Deep Learning and Physics-Based Methods for Macromolecular Structure Prediction and Design

by

Robin Pearce

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in the University of Michigan
2023

Doctoral Committee:

Professor Gilbert S. Omenn, Chair
Professor Charles L. Brooks III
Associate Professor Peter L. Freddolino
Assistant Professor Tobias Giessen
Research Assistant Professor Matthew O'Meara
Professor Janet Smith

Robin Pearce

robpearc@umich.edu

ORCID iD: 0000-0001-6402-734X

ACKNOWLEDGEMENTS

I would like to thank my mentor Dr. Yang Zhang for his invaluable guidance and support. I appreciate all he has invested in me and for encouraging me to explore a wide array of scientific problems. I have truly enjoyed my time in the lab and working under his leadership.

I would like to extend my sincerest gratitude to Dr. Gil Omenn. I appreciate his kindness and tremendous guidance during my PhD studies. This dissertation would not have been possible without his support.

I would like to thank my committee members, Drs. Charles Brooks, Janet Smith, Peter Freddolino, Tobias Giessen, and Matthew O'Meara for their time, feedback, and inciteful discussions.

I would like to thank all the Zhang Lab members, whom I've had the pleasure of working with. I would like to especially thank Xiaoqiang Huang, Wei Zheng, Yang Li, and Chengxin Zhang, who have served as close collaborators on numerous projects.

Finally, I would like to thank my family for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Proteins and non-coding RNA are the macromolecules responsible for performing the vast majority of biological functions in living organisms. These functions are mediated by the diverse structures adopted by different macromolecules, which in turn are determined by their primary sequences. Understanding the principles that govern this sequence-structure-function paradigm has become a hallmark of structural biology. The work presented in this thesis focuses on elucidating these principles by developing state-of-the-art deep learning and physical models for computational protein/RNA structure prediction and protein design.

Despite the immense progress witnessed in protein structure prediction through the use of deep neural networks to predict spatial restraints, the modeling accuracy for proteins that lacked sequence and/or structure homologs remained to be improved. Thus, we developed an open-source program, DeepFold, which integrates spatial restraints predicted by multi-task deep residual neural networks along with a physics-based potential to guide rapid gradient-descent folding simulations. The results on large-scale benchmark tests showed that DeepFold created full-length models with accuracies significantly beyond classical folding approaches and other leading, contemporaneous deep learning methods. Of particular interest was the modeling performance on targets with very few homologous sequences, where DeepFold achieved an average TM-score that was ~40-45% higher than deep learning methods such as trRosetta and DMPfold, while being 262 times faster than traditional folding simulations.

Inspired by the revolutionary advances in self-attention-based structure prediction, we developed DeepFoldRNA, which is an extension of the DeepFold pipeline that predicts RNA structures from sequence by coupling deep self-attention neural networks with gradient-based folding simulations. The method was tested on two independent benchmark datasets, including the RNA-Puzzles experiment, where DeepFoldRNA constructed models with an average RMSD of 2.72 Å, which was significantly better than the best models submitted by the community

(RMSD=6.90 Å). Overall, these findings illustrate the major advantage of advanced deep learning techniques at capturing detailed structural information over human-engineered potentials.

The second area of research that will be covered in the proceeding chapters is protein design, which is often regarded as the conceptual inverse of protein structure prediction. Protein design generally consists of two sub-problems, namely sequence design and structure design. For the first sub-problem, we developed an online server system, EvoDesign, which uses evolutionary profiles alongside a physical potential to guide the sequence search simulations. EvoDesign demonstrated advantages over pure physics-based approaches in terms of more accurately designing proteins that adopt desired target folds. Furthermore, as one of the essential difficulties in computer-based protein design is the expensive cost of experimental validation, the server aims to provide various transparent intermediate data to allow for a detailed annotation and analysis of the confidence of the designed sequences.

Lastly, for the second design sub-problem, we developed FoldDesign to create novel protein folds from specific secondary structure (SS) assignments through sequence-independent replica-exchange Monte Carlo simulations. The method was tested on a large-scale dataset of non-idealized, SS topologies, where FoldDesign outperformed other state-of-the-art methods and consistently created stable structural folds with local characteristics that closely matched native structures. Notably, while sharing similar local characteristics, a large portion of the designed scaffolds possessed novel global folds that were completely different from natural proteins in the PDB. This highlights FoldDesign's ability to explore areas of protein fold space through computational simulations that have not been explored by nature.

CHAPTER 1

Introduction

Proteins and non-coding RNA are the macromolecules that are nearly ubiquitously responsible for carrying out the unique and varied functions necessary to sustain life. These diverse functions are made possible by the unique three-dimensional structures adopted by different molecules. The landmark study by Anfinsen in the 1970s demonstrated that tertiary structure is dictated by primary sequence (1), and since then, understanding the sequence-structure-function paradigm has become a cornerstone of modern biomedical studies. Among the most accurate experimental methods for determining the structures of macromolecules are X-ray crystallography (2), NMR spectroscopy (3), and cryo-electron microscopy (4). However, due to the significant human effort and expenses required for experimental structure determination, the growth in the number of solved structures has lagged far behind the accumulation of sequence data. So far, the structures of approximately 0.18 million proteins and less than 0.01 million RNA have been deposited in the Protein Data Bank (5) (PDB), which accounts for ~0.08% of the 230 million protein sequences in the UniProt database (6) and ~0.03% of the 34 million non-coding RNA sequences in RNAcentral (7). Therefore, it is apparent that there is a large gap between the number of known sequences, which by themselves provide only limited functional insight, and the number of experimentally solved structures.

Nevertheless, due to the tremendous effort made by the community over the last few decades (8-22), an increasing portion of the genes in organisms have had their tertiary structures reliably modeled by computational approaches (23-29). In addition, high-quality structural models are created every day by online structure prediction systems (15, 16, 20, 22, 30-36). These models have been used to assist various biomedical studies, including structure-based protein function annotation (37-41), mutation analysis (42-49), ligand screening (50-57), and drug discovery (58-63). Thus, the development of high-accuracy structure prediction methodologies represents perhaps the most promising, yet challenging approach to address the disparity between the number of known sequences and experimentally solved structures, while also elucidating the fundamental principles that govern the sequence-structure-function paradigm.

Despite the impressive role of natural molecules such as proteins, only a tiny portion of the total possible amino acid sequences and structures appear in nature, which is most likely due to the selective pressures exerted by environmental constraints upon organisms (64). For example, there have been just under 1,500 protein folds classified in the SCOPe database (65) and the evidence indicates that, for proteins, the current PDB is nearly complete, representing the vast majority of natural folds (66, 67). Thus, computational protein design, which aims to create artificial proteins tailored to specific design applications, is a thorough test of our understanding of the principles that underly the folding paradigm. To date, computational design approaches have been applied to create proteins with promising therapeutic potential (68-70), novel ligand-binding activity (71, 72), and complex logical interactions (73). Thus, given the importance of these problems, the remaining sections of this chapter will cover the fields of protein/RNA structure prediction and protein design in more depth, with a particular emphasis on the impact brought about by deep learning for structure prediction and recent progress in *de novo* protein design.

1.1 Protein and RNA Structure Prediction

The goal of protein and RNA structure prediction is to use computational methods to determine the spatial location of every atom in a given molecule starting from its primary sequence. Depending on whether a template structure is used, structure prediction approaches can be generally categorized as either template-based modeling (TBM) or template-free modeling (FM) methods. While TBM constructs models by copying and refining structural frameworks of evolutionarily related protein/RNA molecules, called templates, identified from the PDB, FM aims to predict structures without using global template information. FM methods have also been referred to as *ab initio* or *de novo* modeling approaches. A general pipeline that illustrates the key steps involved in TBM and FM methods is depicted in Fig. 1.1.

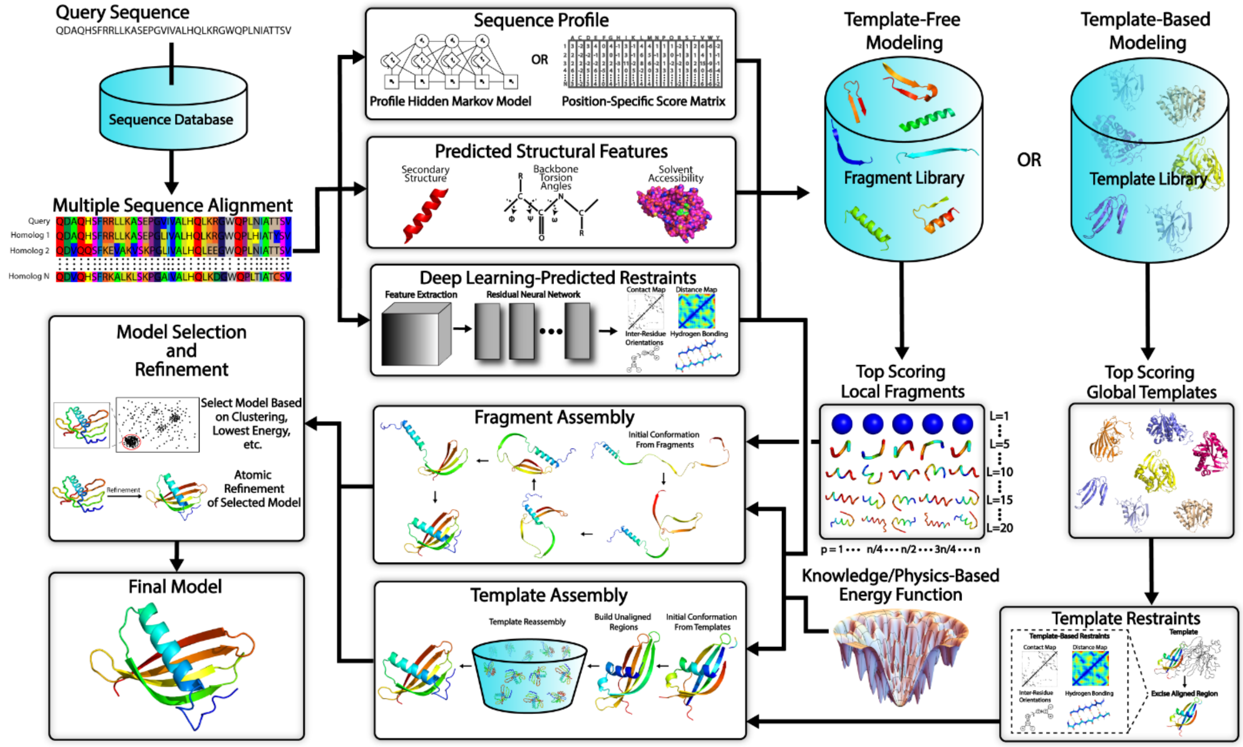

Query Sequence
QDAQHSFRRLLKASEPGVIVALHQLKRGWQPLNIATTSV
Sequence Database
Multiple Sequence Alignment
Query
Homolog 1
Homolog 2
Homolog N
Sequence Profile
Profile Hidden Markov Model
OR
Position-Specific Score Matrix
Predicted Structural Features
Secondary Structure
Backbone Torsion Angles
Solvent Accessibility
Deep Learning-Predicted Restraints
Feature Extraction
Residual Neural Network
Contact Map
Distance Map
Inter-Residue Orientations
Hydrogen Bonding
Template-Free Modeling
Fragment Library
OR
Template-Based Modeling
Template Library
Top Scoring Local Fragments
L=1
L=5
L=10
L=15
L=20
p = 1 ••• n/4 ••• n/2 ••• 3n/4 ••• n
Top Scoring Global Templates
Model Selection and Refinement
Select Model Based on Clustering, Lowest Energy, etc.
Refinement
Atomic Refinement of Selected Model
Final Model
Fragment Assembly
Initial Conformation From Fragments
Template Assembly
Template Reassembly
Build Unaligned Regions
Initial Conformation From Templates
Knowledge/Physics-Based Energy Function
Template Restraints
Template-Based Restraints
Contact Map
Distance Map
Inter-Residue Orientations
Hydrogen Bonding
Template
Excise Aligned Region

Figure 1.1 Typical steps involved in template-free and template-based protein structure prediction approaches. Starting from a query sequence, a multiple sequence alignment (MSA) is generated by identifying homologous sequences from a sequence database. The MSA is then converted into a sequence profile and is also used to predict several structural features such as the secondary structure, backbone torsion angles and solvent accessibility. For fragment assembly-based template-free modeling methods, these structural features together with the sequence profile are used to search a fragment library in order to identify high scoring local fragments. For template-based modeling methods, they are used by threading protocols to identify global template structures. Meanwhile, co-evolutionary information is extracted from the MSA and fed into a deep residual neural network in order to predict important spatial restraints such as inter-residue contacts, distances, hydrogen bonds and torsion angles. For full-length model construction, structure assembly simulations are performed under the guidance of a composite force field which usually combines the generic knowledge- and/or physics-based energy function with deep neural network feature prediction (plus template-based restraints in the case of template-based modeling). Finally, representative models are selected typically from the lowest energy conformations or based on structural clustering, followed by atomic-level refinement to generate the final model.

