Overcoming barriers to membrane protein structure determination.,” *Nature biotechnology*, vol. 29, no. 4, pp. 335-40, Apr. 2011.

[101] A. Oberai, Y. Ihm, S. Kim, and J. U. Bowie, “A limited universe of membrane protein families and folds.,” *Protein science : a publication of the Protein Society*, vol. 15, no. 7, pp. 1723-34, Jul. 2006.

[102] S. Yooseph et al., “The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families.,” *PLoS biology*, vol. 5, no. 3, p. e16, Mar. 2007.

[103] K. Karplus et al., “Predicting protein structure using hidden Markov models.,” *Proteins*, vol. 1, pp. 134-9, Jan. 1997.

[104] J. Meiler and D. Baker, “Coupled prediction of protein secondary and tertiary structure.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12105-10, Oct. 2003.

[105] J. J. Ward, L. J. McGuffin, B. F. Buxton, and D. T. Jones, “Secondary structure prediction with support vector machines,” *Bioinformatics*, vol. 19, no. 13, pp. 1650-1655, Sep. 2003.

[106] M. Kuhn, J. Meiler, and D. Baker, “Strand-loop-strand motifs: prediction of hairpins and diverging turns in proteins.,” *Proteins*, vol. 54, no. 2, pp. 282-8, Feb. 2004.

[107] D. T. Jones and J. J. Ward, “Prediction of disordered regions in proteins from position specific score matrices.,” *Proteins*, vol. 53 Suppl 6, pp. 573-8, Jan. 2003.

[108] R. Linding, L. J. Jensen, F. Diella, P. Bork, T. J. Gibson, and R. B. Russell, “Protein disorder prediction: implications for structural proteomics.,” *Structure (London, England : 1993)*, vol. 11, no. 11, pp. 1453-9, Nov. 2003.

[109] O. Graña et al., “CASP6 assessment of contact prediction.,” *Proteins*, vol. 61 Suppl 7, pp. 214-24, Jan. 2005.

[110] J. Liu and B. Rost, “Comparing function and structure between entire proteomes.,” *Protein science : a publication of the Protein Society*, vol. 10, no. 10, pp. 1970-9, Oct. 2001.

[111] O. V. Galzitskaya and B. S. Melnik, “Prediction of protein domain boundaries from sequence alone.,” *Protein science : a publication of the Protein Society*, vol. 12, no. 4, pp. 696-701, Apr. 2003.

[112] D. E. Kim, D. Chivian, L. Malmström, and D. Baker, “Automated prediction of domain boundaries in CASP6 targets using Ginzu and RosettaDOM.,” *Proteins*, vol. 61 Suppl 7, pp. 193-200, Jan. 2005.

[113] A. Valencia and F. Pazos, “Computational methods for the prediction of protein interactions.,” *Current opinion in structural biology*, vol. 12, no. 3, pp. 368-73, Jun. 2002.

[114] A. Ben-Hur and W. S. Noble, “Kernel methods for predicting protein-protein interactions.,” *Bioinformatics (Oxford, England)*, vol. 21 Suppl 1, pp. i38-46, Jun. 2005.

[115] P. Bradley et al., “Free modeling with Rosetta in CASP6.,” *Proteins*, vol. 61 Suppl 7, pp. 128-34, Jan. 2005.

[116] H. Zhou, S. B. Pandit, and J. Skolnick, “Performance of the Pro-sp3-TASSER server in CASP8.,” *Proteins*, vol. 77 Suppl 9, pp. 123-7, Jan. 2009.

[117] H. Zhou and J. Skolnick, “Ab initio protein structure prediction using chunk-TASSER.,” *Biophysical journal*, vol. 93, no. 5, pp. 1510-8, Sep. 2007.

[118] J. Zimmer, Y. Nam, and T. A. Rapoport, “Structure of a complex of the ATPase SecA and the protein-translocation channel.,” *Nature*, vol. 455, no. 7215, pp. 936-43, Oct. 2008.

[119] B. L. Sibanda, D. Y. Chirgadze, and T. L. Blundell, “Crystal structure of DNA-PKcs reveals a large open-ring cradle comprised of HEAT repeats.,” *Nature,* vol. 463, no. 7277, pp. 118-21, Jan. 2010.

[120] L. Skrisovska, M. Schubert, and F. H.-T. Allain, “Recent advances in segmental isotope labeling of proteins: NMR applications to large proteins and glycoproteins.,” *Journal of biomolecular NMR*, vol. 46, no. 1, pp. 51-65, Jan. 2010.

[121] B. Qian et al., “High-resolution structure prediction and the crystallographic phase problem.,” *Nature*, vol. 450, no. 7167, pp. 259-64, Nov. 2007.

[122] S. Raman et al., “NMR structure determination for larger proteins using backbone-only data.,” *Science (New York, N.Y.)*, vol. 327, no. 5968, pp. 1014-8, Feb. 2010.

[123] B. I. Dahiyat and S. L. Mayo, “De novo protein design: fully automated sequence selection.,” *Science (New York, N.Y.)*, vol. 278, no. 5335, pp. 82-7, Oct. 1997.

[124] B. Kuhlman and D. Baker, “Native protein sequences are close to optimal for their structures.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 19, pp. 10383-8, Sep. 2000.

[125] R. L. Dunbrack, “Rotamer libraries in the 21st century.,” *Current opinion in structural biology,* vol. 12, no. 4, pp. 431-40, Aug. 2002.

[126] A. Sali and T. L. Blundell, “Comparative protein modelling by satisfaction of spatial restraints.,” *Journal of molecular biology,* vol. 234, no. 3, pp. 779-815, Dec. 1993.

[127] D. Baker, “A surprising simplicity to protein folding.,” *Nature,* vol. 405, no. 6782, pp. 39-42, May 2000.

[128] V. Grantcharova, E. J. Alm, D. Baker, and A. L. Horwich, “Mechanisms of protein folding.,” *Current opinion in structural biology,* vol. 11, no. 1, pp. 70-82, Feb. 2001.

[129] R. Bonneau, I. Ruczinski, J. Tsai, and D. Baker, “Contact order and ab initio protein structure prediction.,” *Protein science : a publication of the Protein Society,* vol. 11, no. 8, pp. 1937-44, Aug. 2002.

[130] B. Rost, C. Sander, and R. Schneider, “Redefining the goals of protein secondary structure prediction.,” *Journal of molecular biology,* vol. 235, no. 1, pp. 13-26, Jan. 1994.

[131] O. Carugo and S. Pongor, “A normalized root-mean-square distance for comparing protein three-dimensional structures.,” *Protein science : a publication of the Protein Society,* vol. 10, no. 7, pp. 1470-3, Jul. 2001.

[132] J. Moult, “A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction.,” *Current opinion in structural biology,* vol. 15, no. 3, pp. 285-9, Jun. 2005.

[133] S. F. Altschup, W. Gish, T. Pennsylvania, and U. Park, "Basic Local Alignment Search Tool 2Department of Computer Science," *Methods*, pp. 403-410, 1990.

[134] S. F. Altschul et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.," *Nucleic acids research*, vol. 25, no. 17, pp. 3389-402, Sep. 1997.