1.1.1 Classical Approaches to Template Based Modeling

There are four key steps involved in TBM methods: (1) identification of experimentally solved structures (templates) that are related to the protein/RNA to be modeled, (2) alignment of the protein/RNA of interest (query) and the templates, (3) construction of initial structural frameworks by copying the aligned regions of the template structure, and (4) construction of the unaligned regions and refinement of the global fold.

Depending on the evolutionary distance between the query and template, TBM has been historically divided into comparative modeling (CM, see Fig. A.1 in Appendix A) and threading (see Fig. 1.1). CM is designed for targets with close homologous templates where the templates can typically be identified by sequence-based alignment, while threading is designed for detecting more distantly homologous templates by combining sequence profiles and/or Hidden Markov Model (HMM) alignments with local structure feature prediction (74-76). With the progress in the field, the difference between CM and threading has become increasingly blurred, especially for protein structure prediction, and most of the modern TBM approaches start with templates identified by advanced threading programs. Since different threading programs are trained with different scoring function and alignment algorithms, the template recognition and alignment results are often diverse for the same query sequence. This has resulted in the prevalence of meta-threading programs (77, 78), which collect and combine template alignments from a set of complementary threading algorithms. While rigorous theoretical studies to explain the consistent improvement brought about by combining multiple structures were not available until many years later (79), the intuition behind the usage of multiple threading templates is simple. Given that there are many more ways for threading to generate incorrect alignments than to generate a correct alignment, it is much easier to get a consensus correct alignment than multiple consistent but incorrect alignments (80).

Since threading templates only provide gapped traces, which have no practical use for detailed function annotation and/or virtual ligand screening, many programs have been developed to assemble and refine full-length atomic structural models starting from the template alignments. Among the methods for protein-specific TBM, MODELLER (13) was one of the first programs and builds atomic models by optimally satisfying spatial restraints derived from a threading alignment, where the restraints are expressed as probability density functions for the restrained features. One of the most successful classical TBM methods is I-TASSER (17), which is an extension of TASSER (25). I-TASSER has been consistently ranked as the top automated method in the community-wide Critical Assessment of Structure Prediction (CASP) experiments, where the goal of CASP is to benchmark the state of the art in protein structure prediction (81, 82). In the I-TASSER pipeline, continuous fragments are excised from the template alignments and reassembled through replica-exchange Monte Carlo (REMC) simulations, where the unaligned regions (mainly loops) are built *ab initio* using a lattice-based system in junction with the aligned fragments. One of the key reasons for the success of I-TASSER is its effective combination of multiple threading templates (often more than 20-50) under the guidance of an optimal knowledge-based force field whose parameters were extensively optimized using large-scale structural decoys (83). Following a similar idea, RosettaCM was developed which assembles global structural folds by recombining aligned segments of threading templates and building unaligned regions *de novo* in torsion space using gradient-based minimization (84).

In comparison to protein structure prediction, the field of RNA structure prediction has witnessed considerably less progress in TBM modeling (85, 86). This is in part due to the fact that, compared to proteins, there is much less structural information present in the PDB for RNA, which makes template-based modeling less effective. For example, of the 4,192 Rfam RNA families, only 99 families have solved structures (87). This is drastically different from protein structure prediction, where the PDB is nearly complete, representing the vast majority of single domain protein folds (66, 67). Nevertheless, there has been some work in template-based RNA modeling. For example, ModeRNA, which is mainly used for CM, copies the coordinates from the aligned region of a template and rebuilds the unaligned regions using a cyclical coordinate descent algorithm guided by a knowledge-based potential to ensure proper loop closure (88). Other RNA-specific TBM methods, such as RNAbuilder (89), combine restraints from multiple template structures with a physical potential that accounts for factors such as steric clashes and base-pairing. Structural minimization is then performed in torsion angle space, where the RNA bond lengths and angles are kept fixed, and the dihedral angles are optimized with respect to the template- and physics-based potential.

1.1.2 Classical Approaches to Template-Free Modeling

Unlike TBM, FM approaches predict structures without the use of global template information. One of the most effective methods for constructing FM models is fragment assembly, where the idea was pioneered by Bowie and Eisenberg in 1994 for protein structure prediction (90). More modern protein fragment assembly approaches include Rosetta (14) and QUARK (16). These methods first identify local fragments, ranging from 1-20 residues long, from solved protein structures based on the profile-profile similarity and comparison of the local structural features such as secondary structure, solvent accessibility and torsion angles, either predicted for the query or extracted from the templates (16). In the next step of the fragment assembly simulations, the backbone torsion angles for a specific region of the simulated structure are replaced with those from a selected fragment, either assuming ideal bond lengths and angles (14), or directly taking these from the fragments themselves (16). Loop closure may also be used, which adjusts the torsion angles around the substitution site in order to prevent large conformational changes downstream (91). The rationale for constructing models through fragment assembly is two-fold: it reduces the size of the conformational search space, while ensuring the local structures of models are well formed as the fragments are selected from experimentally determined structures, which can help compensate for inaccuracies in the energy functions used for modeling. To improve the efficiency of conformational sampling, Rosetta (14) uses simulated annealing Monte Carlo simulations, while QUARK (16) uses REMC simulations with as many as 11 different conformational moves and extracts distance-profile restraints from the generated fragments in order to guide the simulations towards the native structure (92).

Fragment assembly has also been among the most popular methods for RNA structure prediction given the lack of template information in the PDB for RNA molecules. For example, FARNA (93), which was introduced in 2007 and later extended to FARFAR (94), is an extension of the Rosetta fragment assembly protocol, where the altered protocol includes an RNA-specific fragment library as well as base-pairing and stacking potentials derived from PDB statistics. The core procedure of FARFAR is similar to the procedure for Rosetta-based protein structure prediction, where small, evolutionarily related structural fragments are identified from a fragment library and assembled during the Monte Carlo folding simulations. Other successful fragment assembly-based approaches to RNA structure modeling include methods such as 3dRNA (95), RNAComposer (96), and VfoldLA (97).

Fragment assembly methods have consistently been among the top performers in the FM section of the CASP experiment as well as the RNA-Puzzles challenge by successfully folding protein and RNA targets that lack identifiable homology templates (86, 98-101). Here, the RNA-Puzzles challenge is similar to the CASP experiment in that it is a blind modeling challenge whose aim is to identify the top RNA tertiary structure prediction methods (85, 86, 101, 102). Despite the success, the Monte Carlo simulation-based fragment assembly process can be time-consuming compared to TBM approaches, since FM methods need to create models starting from random conformations. These computational limitations also impose further restrictions on the energy functions, which typically use coarse-grained representations that account for only a fraction of the atoms that make up each residue.

1.1.3 Early Effort in Inter-residue Contact Prediction to Assist FM Approaches

Given the inability of threading-based methods to reliably identify high-quality templates for many targets as well as the sampling and physical energy function limitations, an additional source of information was needed to guide structure prediction approaches, particularly for FM targets. Thus, the use of statistical models and machine learning methods to predict pairwise spatial restraints has become a major area of research in the field. This is because the tertiary structures of proteins and RNA are formed and stabilized by interactions between the atoms that make up each residue, and prediction of these interactions provides extremely useful information that can guide modeling approaches. Initially, these pairwise spatial restraints took the form of contact map prediction, where a contact map for a protein or RNA with length L is a symmetric, binary $L \times L$ matrix and each element of the matrix indicates whether the distance between two residues falls below a specific cutoff (typically <8Å).

One of the earliest sequence-based contact prediction methods used correlated mutations observed in multiple sequence alignments (MSAs) to predict inter-residue contact maps (103). Here, an MSA is an alignment of sequences that are evolutionarily related or share sequence homology to a given query sequence (104). The hypothesis behind the approach was that if mutations that occur at two positions in an MSA are correlated, these positions are more likely to form a contact in 3D space (105). This is because there is evolutionary pressure to conserve the structures of proteins/RNA and a mutation at one position may be rescued by a corresponding mutation at a nearby residue. The accuracy of co-evolution-based contact map prediction remained low for many years due to the inability to distinguish between direct and indirect interactions (106, 107), where indirect interactions occur when residues appear to co-evolve but do not actually form contacts. For example, if Residues A and B are both in contact with Residue C, A and B often appear as if they co-evolve even when there is no physical contact between them. There is evidence that shows such co-evolution may have a functional cause rather than a structural one, which resulted in the failure of structure-based contact derivation (108).

Progress in contact prediction remained stagnant for some time. However, a leap in contact prediction accuracy took place when algorithms started utilizing global prediction approaches.

Early methods mainly predicted contacts between residue pairs one-at-a-time using techniques such as mutual information, thus ignoring the interactions with other residue pairs and the global context in which the interactions took place (109). This was largely why it was difficult for these local methods to distinguish between direct and indirect interactions. The introduction of global statistical models determined through the use of direct coupling analysis (DCA) was more successfully able to distinguish between these direct and indirect interactions (106, 107). The goal of such global statistical models is to determine the set of direct interactions that most harmoniously accounts for the observed sequence co-variation by simultaneously considering the entire set of pairwise interactions. Since all pairwise interactions are simultaneously considered, instead of just considering one interaction at a time and ignoring the global context in which the interactions take place, DCA was able to significantly improve the contact prediction accuracy.

Many DCA techniques fit a Markov random field (MRF), or more specifically a Potts model, to an MSA (106, 107, 110, 111). An MRF is a graphical model that represents each column of an MSA as a node that describes the distribution of residues at a given position, where the edges between nodes indicate the joint distributions of residues between each pair of positions. The couplings or co-evolutionary parameters can then be determined from the edge weights. Since fitting an MRF model using its actual likelihood function is computationally intractable due to the need to calculate the partition function, various approximations have been developed including those based on message passing (106), Gaussian approximation (111), mean-field approximation (107), and pseudo-likelihood maximization (110). Another popular method was introduced by PSICOV (112), which determines the coupling parameters by estimating the inverse covariance matrix or precision matrix using a graphical LASSO penalty (L1 regularization) instead of directly fitting an MRF model to an MSA. This was later extended by ResPRE (113), where the inverse covariance matrix is estimated using L2 regularization instead of L1 regularization. Network deconvolution has also been used to distinguish direct from indirect interactions determined from co-evolutionary data (114).

1.1.4 Accurate Restraint Prediction through Deep Residual Neural Networks

Although encouraging progress in contact prediction was made by DCA, the accuracy remained unsatisfactory in many cases, particularly for targets with few homologous sequences and shallow alignment depths (115). However, a breakthrough in protein contact map prediction came in 2017 when Xu's group proposed RaptorX-Contact (19), which reformulated the contact prediction problem through the introduction of deep residual convolutional neural networks (ResNets (116)). Here, a ResNet is a convolutional neural network that adds an identity map of the input to the output of the convolutional layers, allowing gradients to flow smoothly from deeper to shallower layers and enabling the training of deep networks with many layers. Under this framework, the contact map prediction problem is considered an image segmentation task, i.e., a pixel-level labeling problem, where the whole contact map is an image in which each residue pair corresponds to a pixel. Image segmentation is a task for which ResNets, originally developed for computer vision, have demonstrated excellent performance. While the features used by RaptorX-Contact, such as co-evolutionary information obtained through DCA, predicted secondary structures, and PSSMs, were quite similar to other predictors, the introduction of deep ResNets with approximately 60 hidden layers enabled RaptorX-Contact to dramatically outperform other methods**.** The approach introduced by RaptorX-Contact was adapted by methods such as ResPRE (113) and TripletRes (117), which used a similar deep learning architecture but with a unique set of features that included multiple co-evolutionary coupling matrices.

Similar ResNets were later extended to multi-class distance prediction, which predicts the binned distance between two residues as opposed to a binary contact value (36). The power of distance map-guided folding was convincingly demonstrated by AlphaFold in the CASP13 experiment, in which the program utilized an ultra-deep neural network composed of 220 ResNet blocks to predict distance maps for a query sequence (118). The distance maps were then used to guide rapid gradient descent-based folding simulations (118).

The success of deep learning contact and distance map prediction raised the question of what other restraints could be accurately predicted using deep learning. As structure modelers have known for years that knowledge-based energy functions that are dependent only on residue-residue distances are often not as accurate as those that use both distances and orientations (119), a natural extension of distance prediction was inter-residue orientation prediction. Orientation-dependent energy functions are important as certain types of residue-residue interactions require not only distance proximity but also specific orientations between the residue pairs, e.g. β-strand pairing. Furthermore, it is not possible to uniquely determine the geometry of a structure without orientation information, as distance information alone cannot differentiate between a pair of mirrored structures. Orientation prediction in a deep learning framework was introduced by

NEMO (120) and later refined by trRosetta, which simultaneously predicts both pairwise residue distances and inter-residue orientations from co-evolutionary features using a unified deep ResNet (22). As will be discussed in Chapter 2, inspired by these advances, we developed DeepPotential/DeepFold, which predicts an ensemble of contact, distance, orientation and hydrogen bonding maps and converts these into a deep learning-based potential that is minimized using rapid gradient-based folding simulations (121, 122). This approach was found to be highly effective at modeling non-homologous protein targets in the CASP14 experiment and independent benchmark analyses (121, 122).

1.1.5 Highly Accurate Protein Structure Prediction by AlphaFold2

The most exciting progress to date in the field of protein structure prediction was recently brought about by AlphaFold2 (123), the second iteration of AlphaFold developed by DeepMind, which achieved unprecedented modeling accuracy in the CASP14 experiment. Compared to the first iteration of AlphaFold in CASP13, which was driven by convolutional neural network-based distance map prediction, one of the major advancements of AlphaFold2 is the incorporation of a self-attention-based neural network architecture known as the Transformer. Transformers are a novel machine learning architecture that was introduced in 2017 by Google, and have significantly impacted the field of natural language processing, outperforming recurrent and convolutional networks (124). Briefly, transformers pass inputs through a series of self-attention and feedforward connections, which allow the network to attend to relevant information from the input and build up complex representations that incorporate long-range dependencies. Moreover, instead of using gradient-descent optimization to construct models based on the predicted distance restraints, as AlphaFold did in CASP13, AlphaFold2 utilizes a full end-to-end training system from sequence to structure using iterative structural refinement. As part of this, the system replaces traditional folding simulations with a structure module composed of 3D equivariant transformer neural networks, which treat each amino acid as a gas of 3D rigid bodies and allows for the direct generation of structure models.

The accuracy of AlphaFold2 was convincingly demonstrated in CASP14, where it dramatically outperformed all other methods. As evidence of this, Fig. 1.2.A depicts the historical modeling results from CASP7 and CASP11-14 on FM and TBM targets in terms of the mean TM-scores of the best first submitted model for each target. Here, TM-score is sequence length independent metric that ranges from (0, 1], where a score >0.5 indicates the predicted and native structures share the same global topology and a score >0.914 may be used as a cutoff for low-to-medium resolution experimental accuracy (125). From Fig. 1.2.A, it can be seen that in the eight years from CASP7-11 the average TM-score on FM targets improved slowly from 0.38 to 0.47. However, with the wide-spread adoption of DCA and shallow neural networks in CASP12 and deep ResNets in CASP13 for restraint prediction, the modeling accuracy for FM targets improved significantly from 0.47 to 0.65 over the span of four years. Notably, the accuracy on TBM targets remained largely stagnant during this time (TM-score 0.80-0.83).

In CASP14, most top predictors used deep ResNets to predict distance and orientation maps, which were then used to guide the folding simulations, where the average performance of the best submitted models for FM and TBM targets improved to 0.69 and 0.84, respectively. However, AlphaFold2 alone was able to achieve an average TM-score of 0.84 for FM targets and 0.93 for TBM targets. From Figs. 1.2.B-C, we can also see a marked increase in the number of models produced with experimental accuracy when considering a cutoff TM-score of 0.914 (125). In previous CASP experiments, none of the FM targets could be folded with such high accuracy, but in CASP14, AlphaFold2 was able to fold ~33% of the FM targets with experimental accuracy, and almost 80% of the TBM targets. Thus, AlphaFold2 was able to produce FM predictions with accuracies comparable to TBM models generated by other groups, and their models for TBM targets had an average accuracy comparable to low-to-medium resolution experimental structures.

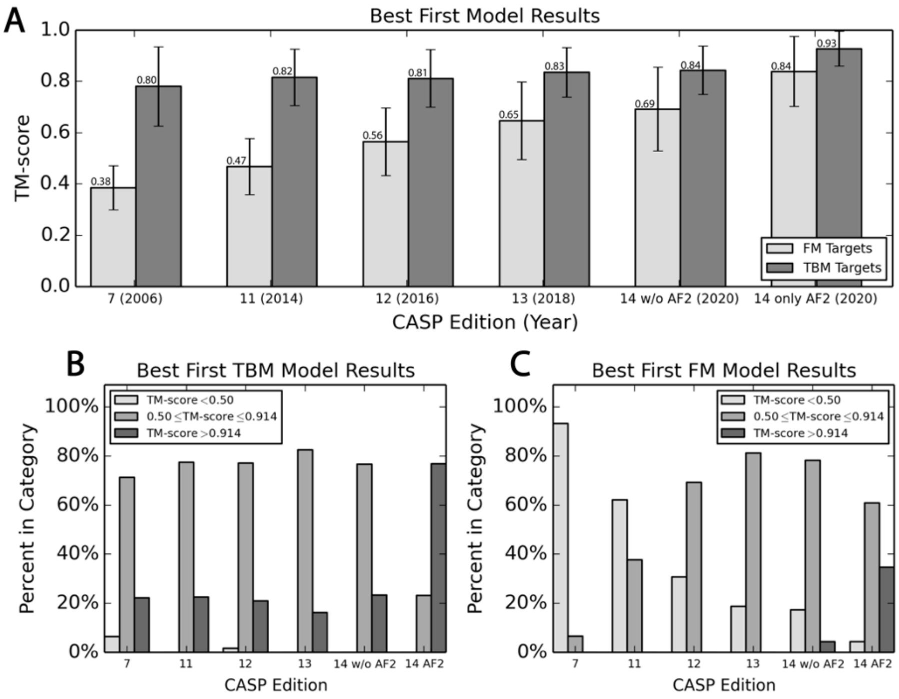


Figure 1.2 CASP modeling results from CASP7 through CASP14. (A) Mean TM-score of the best first TBM and FM models submitted in the corresponding CASP competitions. (B) Results for the best first TBM models (including TBM, TBM-easy, TBMA-hard, and FM/TBM) submitted by any group in CASP7/11-14, where the models are categorized into one of three categories based on their TM-scores: [0, 0.5), [0.5, 0.914], (0.914, 1.0]. (C) Results for the best first FM models submitted by any group in CASP7/11-14, where the models are categorized into one of three categories based on their TM-scores: [0, 0.5), [0.5, 0.914], (0.914, 1.0].

1.1.6 Other Self-Attention-based Networks for Structure Prediction

Inspired by the remarkable performance of AlphaFold2, current state-of-the-art structure prediction methods have followed suit in using deep self-attention networks. For example, RosettaFold, which was introduced in the months following CASP14, combines a self-attention-based MSA trunk network with a structure-based, SE(3)-equivariant graph transformer network to produce either the predicted coordinates for a given protein sequence (end-to-end version) or the predicted distance and orientation maps (pyRosetta) (126). For the pyRosetta version, the predicted restraints are used to guide gradient-based folding simulations to generate a final model. Interestingly, the authors found that the performance of the end-to-end version was slightly worse than that of the restraint-based pyRosetta version, which in part reflects the difficulty in end-to-end structure prediction training. Overall, RosettaFold was able to significantly outperform deep ResNet-based models in a retrospective benchmark test on CASP14 targets but was less accurate than AlphaFold2 (126).

Motivated by such advancements, we introduced DeepFoldRNA (127), which was the first available self-attention-based network for RNA tertiary structure prediction (Chapter 3). DeepFoldRNA is an extension of our DeepFold pipeline (122), where the ResNet architecture was replaced with a deep self-attention-based network that predicts the combined distance and orientation maps for an RNA molecule. Similar to the trunk network of AlphaFold2, DeepFoldRNA takes as input the MSA built for a query sequence, which is then processed using multiple layers of row- and column-wise self-attention to extract the evolutionary and positional information encoded in an alignment. The processed MSA embedding is then projected to a pair-wise positional embedding using an outer-product mean operation and refined using a triangular self-attention scheme as introduced by AlphaFold2 (123). Lastly, the distance and orientation maps are predicted from the final pair-wise embedding and converted into a deep learning-based potential that is minimized using gradient descent to produce a full-length model. Benchmark tests revealed that DeepFoldRNA significantly outperformed other leading RNA folding methods, with greatly reduced folding simulation times.

1.2 *De Novo* Protein Design

Having covered the field of protein/RNA structure prediction, we will now turn our attention toward protein design. Unlike protein structure prediction, which aims to model unknown 3D structures from known sequences, protein design attempts to identify new amino acid sequences that fold into specific 3D structures. *De novo* protein design usually contains two steps, the construction of a specific tertiary structure (or fold) and the identification/optimization of new amino acid sequences for that structure.

In addition to its use in protein structure prediction, fragment assembly has been successfully used to address the first step in *de novo* protein design, which is the construction of new protein folds beyond those observed in nature. One of the landmark achievements in *de novo* protein design was the design of Top7 in 2003 (128), which is one of the few proteins designed without a natural structural analog. The design of Top7 and other more recent *de novo* designed proteins have expanded on the strategies used by fragment assembly-based structure prediction methods, where a generic pipeline for such approaches is highlighted in Fig. 1.3.

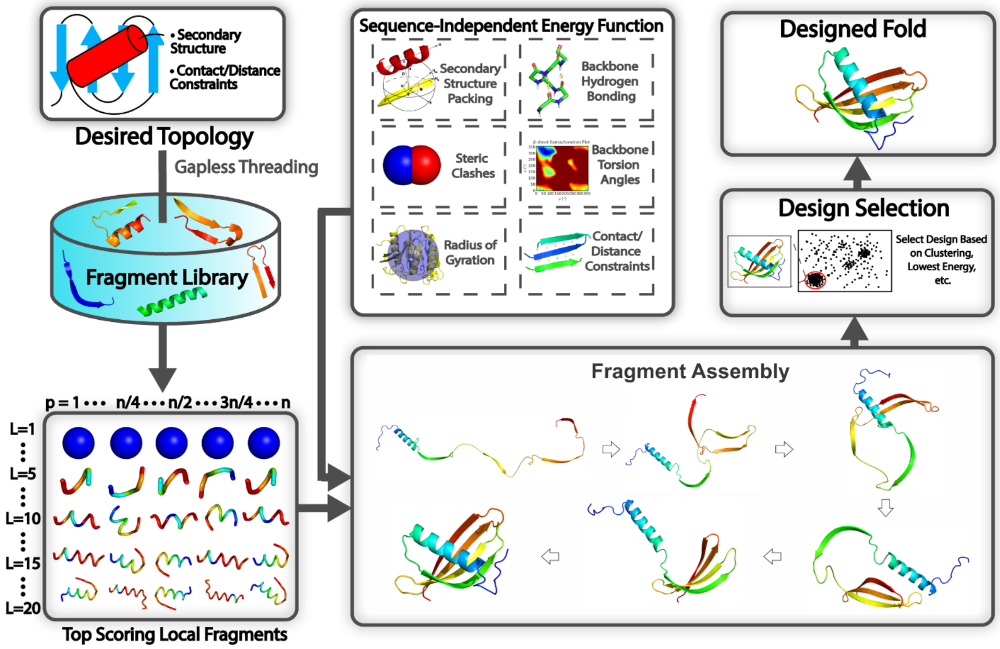

• Secondary Structure
• Contact/Distance Constraints
Desired Topology
Gapless Threading
Fragment Library
p = 1 ••• n/4 ••• n/2 ••• 3n/4 ••• n
L=1
L=5
L=10
L=15
L=20
Top Scoring Local Fragments
Sequence-Independent Energy Function
Secondary Structure Packing
Backbone Hydrogen Bonding
Steric Clashes
Backbone Torsion Angles
Radius of Gyration
Contact/ Distance Constraints
Designed Fold
Design Selection
Select Design Based on Clustering, Lowest Energy, etc.
Fragment Assembly

Figure 1.3 Typical steps involved in a fragment assembly-based approach to design new protein structures. Starting from the desired secondary structure together with any user-defined packing restraints, such as residue-residue contact/distance restraints, the query is searched through a non-redundant PDB structure library using gapless threading to generate position-specific fragment structures. High scoring fragments, which may range from 1-20 residues long, are identified based on the complementarity between the desired secondary structure and a fragment's secondary structure and backbone torsion angles. Then during the folding simulations, the top scoring local fragments are assembled under the guidance of a sequence-independent energy function, which accounts for fundamental rules that govern protein folding such as secondary structure packing, backbone hydrogen bonding, favorable backbone torsion angles, steric clashes, radius of gyration, as well as the artificial contact/distance restraints supplied by the user. As the method is sequence independent, generic side-chain centers of mass, typically those for valine, are used to evaluate energy terms such as steric clashes. Following the folding simulations, the final design may be selected based on clustering of the simulation decoys, by selecting the lowest energy structure, or through whatever filter the user deems appropriate.

Instead of starting from an amino acid sequence, leading structure design methods such as Rosetta (129) start from a predefined secondary structure and other user-defined constraints such as inter-residue distances, which define a target fold. Fragments are then picked with secondary structures and backbone torsion angles that are compatible with the predefined secondary structure. The simulation strategy is slightly altered as the amino acid-specific energy function is replaced with an energy function that is independent of the amino acid sequence and generic side-chain centers of mass are used to avoid steric clashes (129). Another popular method for designing backbone structures is to generate them using idealized parametric models (130), although this approach is typically more useful for designing helical bundle proteins and is not as effective at designing proteins with more complex topologies or hydrogen bonding networks.

Following the generation of the initial target folds based on the input constraints, iterative rounds of sequence and structure optimization are performed for amino acid sequence design (129). Here, sequence design and structure optimization can be performed using combined physics and knowledge-based energy functions such as Rosetta (131) or EvoEF2 (132). These approaches may start from a fixed protein backbone or perform flexible backbone refinement, where the amino acid side-chain conformation or rotamer of a randomly selected position is substituted for another rotamer randomly selected from a rotamer library (129). The corresponding energy changes caused by the mutation are then calculated using the physical energy function, where mutations are accepted or rejected based on the Metropolis criterion.

While most current protein design methods utilize physical energy functions to search for low free energy states in sequence space, the design results may be limited by the inability of physical energy functions to accurately recapitulate inter-atomic interactions or recognize correct folds, which has also been manifested in various protein folding and structure prediction studies (80, 133). Motivated by these limitations, as will be discussed in Chapter 4, we proposed EvoDesign (134), which includes evolutionary profiles derived from natural structural analogs in the force field in order to enhance the folding stability of the designed sequences and accommodate for the inaccuracies in purely physics-based energy models. For protein-protein interaction (PPI) design, EvoDesign starts from an input complex structure and identifies both monomeric and interface structural analogs from databases of solved protein structures. These structural analogs are converted into PPI evolutionary profiles, which are then combined with a physical energy function to guide the REMC sequence design simulations.

1.2.1 De Novo Design of Proteins with Complex Structures and Functions

The past few years have seen encouraging progress in *de novo* protein design, where proteins with increasingly complex structural characteristics and functions have been created (135-145). Although many *de novo* designed proteins have highly idealized structures with a single low energy conformation, recent work by Wei et al. demonstrated that it is possible to design proteins that adopt multiple low energy states that assume significantly different conformations (135). In the study, the authors used Rosetta to design a helical bundle that either adopted a short (~66 Å height) or long (~100 Å height) state based on the environmental conditions, which mimicked the action of membrane fusion proteins. Additionally, new studies have focused on designing proteins with more complex logical functions for use in synthetic biology. In this regard, Chen et al. was able to design logic gates that controlled transcription and enzymatic activity via the association of different designed coiled-coil heterodimers (136). The backbone structures of each coiled-coil were designed in a previous study using parametric modeling to generate the helices and loop fragments to connect them into a single chain (137). The association between different heterodimers was achieved using the Rosetta HBNet protocol (138), which can be used to exhaustively enumerate all of the hydrogen bond networks available for a given design space in order to design highly specific protein-protein interactions.

Rosetta has also been applied to the classical problem of designing proteins with significant β-sheet content, which have enriched hydrogen bonding patterns. For example, Dou et al. designed fluorescence-activated β-barrel proteins using either ideal parametric models or fragment assembly (139). Interestingly, the authors found that the ideal backbones generated by the parametric models had unfavorable steric strain and hydrogen bonding interactions. These problems were alleviated by building backbones using fragment assembly and introducing kinks and bulges into the structures, producing a stable and functional protein. Another challenging problem in protein design is the ability to create proteins that can bind to highly functionalized small molecules. Polizzi et al. tackled this problem by creating a unit of protein structure called the van der Mer, which directly maps the backbone of each amino acid to preferred positions of interacting chemical groups (145). The method was then used to design proteins capable of binding the complex drug apixaban, which has implications for the *de novo* design of customized biosensors and enzymes, among other applications (145).

1.2.2 De Novo Design of Therapeutic Proteins

Other studies have focused on designing proteins for therapeutic applications. One strategy to accomplish this goal is to design proteins that are capable of binding natural targets with high affinity. For instance, Chevalier et al. described a protocol for generating large pools of mini-proteins with different backbone scaffolds composed of ~40 residues produced by fragment assembly (140). The authors demonstrated that given advances in high throughput experimental techniques and computational modeling, an unprecedented number of designed proteins could be tested. This resulted in the production of highly stable designs that could bind to influenza hemagglutinin and provide prophylactic protection without eliciting an adverse immune response

(140). Another study by Silva et al. used parametric modeling to design mimics of IL-2 and IL-15 capable of binding the IL-2 receptor $\beta\gamma_c$ heterodimer but without binding sites for CD25 and CD215, producing a potent anti-cancer effect without the toxicity of natural IL-2 therapeutics (141). Furthermore, methods such as TopoBuilder have been used to generate computationally designed immunogens with topologies designed to stabilize functional motifs that induce the production of virus-neutralizing antibodies (142-144). These successes highlight the potential for *de novo* protein design to create therapeutics with tailor-made characteristics and superior efficacy compared to those produced by traditional approaches.

Lastly, during the COVID-19 pandemic, researchers sought to develop new proteins that could serve as therapeutic treatments. Along this line, in the study by Huang *et al.*, we proposed the design of *de novo* peptides to inhibit the association of the SARS-CoV-2 Spike protein with the human ACE2 receptor (146). The *in silico* assay experiments showed that the peptide inhibitors designed by EvoEF2 and EvoDesign had a significantly higher affinity for the binding domain of the Spike protein than the wildtype hACE2 receptor did. With a similar goal, Cao et al. applied Rosetta's fragment assembly design method to design protein inhibitors for the SARS-CoV-2 Spike protein (68). The authors used two design strategies, either incorporating the native helical interface between ACE2 and the Spike protein or generating novel interfaces *de novo* by optimizing the rotamer interaction field. After affinity maturation, they found the second approach was able to create proteins capable of potently inhibiting SARS-CoV-2 with picomolar affinity.

1.2.3 Recent Advancements in De Novo Protein Design Methodologies

Despite the successes, *de novo* protein design still remains somewhat of an art form, where large-scale experimental optimization is often required to generate successful designs (68, 70). In particular, extensive user-intervention during scaffold creation and selection is frequently necessary (71, 147). Furthermore, even given the examples in the previous sections, the ability to consistently design stable structures for non-idealized fold definitions or to create novel folds remains an outstanding problem in the field (148).

Recently, Anishchenko et al. performed an interesting study that combined deep neural-network training with structural refinement simulations to 'hallucinate' proteins; it could create novel protein sequences but the structural folds were generally close to PDB structures (with an average TM-score=0.78) (149). Meanwhile, the resulting protein folds were largely randomized depending on the stochastic process of the design iterations, where the method was further extended to allow for the incorporation of specific functional sites or structural motifs (150). In another recent approach, Huang et al. combined a neural network-derived, sidechain-independent potential (SCUBA) with stochastic dynamics simulations and demonstrated an impressive ability to generate successfully folded designs (151). Notably, the method should be used in tandem with 3D backbone sketches adapted from a ‘periodic table’ of protein structures (152) through manual manipulation and thus the conformational space of the final structures is limited to the topological area defined by the initial backbone sketches.

Similarly, extensions of the Rosetta fragment assembly protocol such as the aforementioned TopoBuilder require pre-definition of a target fold in the form of sketches that specify the 3D arrangement of the desired secondary structure (SS) elements. Then the sketches are parametrically optimized based on matching the desired fold with analogous structures in the PDB and assembled from fragments that match the fold definition using Rosetta (153). Other methods like SEWING (154) have been successful at producing stable designs by reassembling relatively large helical substructures identified from the PDB; however, the approach is limited to the conformations adopted by large substructures present in the PDB and has been benchmarked only on helical folds (154, 155). Additionally, as mentioned, most of the successful *de novo* designs have highly idealized structures with optimized SS compositions that lack the complex irregularities often present in native proteins, where a significant portion of the designed folds are well represented in nature or may be described through ideal parametric geometries (148, 156-159). Thus, development of automated algorithms capable of precisely designing any required fold type, including those without structure analogs in the PDB or idealized SS compositions, with limited human intervention is critical to improve the scope and success rate of *de novo* protein design.

To address this issue, we developed FoldDesign, which will be covered in Chapter 5, to create novel protein folds from specific secondary structure (SS) assignments through sequence-independent replica-exchange Monte Carlo (REMC) simulations. Detailed data analyses revealed that the major contributions to the successful structure design lay in the optimal energy force field, which contains a balanced set of secondary structure and novel fragment-based energy terms, and the efficient REMC simulations, which combine fragment assembly with multiple auxiliary movements to search the conformational space. On a large benchmark dataset of non-idealized, complex SS topologies, FoldDesign was able to consistently generate stable structure designs, where roughly 1/4 of the designs possessed novel folds that were not represented in the PDB, illustrating an important ability of the program to explore the areas of protein fold space unexplored by natural evolution.

1.3 Thesis Overview

The goal of my thesis is to develop new state-of-the-art methods for protein/RNA structure prediction and protein design. In the remaining chapters, I will cover representative works in each of these areas as follows.

Chapter 2 describes DeepFold, a method for *ab initio* protein structure prediction. DeepFold uses deep ResNets to predict the combined contact, distance, and orientation restraints from an MSA generated for a query sequence. These restraints are then converted to a deep learning-based potential that is combined with a general physical energy function, where rapid gradient-descent minimization is used to generate a full-length protein structure model.

Chapter 3 focuses on DeepFoldRNA, an extension of the DeepFold pipeline for *ab initio* RNA structure prediction. In DeepFoldRNA, the ResNet architecture of DeepFold is replaced with a deep self-attention-based network that generates predicted distance and orientation maps. These are then converted to a potential and minimized using gradient-descent simulations to produce a full-length RNA structure model.

Chapter 4 describes EvoDesign, an online tool for functional protein sequence design. EvoDesign combines evolutionary profiles collected from analogous protein folds with an optimized physics-based potential to generate new amino acid sequences for a given fold.

Chapter 5 describes FoldDesign, a method for *de novo* protein structure design. FoldDesign uses a sequence-independent energy function with REMC-based fragment assembly simulations to design new protein folds given a specific secondary structure topology definition.

Chapter 6 summarizes the findings and presents future directions.

CHAPTER 2

DeepFold: Fast and Accurate *Ab Initio* Protein Structure Prediction Using Potentials from Deep Learning

In this chapter, we will focus on the protein structure prediction problem. As mentioned in the introduction, depending on whether reliable structural templates are available in the PDB, protein structure prediction methods have been divided into template-based modeling (TBM) and template-free (FM) approaches (80). For many years, TBM has been the most reliable method for modeling protein structures; however, its accuracy is essentially determined by the availability of close homologous templates and the quality of the query-template alignments. Conversely, FM methods were designed to use advanced energy functions and sampling techniques to improve the folding performance for proteins that lack homologous templates in the PDB. However, due to the inaccuracy in force field design and the limitations of conformational search engines, the performance of the physics-based FM methods for non-homologous targets has remained significantly worse than that of the TBM methods for targets with readily identifiable homologous templates (160, 161).

Throughout the last few years, the use of deep learning techniques to predict spatial restraints has dramatically improved the accuracy of *ab initio* structure prediction (125). For example, in CASP11 and CASP12, predictors primarily used direct coupling analysis and shallow neural networks to predict contact maps, where the prediction accuracy largely relied on the identification of abundant sequence homologs in order to accurately predict contacts based on the information from correlated mutation patterns (115). In the CASP13 experiment, however, the top-ranked server groups, Zhang-Server and QUARK, used contact maps predicted by deep convolutional residual networks (ResNets) (162) to guide the I-TASSER (17) and QUARK (16) folding simulations, respectively, which greatly improved the contact prediction and folding accuracies for the physics- and knowledge-based modeling approaches. This was especially apparent for targets that lacked homologous templates and high-quality MSAs (115). In the recent CASP14 experiment, multiple deep learning constraints, including distance maps, which are conceptually similar to contact maps but include inter-residue distance information (36, 163), inter-residue dihedral angles (22) and hydrogen-bonding networks (164), were integrated with the folding simulations. The results demonstrated significant improvements over the contact-based structure assembly approaches, due to the introduction of more precise spatial information to guide the folding simulations (164).

Despite the improvement in modeling accuracy, the approaches built on traditional fragment/template assembly folding techniques, such as I-TASSER (17), Rosetta (14) and QUARK (16), often require lengthy simulation times, especially for longer proteins, which hinders them from large-scale modeling applications. In fact, the necessity of extensive conformational sampling required for *ab initio* modeling is due to the immense structure space and complex energy landscape associated with protein folding. Although this may still be required when integrated with sparse spatial constraints from threading alignments and low-resolution experiments (165-167), the advanced deep learning techniques can now provide abundant high-quality restraints. These abundant and accurate restraints can significantly smooth the rough protein folding energy landscape. In this regard, extensive folding simulations may no longer be needed, which partially explains the remarkable success enjoyed by other teams in the CASP experiments such as the first iteration of AlphaFold (163) in CASP13 and trRosetta (22), which construct structural models using local gradient-descent based conformational searching procedures.

Inspired by these advances, we developed a fast open-source protein folding pipeline, DeepFold, which combines a general physical force field and deep learning-based potential with rapid L-BFGS folding simulations to improve the speed and accuracy of FM protein structure prediction. The pipeline was carefully benchmarked on large-scale datasets and showed superiority over other leading structure prediction approaches, all with greatly reduced simulation times compared to traditional folding simulation methods. Notably, following the development of DeepFold, the newest self-attention-based methods, such as AlphaFold2 (123) and RosettaFold (126), were released and showed greatly improved modeling accuracy compared to deep convolutional ResNet architectures. Nevertheless, utilizing restraints from these methods, DeepFold was able to achieve similar or slightly better performance than the newest self-attention-based networks, demonstrating that it is a versatile platform that can be easily adapted for advances in the state of the art. Each component of the program, including the deep learning models and L-BFGS structure optimization pipeline, is integrated into an easy-to-use, stand-alone package available at both https://zhanggroup.org/DeepFold and https://github.com/robpearc/DeepFold. Meanwhile, an online webserver for DeepFold is available at https://zhanggroup.org/DeepFold, where users can apply the method to generate structure models for their own protein sequences.

2.1 Results and Discussion

2.1.1 Distance and orientation restraints have the dominant impact on global fold accuracy

As shown in Fig. 2.1, DeepFold starts by searching the query sequence through multiple whole-genome and metagenomic databases using DeepMSA2 (164) to create a multiple sequence alignment (MSA). Next, the co-evolutionary coupling matrices are extracted from the resulting MSA and used as input features by the deep ResNet architecture of DeepPotential to predict spatial restraints, including distance/contact maps and inter-residue torsional angle orientations. These restraints are then converted into a deep learning-based potential, which is used along with a general physical potential to guide the L-BFGS folding simulations for full-length model generation (see Methods).

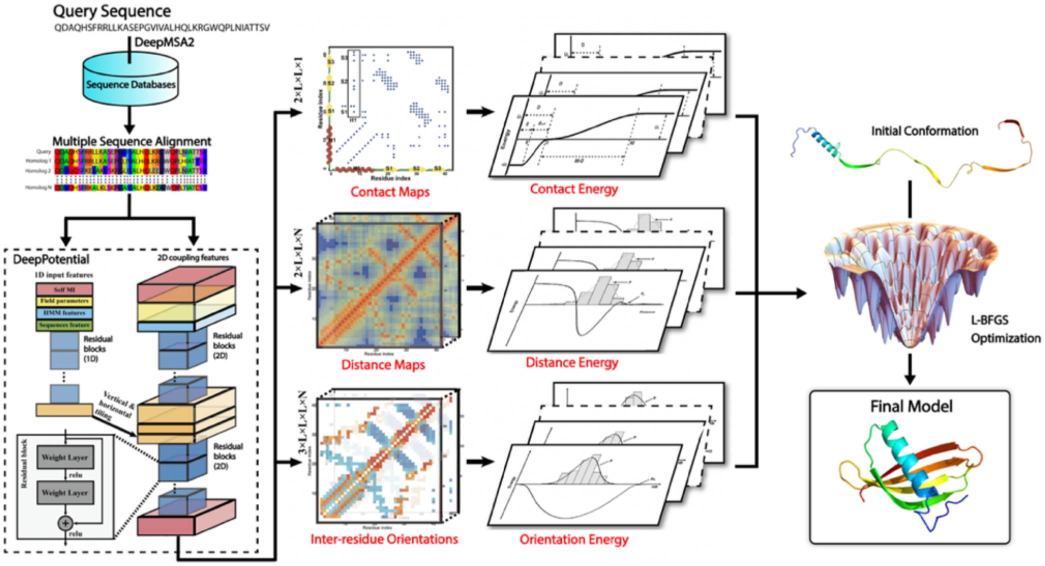


Figure 2.1 Overview of the DeepFold pipeline. Starting from a query amino acid sequence, DeepMSA2 is used to search the query against multiple whole-genome and metagenome sequence databases to create a multiple sequence alignment (MSA). The MSA is then used by DeepPotential to derive input features based on co-evolutionary analyses for the deep ResNet training. DeepPotential outputs the probability distribution of Cβ-Cβ/Cα-Cα contact and distance maps as well as the inter-residue orientations. These restraint potentials along with the inherent statistical

energy function are used to guide the L-BFGS folding simulations for final full-length structure model construction.

To test DeepFold, we collected a set of 221 non-redundant (<30% sequence identity to each other) protein domains from the SCOPe 2.06 database and FM targets from CASP9-12. These proteins were non-homologous (with a sequence identity <30%) to the training dataset of DeepFold, were solved at 3 Å resolution or better by X-ray crystallography, had lengths between 100-500 residues, and were all defined as Hard threading targets by LOMETS (168) after excluding homologous templates with >30% sequence identity to the query. Here, a Hard target is a protein for which LOMETS could not identify a significant template, allowing for a systematic evaluation of the developed method on FM modeling targets. To examine the importance of the different components of the DeepFold energy function, we ran DeepFold using different combinations of spatial restraints from DeepPotential for the 221 test proteins, where the modeling results are summarized in Fig. 2.2 and Table B.1 in Appendix B.

Overall, the baseline potential using just the general physical energy function (GE in Table B.1 and Fig. 2.2) achieved an average TM-score of only 0.184. Furthermore, when considering a cutoff TM-score ≥0.5 to indicate a correctly folded model, which would mean the predicted model and native structure share the same global fold (169, 170), the baseline energy function was unable to correctly fold any of the test proteins (Table B.1). Given that the coupling of a similar force field with replica-exchange Monte Carlo simulations in QUARK could fold substantially more proteins with a much higher average TM-score (16), this result suggests that one major reason for the failure here is due to the frustration of the baseline energy landscape, which cannot be quickly explored by gradient-based searching methods. The further inclusion of Cα and Cβ contact restraints improved the TM-score to 0.263, where 4 of the 221 test proteins, or 1.8%, were successfully folded with TM-scores ≥0.5. The addition of the Cα and Cβ distance restraints dramatically improved the average TM-score on the test dataset to 0.677, representing an increase of 157.4%, where 76.0% of the test proteins were correctly folded. Lastly, the inclusion of the inter-residue orientations further improved the average TM-score to 0.751 and the percent of successfully folded proteins to 92.3%. Overall, as the level of detail in the restraints increased, the energy landscape became increasingly smooth and thus the L-BFGS folding simulations resulted in increased average TM-scores across the test proteins.

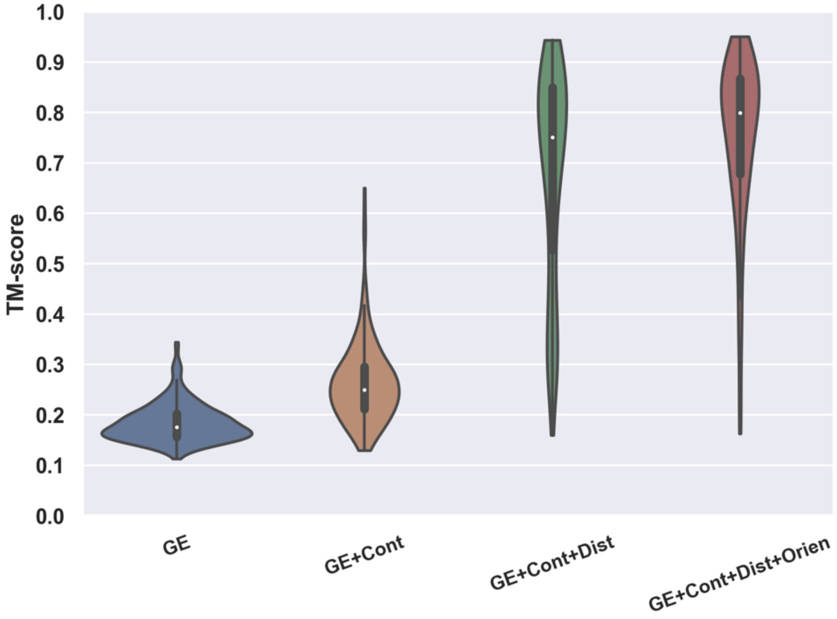


Figure 2.2 Contribution of the various spatial restraints and energy terms on the DeepFold modeling accuracy. The violin plot shows the TM-score of DeepFold using different combinations of energy terms/restraints on the 221 test proteins, including the general physical energy function (GE), contact restraints (Cont), distance restraints (Dist), and orientation restraints (Orien).

Although the addition of inter-residue distances to the energy function brought about the largest improvement in accuracy, one interesting observation is the synergistic effect observed when combining different components of the restraints. For example, the addition of inter-residue orientations improved DeepFold's ability to find structures that optimally satisfied the distance restraints. As evidence of this, in Table B.2 we present the mean absolute errors (MAEs) for the top $n*L$ long-range distance restraints, where L is the protein length and n is a chosen scale factor, which were calculated between the DeepPotential predicted distance maps and the final DeepFold models with and without the use of the orientation restraints.

The table shows that the introduction of inter-residue orientations helped to significantly decrease the MAE between the predicted distance maps and the structure models. For example, when considering the top $2*L$ distance restraints, which were sorted by their DeepPotential distance prediction confidence scores, the MAE was 0.74 Å when DeepFold was run using the GE and contact/distance restraints, whereas the MAE was reduced by 17.6% to 0.61 Å when the orientation restraints were added. Therefore, not only do orientations provide useful geometric information on their own, they also help further smooth the energy landscape and facilitate the L-BFGS search to identify energy basins that satisfy the ensemble of spatial restraints.

Furthermore, inter-residue orientations were particularly useful for folding β-proteins. As seen in Table B.3, the inclusion of orientations increased the average TM-score for β-proteins from 0.590 to 0.706, corresponding to a 19.7% improvement, which was significantly higher than the 10.9% improvement observed on the overall dataset (Table B.1); this makes sense intuitively given the intricate hydrogen bonding patterns present in β-proteins that would require more detailed local inter-residue dihedral angle restraint information to properly recapitulate. Fig. 2.3 presents an illustrative example from SCOPe protein d1jqpa1, which adopts a β-barrel fold. The model built without orientations had a low TM-score of 0.313 and an RMSD of 11.43 Å, where the MAE between the top 2*L DeepPotential distances and the model without orientations was 0.87 Å. In contrast, the model built using the orientation restraints had a drastically improved TM-score of 0.800 and an RMSD of 2.74 Å. Additionally, the MAE between the top 2*L DeepPotential distances and the model improved to 0.61 Å. Thus, the orientation restraints provide complementary information to the distance maps and had a particularly important role for folding β-proteins.

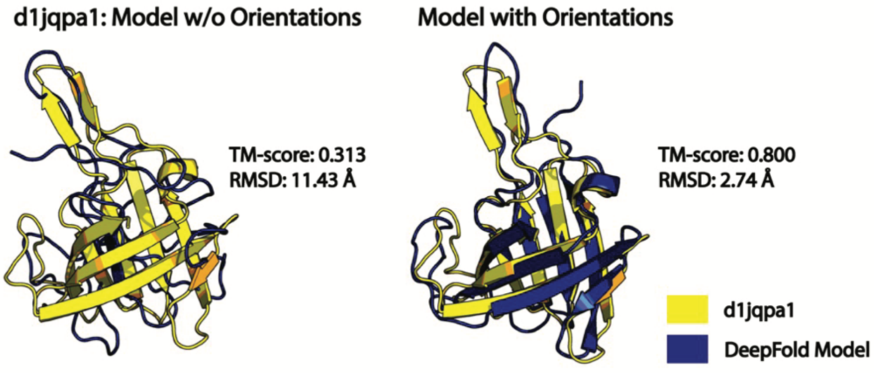


Figure 2.3 Case study from SCOPe protein d1jqpa1 that demonstrates the importance of inter-residue orientations for folding β-proteins. The native structure is shown in yellow, and the superposed predicted models built without (left) and with (right) orientation restraints are shown in blue.

2.1.2 The general physical energy function improves local structure quality

The rapid improvement in the accuracy of deep learning-based restraint prediction has called into question the role of the physical energy function in the era of deep learning. Indeed, we saw that the major contributor to DeepFold's accuracy is the high number of accurately predicted restraints generated by DeepPotential, where their addition dramatically improved the average TM-score from 0.184 to 0.751 (Fig. 2.2). Nevertheless, the physical energy function, which accounts for fundamental forces that drive protein folding, such as hydrogen bonding interactions and van der Waals clashes, plays an important role in improving the physical quality of the predicted models; this is especially true when the model quality is poor. As evidence, Table 2.1 lists several model quality metrics for models generated with and without the use of the GE function.

On the overall test set of 221 hard protein targets, the inclusion of the GE potential provided a modest yet consistent enhancement in the physical model quality, as reflected in the improvement of the MolProbity score (171) from 1.735 to 1.692 with the addition of the GE function (Table 2.1). Similar trends were observed for the secondary structure quality (SOV score (172)), the number of Ramachandran outliers, and the steric clash score, all of which improved with the inclusion of the GE (Table 2.1). The most notable improvement was observed in the clash score, which improved by 13.3% on the overall dataset.

More significant improvements were witnessed for the 16 targets with poor physical quality, as measured by a MolProbity score in the 50th percentile or lower from the PDB structures. For these targets, the physical energy function improved the average MolProbity score from 2.882 to 2.308, representing an improvement of 19.9% compared to 2.5% on the overall dataset. Similarly, these improvements were consistent across the SOV score, number of Ramachandran outliers, and the clash score for these targets. Again, the most dramatic improvement occurred for the clash score, which decreased from 17.5 to 8.6, representing an improvement of 50.9%.

Target Type (# of Proteins)	**DeepFold Energy Function**	**SS SOV**	**Rama Outliers**	**Clash Score**	**MP-score**
All Targets (221)	w/o General Energy	79.68%	6.52	3.61	1.735
	with General Energy	**79.71%**	**5.92**	**3.13**	**1.692**
MP-score <50th Percentile (16)	w/o General Energy	58.41%	13.00	17.54	2.882
	with General Energy	**61.44%**	**9.81**	**8.58**	**2.308**

Table 2.1 Impact of the general energy (GE) function on DeepFold's modeling performance. Specifically, the table presents the effect of the GE on the secondary structure SOV score, number of Ramachandran outliers, the MolProbity clash score, and the total MolProbity score on the overall dataset and those targets with poor physical model quality.

Fig. 2.4 illustrates a case study from SCOPe protein d1xsza2, where models were generated with and without the inclusion of the general physical energy function. In the model built without the GE function, there are several residues that directly overlap each other leading to severe steric clashing, as shown in the inset. These clashes among other factors led to a model with a large, and thus unfavorable, MolProbity score of 3.908 (3^{rd} percentile) along with a very high clash score of 212.8. As shown in the inset in Fig. 2.4, these clashes were resolved with the inclusion of the GE potential and its term for van der Waals clashes, where the resulting model had a reduced MolProbity score of 1.624 (92^{nd} percentile) and a low clash score of 1.2. Clearly, simply satisfying the geometric restraints provided by deep learning may lead to models that are physically unrealistic, where the introduction of physical energy terms may partially alleviate this problem.

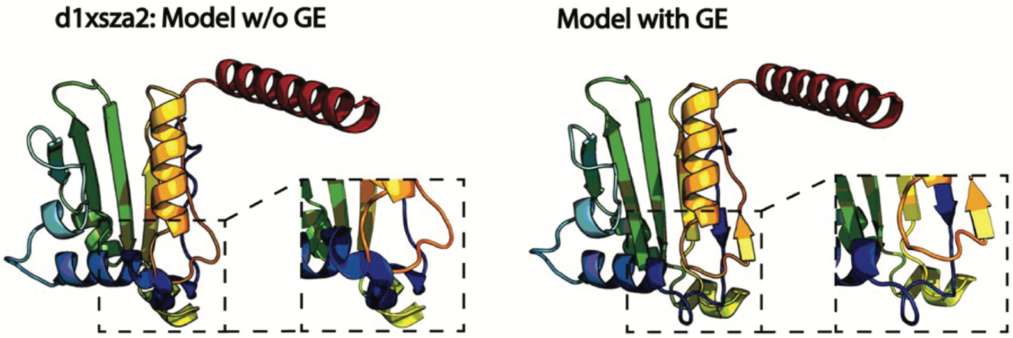


Figure 2.4 Case study from SCOPe protein d1xsza2, which highlights the importance of the general energy function for improving the physical quality of the models. The models built without (left) and with (right) the general physical energy function are depicted in rainbow coloring, where the clashing region is shown in the inset.

2.1.3 Comparison of DeepFold with other leading modeling methods

To further evaluate the performance of DeepFold, we compared the modeling results on the 221 test proteins with a leading contact map-based folding program (C-I-TASSER (173)), two top distance (DMPfold (174)) and distance/orientation-based (trRosetta (22)) methods, and the classic I-TASSER pipeline (17), where the results are summarized in Table 2.2. To provide a fair comparison, we used the same MSAs that DeepFold used, which were produced by DeepMSA2 (164) (see Fig. 2.12 in Methods section 3.3.1), for the deep learning restraint prediction by DMPfold, trRosetta and C-I-TASSER, as well as for template identification by LOMETS in I-TASSER and C-I-TASSER. Furthermore, templates with ≥30% sequence identity to the query were excluded from I-TASSER and C-I-TASSER.

As shown in Table 2.2, the average TM-score of the DeepFold models for the 221 test proteins was significantly higher than all the control methods. For instance, the average TM-score for the models produced by I-TASSER was only 0.383, where DeepFold achieved an average TM-score (0.751) that was 96.1% higher than I-TASSER with a *p*-value of 9.4E-80 as determined by a paired, two-sided Student's t-test (Table 2.2). This result is understandable as I-TASSER does not use any deep learning spatial restraints, making the modeling accuracy more reliant on the templates, while, by design, all homologous templates were excluded for the Hard threading targets. The inclusion of deep learning contact maps into C-I-TASSER greatly increased the TM-score to 0.584. Nevertheless, DeepFold still achieved an average TM-score that was 28.6% higher than C-I-TASSER with a *p*-value of 1.8E-55. This is mainly due to the fact that DeepFold utilizes both distance and orientation restraints, which contain more detailed information than the contact maps used in C-I-TASSER (115).

Method	**TM-score (*p*-value)**	**RMSD (*p*-value)**	**Correct Folds**[*]	**$TM_{DeepFold}$ > TM_{Method}**[‡]
I-TASSER	0.383 (9.4E-80)	15.10 (7.1E-25)	24.0%	95.9%
C-I-TASSER	0.584 (1.8E-55)	8.89 (4.0E-26)	67.0%	95.9%
DMPfold	0.657 (5.6E-37)	7.81 (2.0E-18)	79.6%	92.3%
trRosetta	0.694 (8.3E-24)	6.81 (4.7E-09)	85.5%	87.8%
DeepFold	**0.751**	**5.61**	**92.3%**	**-**

** This column represents the percent of proteins with TM-scores ≥0.5.*

‡ This column indicates the percent of test proteins for which DeepFold generated a model with a higher TM-score than the control method.

Table 2.2 Summary of structure modeling results by DeepFold and the control methods on the 221 test proteins. The *p*-values were calculated between DeepFold and the control methods using paired, two-sided Student's t-tests.

Interestingly, there were two targets (d1ltrd and d1nova) for which I-TASSER and C-I-TASSER produced models that were significantly more accurate than DeepFold. To examine the reason for the discrepancy in performance, Fig. 2.5 depicts the models generated by I-TASSER, C-I-TASSER, and DeepFold superposed with the native structures along with the top templates used by I-TASSER and C-I-TASSER for these proteins. For d1ltrd, despite the fact that it was a hard threading target, LOMETS was able to identify a reliable template from the PDB (1prtI) with a coverage of 92.6% and a TM-score of 0.553; thus, both I-TASSER and C-I-TASSER constructed accurate models with TM-scores of 0.663 and 0.637, respectively. Conversely for DeepFold, the generated MSA contained few homologous sequences with a normalized number of effective sequences (or Neff, defined in Text D.1 in Appendix D) of 0.42, resulting in inaccurate predicted restraints with an MAE of 2.60 Å for the top 2**L* distances. This ultimately lead DeepFold to produce an inaccurate model with a TM-score of 0.326. Additionally, the contact precision for the top *L*/2 contacts used by C-I-TASSER was only 50.0%, which is largely why the C-I-TASSER model was worse than the I-TASSER model.

Similarly, for d1nova, LOMETS was able to identify a reliable template (PDB ID 1hofC) with a coverage of 100% and a TM-score of 0.544, which resulted in accurate I-TASSER and C-I-TASSER models with TM-scores of 0.631 and 0.713 for the two methods, respectively. Again, for DeepFold, the generated MSA was shallow with a normalized Neff value of 9.40. Nevertheless, the predicted distance restraints were still accurate with an MAE of 0.90 Å for the top 2**L* distances; however, the predicted orientations were inaccurate, particularly the Ω orientation, which had an MAE of 31.3° for the top 2**L* restraints. This resulted in a model with a TM-score of 0.546, which still possessed a correct fold, but was less accurate than the models generated by I-TASSER and C-I-TASSER. Unlike the previous example, the C-I-TASSER model was more accurate than the I-TASSER model for d1nova as the predicted contacts were accurate with a precision of 98.7% for the top *L*/2 contacts. These two examples highlight that even with the advances in deep learning methods, template-based modeling still remains important, particularly given the reliance of deep learning techniques on the generated MSAs, which may be lower quality than the identified templates for numerous targets.

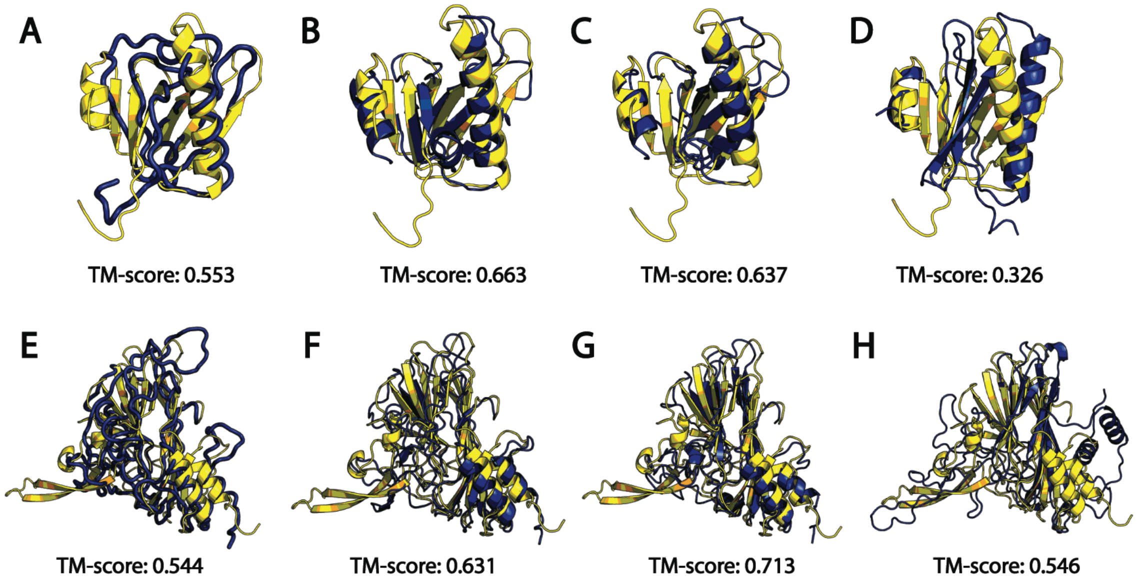


Figure 2.5 Case study from two targets, d1ltrd (A-D) and d1nova (E-H), for which I-TASSER/C-I-TASSER outperformed DeepFold. A) LOMETS template (blue) superposed with the native structure for d1ltrd (yellow); B) I-TASSER model (blue) superposed with the native structure (yellow); C) C-I-TASSER model (blue) superposed with the native structure (yellow); D) DeepFold model (blue) superposed with the native structure (yellow); E) LOMETS template (blue) superposed with the native structure for d1nova (yellow); F) I-TASSER model (blue) superposed with the native structure (yellow); G) C-I-TASSER model (blue) superposed with the native structure (yellow); H) DeepFold model (blue) superposed with the native structure (yellow).

DeepFold also outperformed the two other leading distance (DMPfold) and distance/orientation-based (trRosetta) methods, where DMPfold achieved an average TM-score of 0.657 and trRosetta obtained an average TM-score of 0.694. Therefore, DeepFold's average TM-score was 14.3% higher than DMPfold and 8.2% higher than trRosetta, where the differences were statistically significant with *p*-values of 5.6E-37 and 8.3E-24, respectively (see Table 2.2). Furthermore, Fig. 2.6 presents a head-to-head comparison of DeepFold with the control methods, where DeepFold outperformed trRosetta and DMPfold on 194 and 204 of the 221 test proteins, respectively.

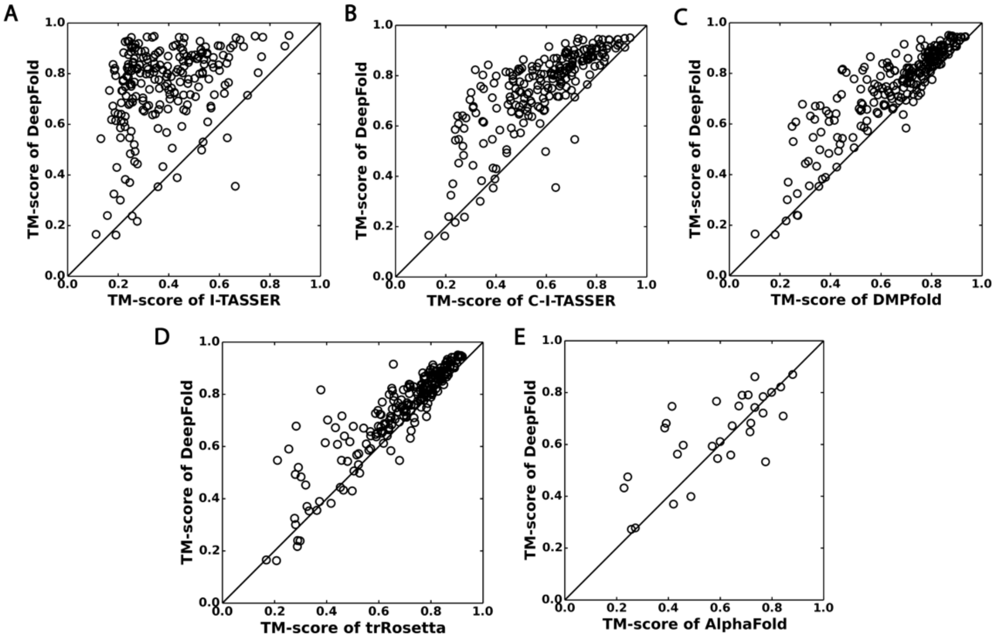


Figure 2.6 Head-to-head TM-score comparisons between DeepFold and other protein structure prediction methods.A) I-TASSER; B) C-I-TASSER; C) DMPfold; D) trRosetta; E) AlphaFold. (A-D) are based on the 221 Hard benchmark proteins, while (E) is on 31 FM targets from CASP13.

Compared to DMPfold, an obvious advantage of DeepFold is the use of inter-residue dihedral angle orientations, which resulted in a substantial TM-score increase for DeepFold as shown in Fig. 2.2. Compared to trRosetta, since both methods use distance and orientation restraints, the major advantage of DeepFold is the high accuracy of the restraints generated by DeepPotential. Therefore, in Table 2.3, we provide an accuracy comparison for the $C\beta$ distance predictions by different programs, where the distance maps by DeepPotential had a significantly lower MAE to the native structures than those produced by both trRosetta and DMPfold across all cutoff values.

Method	L/2 (*p*-value)	L (*p*-value)	2L (*p*-value)	5L (*p*-value)	10L (*p*-value)
DeepPotential	**0.974 (*)**	**1.018 (*)**	**1.090 (*)**	**1.302 (*)**	**1.613 (*)**
trRosetta	1.050 (4.9E-02)	1.154 (5.9E-04)	1.328 (2.8E-06)	1.730 (2.0E-07)	2.241 (1.4E-11)
DMPfold	1.779 (1.4E-15)	1.930 (7.6E-22)	2.184 (7.5E-28)	2.695 (1.6E-33)	3.488 (1.1E-41)

Table 2.3 MAEs of the top $n*L$ long-range distances by different distance predictors on the 221 test proteins. The *p*-values were calculated using paired, two-sided Student's t-tests between the DeepPotential results and the control methods.

In Table B.4, we also list the modeling results of trRosetta using the DeepPotential restraints. Although trRosetta+DeepPotential resulted in a higher average TM-score (0.735) than trRosetta alone, due to the use of the more accurate restraints from DeepPotential, the average TM-score of DeepFold was still significantly higher than that of trRosetta+DeepPotential with a p-value of 3.9E-09. This is likely due to the unique DeepFold knowledge-based force field and the utilization of the additional $C\alpha$ distance maps that are not used by trRosetta. In addition, the simultaneous optimization of the DeepFold force field with the L-BFGS search engine (see Methods) helped enhance the structure construction process.

Here, of particular interest is the modeling performance for those hard targets with very few effective sequences in their MSAs, which are the most difficult targets to fold using deep learning approaches. For this purpose, we collected a set of 16 targets with normalized Neff values less than 1 and calculated the TM-scores for the models produced by DeepFold, trRosetta, and DMPfold. On these targets, DeepFold achieved an average TM-score of 0.494, which was 40.3% higher than trRosetta (0.352) and 44.9% higher than DMPfold (0.341). In Fig. 2.7, we present a scatter plot of TM-score vs. the logarithm of the normalized MSA *Neff* value for the three methods on all 221 test proteins, where DeepFold demonstrated a lower correlation between the TM-score and *Neff* value than trRosetta and DMPfold, which partially explains the superior performance of DeepFold.

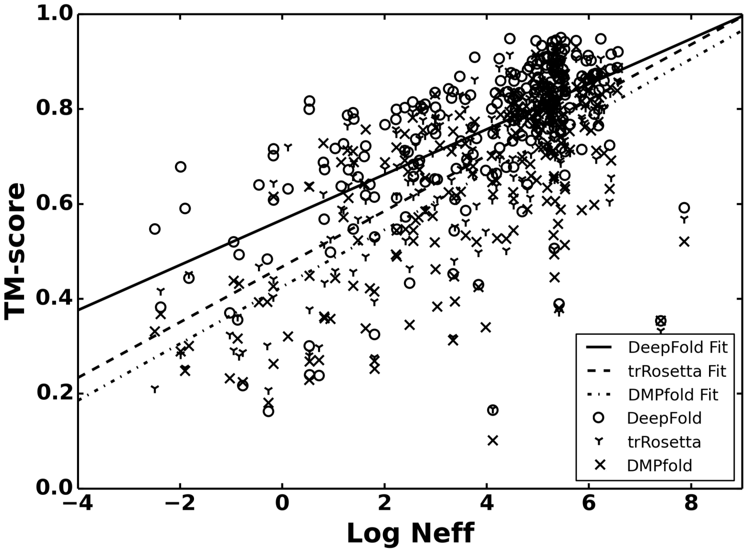


Figure 2.7 Model TM-score vs. the logarithm of the MSA Neff value for DeepFold, trRosetta, and DMPfold. The fitted models were obtained by linear regression with Pearson's Correlation Coefficients of 0.615, 0.712, and 0.675 for DeepFold, trRosetta, and DMPfold, respectively.

Lastly, we compared the modeling accuracy of DeepFold with the first version of AlphaFold on the 31 CASP13 FM targets that the AlphaFold human group submitted models for (Table B.5).

Note, we could not benchmark the performance of AlphaFold on the 221 test proteins as the feature generation scripts and folding pipelines are not publicly available. It can be seen from Table B.5 that DeepFold outperformed AlphaFold on 20 of the 31 FM targets, where, on average, the TM-score of DeepFold was 0.636 compared to 0.589 for AlphaFold (*p*-value=0.025, Table B.5). It is also important to note that the AlphaFold human group performed thousands of different optimization runs for the CASP13 targets as reported (163), while DeepFold only used a single optimization run in this study.

2.1.4 Comparison of DeepFold with AlphaFold2 and RosettaFold

Since DeepFold uses restraints from DeepPotential, which was developed before the advances made by AlphaFold2 (123) in CASP14, it is also of interest to compare the results against the most recent self-attention-based neural network methods, namely, AlphaFold2 and RosettaFold (126). Thus, in Fig. C.1 in Appendix C, we provide a head-to-head comparison of the DeepFold modeling results utilizing the restraints from DeepPotential with RosettaFold and AlphaFold2 on the 221 test proteins in terms of the model TM-scores, where the results are summarized in Table 2.4.

Overall, the average TM-score of the RosettaFold end-to-end pipeline was 0.812 and the average TM-score of the Pyrosetta version was 0.838, which were higher than the results by DeepFold (TM-score=0.751) with *p*-values of 3.6E-10 and 8.0E-22, respectively. Similarly, the average TM-score of AlphaFold2 was 0.903, which was higher than DeepFold with a p-value of 1.4E-49. These results were expected given that the advances in deep self-attention neural networks and end-to-end training by AlphaFold2 and, subsequently, RosettaFold showed greatly improved modeling accuracy over previously introduced convolutional ResNet architectures, such as DeepPotential.

Method	**Mean TM-score** **(*p*-value)**	**Correct** **Folds***
RosettaFold (End-to-End)	0.812 (3.6E-10)	93.7%
RosettaFold (Pyrosetta)	0.838 (8.0E-22)	**95.5%**
AlphaFold2	**0.903 (1.4E-49)**	95.0%
DeepFold	0.751	92.3%

** This column represents the percent of proteins with TM-scores ≥0.5.*

Table 2.4 Modeling results of DeepFold using the DeepPotential restraints vs RosettaFold and AlphaFold2 on the 221 test proteins. For the mean TM-scores, the *p*-values were calculated using paired, two-sided Student's t-tests.

Notably, there were 7 targets for which DeepFold outperformed AlphaFold2. In Fig. 2.8, we illustrate two examples where DeepFold generated models that were significantly more accurate than AlphaFold2. The first example is from SCOPe protein d1a34a, for which DeepFold generated a model with a TM-score of 0.613, while AlphaFold2 generated a model with a TM-score of 0.242. For this target, DeepMSA2 was not able to identify any sequence homologs, resulting in an MSA composed of only the query sequence and an extremely low normalized Neff value of 0.08. Nevertheless, DeepPotential generated accurate restraints with an MAE of 1.10 Å for the top 2*L distances, resulting in a higher quality model than that produced by AlphaFold2.

The second example is from SCOPe protein d1s2xa, for which DeepFold generated a model with a TM-score of 0.590, while AlphaFold2 generated a model with a TM-score of 0.369. Again, for this target, DeepMSA2 was only able to identify two sequence homologs, which resulted in a very low normalized Neff value of 0.15. Additionally, the DeepPotential restraints were fairly inaccurate with an MAE of 2.54 Å for the top 2*L distances and 59.29° for the 2*L Ω orientations. Surprisingly, even though the orientation restraints were inaccurate, their inclusion greatly improved the modeling accuracy, as the model built using only the contact and distance restraints possessed a low TM-score of 0.268, while the model built using the full set of contact/distance and orientation restraints had a TM-score of 0.514. Moreover, the addition of the general knowledge-based energy function further improved the TM-score to 0.590. This suggests that even when inaccurate, the combination of various restraints with a general energy function may act synergistically to filter out inaccuracies in the predictions.

It is noteworthy that the two preceding examples were from proteins with few to no homologous sequences. In fact, if we consider the 5 proteins in the benchmark dataset with the least homologous sequence information (<3 sequence homologs) and normalized Neff values <0.20, DeepFold generated more accurate models than AlphaFold2 for 4 of these targets, where the average TM-score of DeepFold was 0.528 compared to 0.398 for AlphaFold2. This suggests that, while deep self-attention-based protein structure prediction approaches have demonstrated an improved ability to fold proteins with few sequence homologs, the performance on the most extreme cases remains to be improved.

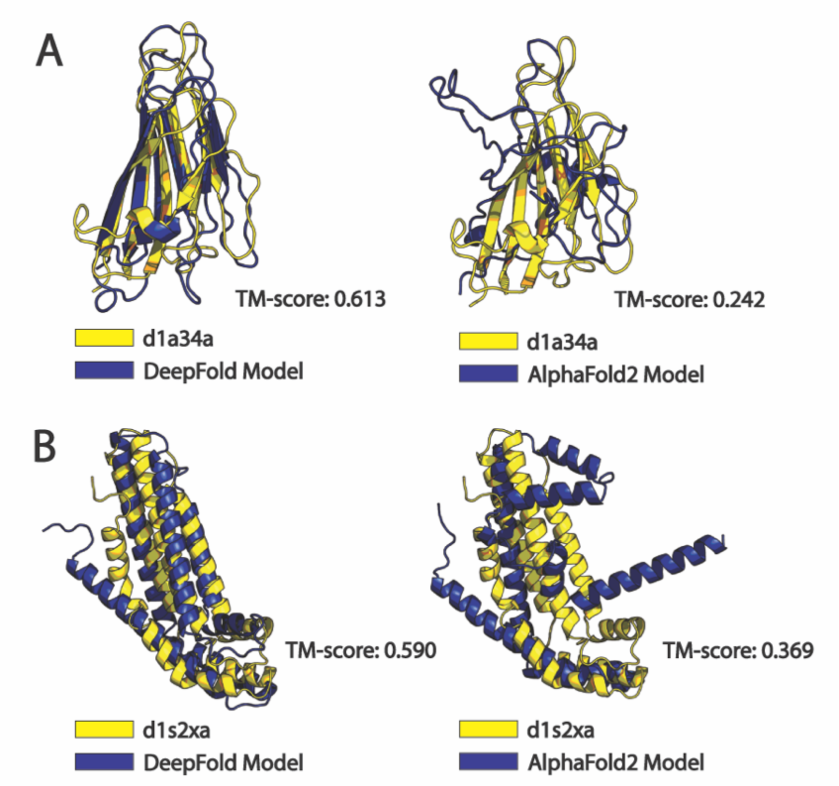


Figure 2.8 Case study from two proteins (d1a34a and d1s2xa) for which DeepFold significantly outperformed AlphaFold2. The DeepFold/AlphaFold2 models are shown in blue superposed with the native structures in yellow.

Lastly, given the importance of the most recent advances in protein structure prediction, we sought to determine whether or not they could be incorporated into DeepFold to further improve its performance. To answer this question, we utilized the restraints from RosettaFold, including the Cβ distances and orientations, as well as the Cα distances/contacts and Cβ contacts from DeepPotential to guide the DeepFold simulations. The results of this analysis are depicted in Table 2.5 and Fig. C.2, which present head-to-head comparisons between DeepFold utilizing the combined restraints with RosettaFold and AlphaFold2 in terms of the model TM-scores on the 221 benchmark proteins.

With the combined RosettaFold and DeepPotential restraints, DeepFold achieved an average TM-score of 0.844, higher than that attained by the end-to-end (TM-score=0.812) and Pyrosetta (TM-score=0.838) versions of RosettaFold with *p*-values of 2.4E-11 and 1.2E-2, respectively. These data demonstrate that the DeepFold knowledge-based force field and DeepPotential contact and Cα distance restraints may improve the results obtained by RosettaFold. Additionally, it shows that DeepFold is a versatile platform that can be easily adapted for any future advances in state-of-the-art deep learning restraint predictors.

Method	**Mean TM-score (*p*-value)**	**Correct Folds**[*]
RosettaFold (End-to-End)	0.812 (2.4E-11)	14.3%
RosettaFold (Pyrosetta)	0.838 (1.2E-02)	95.5%
AlphaFold2	**0.903 (4.1E-11)**	95.0%
DeepFold	0.844	**96.4%**

** This column represents the percent of proteins with TM-scores ≥0.5.*

Table 2.5 Modeling results of DeepFold using the combined RosettaFold and DeepPotential restraints vs RosettaFold and AlphaFold2 on the 221 test proteins. For the mean TM-scores, the *p*-values were calculated using paired, two-sided Student's t-tests.

2.1.5 DeepFold greatly improves the accuracy and speed of protein folding over classical ab initio methods

Rosetta (14) and QUARK (16) are two of the most well-known fragment-assembly methods and have been consistently ranked as the top methods for *ab initio* protein structure prediction in previous CASP experiments (161, 175, 176). However, a major drawback of the traditional *ab initio* folding approaches is that their modeling performance drops as the protein length increases, making them significantly less reliable for modeling larger protein structures composed of more than 150 residues (80). To examine the impact of deep learning on *ab initio* structure prediction for long protein sequences, we compared DeepFold to both Rosetta and QUARK, where Fig. 2.9.C depicts the TM-score of DeepFold, QUARK, and Rosetta vs the protein length. The data show that the performance of DeepFold remained consistent as the protein length increased, where the average TM-score for large proteins composed of 350-450 residues was in fact higher than that for the small proteins in the test set with lengths <150 residues (0.809 vs. 0.742), mostly due to the more favorable MSAs collected for the set of larger proteins. However, the performance of both QUARK and Rosetta noticeably decreased as the protein length increased; the average TM-score for proteins with lengths less than 150 residues was 0.329 for QUARK and 0.304 for Rosetta but was only 0.190 and 0.196 for QUARK and Rosetta, respectively, on proteins with lengths between 350 and 450 residues. From these results, DeepFold outperformed QUARK and Rosetta remarkably on the overall dataset and especially on the longest proteins in the dataset, for which the average TM-score of DeepFold was 325.8% higher than QUARK and 312.8% higher than Rosetta.

Another major limitation of fragment-assembly approaches is that they require lengthy simulations to adequately explore the immense structure space available. In Figs. 2.9.A-B, we list a comparison of the folding simulation time requirement for DeepFold and the QUARK fragment assembly approach for different protein lengths. The results show that the speed of DeepFold is orders of magnitude faster than QUARK, especially for large proteins. Note that we ran QUARK using 5 separate trajectories in parallel and the run time shown in Fig. 2.9.A is the average run time across all 5 simulation trajectories. Thus, if the simulations were run sequentially, the run time would be 5 times longer, which further accentuates the cost of fragment assembly. Therefore, while fragment assembly requires hours to days to fold a protein, DeepFold requires only seconds to minutes. Overall, the average run time of DeepFold on the test set was 6.98 minutes, while the average for QUARK was 1830.82 minutes for an average protein length of 188.1 residues. This indicates that QUARK requires 262.3 times the computing time that DeepFold requires for one simulation trajectory, and the difference was even greater as the sequence length increased. Overall, the run time of DeepFold was similar to trRosetta, which required 5.48 minutes to construct models on the test dataset on average. Of particular importance is that the greatly reduced folding times did not cause the model quality to deteriorate for larger proteins, demonstrating the ability of deep learning restraints to effectively smooth the energy landscape, thereby allowing rapid and accurate optimization across protein lengths.

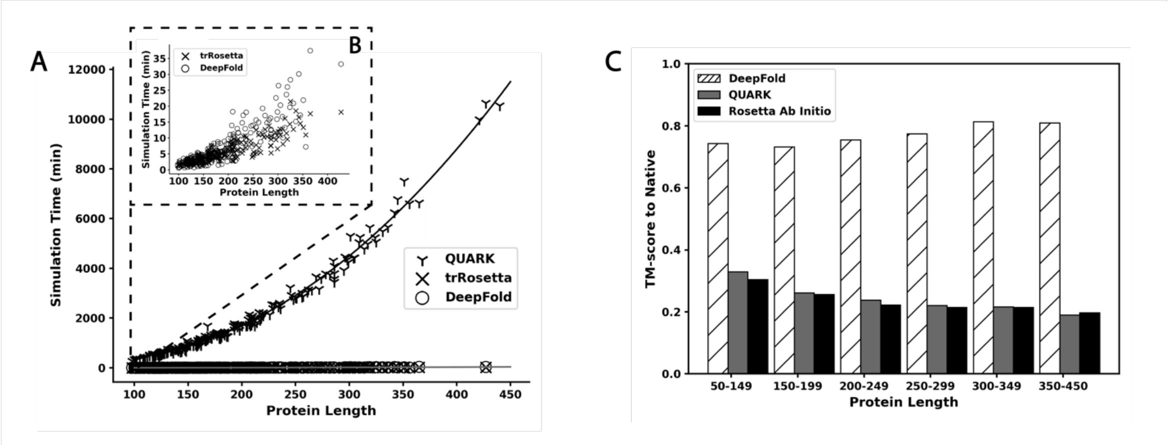


Figure 2.9 Dependence of simulation time and TM-score on protein length. A) Simulation runtime for QUARK, trRosetta, and DeepFold in minutes plotted against the protein length. B) A close up of the runtime vs protein length for DeepFold and trRosetta. C) Analysis of the average TM-score for DeepFold, QUARK, and Rosetta across different protein length ranges.

2.1.6 Gradient-based protein folding requires a high number of deep learning restraints

The success of rapid L-BFGS-based protein folding approaches raises the question on what the role of fragment assembly is in protein structure prediction. As L-BFGS and other gradient-based methods are essentially local optimization techniques that may be prone to becoming trapped in local energy minima, the more extensive conformational sampling performed by fragment assembly may still be necessary in the absence of a high number of deep learning spatial restraints.

To examine this hypothesis, Fig. 2.10 depicts the TM-score for L-BFGS-based protein folding simulations using different numbers of spatial restraints. Consistent with the data in Fig. 2.2, Fig. 2.10 shows that only using the GE function to guide the L-BFGS simulations resulted in a poor average TM-score of 0.184. This was significantly lower than that obtained by QUARK (TM-score =0.274), which uses a similar physical energy function without deep learning restraints (16). These data indicate the frustration of the baseline physical energy force field used by DeepFold, which cannot be quickly explored with gradient-based methods. Inclusion of the top L all-range Cβ distances slightly improved the TM-score to 0.186, and at least the top 5*L distances were required to improve the TM-score to a significant degree. In order to achieve a performance that was better than QUARK, the L-BFGS simulations required 10*L Cβ distance restraints, where the average TM-score using this number of restraints was 0.323. The inclusion of more distance restraints, such as the top 15*L and 20*L restraints, steadily improved the average TM-score to 0.392 and 0.453, respectively.

However, our tests showed that setting a specific probability cutoff for the selection of distance restraints allowed the method to achieve the best result. In DeepFold, all distances with a probability >0.55 were selected for inclusion in the L-BFGS optimization procedure, which corresponded to an average of ~93$*L$ distance restraints on the test set, increasing the TM-score to 0.668. Overall, the addition of the full set of DeepPotential restraints (including contacts, Cα distance and orientations in addition to the Cβ distances) increased the accuracy by an additional 12.4%, resulting in a TM-score of 0.751 for the full pipeline. Thus, it is clear that L-BFGS requires a high number of spatial restraints in order to adequately smooth the energy landscape and make gradient-based protein folding feasible.

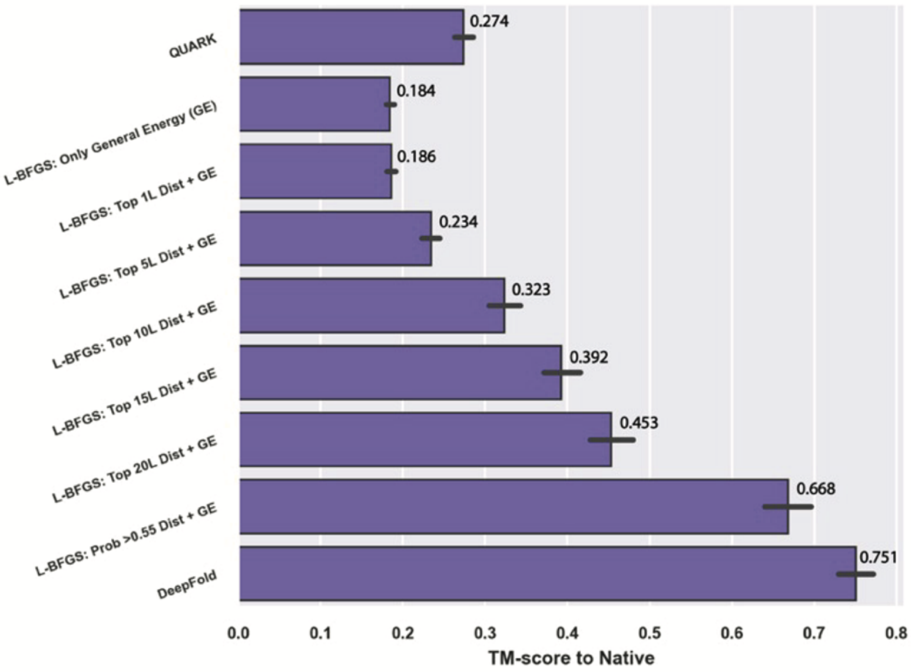


Figure 2.10 Evaluation of the modeling accuracy of QUARK and DeepFold guided by different numbers of spatial restraints. The top $n*L$ distances were selected by sorting the Cβ distances according to their predicted probabilities.

2.1.7 Case study reveals drastically different dynamics in Monte Carlo and L-BFGS folding simulations

To further illustrate the differences in the sampling procedures for the fragment assembly method, QUARK, and the L-BFGS optimization approach, DeepFold, we present in Fig 2.11 a case study from the amino terminal domain of enzyme I from E. coli (SCOPe ID: d1zyma1). Both DeepFold and QUARK generated a correct fold for this target, where the TM-score of the model produced by QUARK was 0.547 and the TM-score for the DeepFold model was very high at 0.923

with an RMSD of 1.29 Å, indicating a close atomic match to the experimental structure.

To show the conformational changes during the QUARK folding simulations, Fig. 2.11.A depicts the TM-score of the conformation for the last replica at REMC cycle *i* relative to the conformation of the previous decoy at cycle *i*-1. From the figure, it can be seen that large changes in the conformation occur throughout the simulation due to the global conformational searching and replica exchange steps. On the other hand, the opposite trend was observed for the L-BFGS folding simulations shown in Fig. 2.11.B, during which large conformational changes occurred early on in the simulation, and the global fold of the protein was largely determined by the 100th L-BFGS step. After that, only small fluctuations in the conformation occurred, where the L-BFGS optimization quickly converged and did not extensively sample the structure space due to the nature of the local optimization of the smooth energy landscape produced by the large number of deep learning restraints. Moreover, Fig. 2.11.C depicts the DeepFold models at L-BFGS steps 100 and 1100 superposed with the experimental structure. While the global fold of the model was determined by the 100th L-BFGS step, substantial conformational changes occurred during the later L-BFGS steps at the two regions, namely the highlighted terminal coil and core helix regions, which were poorly formed at step 100 due to the inconsistency in the spatial restraints in these sections. For the helix region in particular, the model at step 100 had poorly formed secondary structure as well as severely clashing segments. These errors were gradually corrected over the remaining 1000 L-BFGS steps. Therefore, while the global folds of proteins may quickly be determined by the consensus DeepPotential restraints during the L-BFGS simulations, additional steps are often needed to precisely fine-tune the model quality under the guidance of the atomic force field.

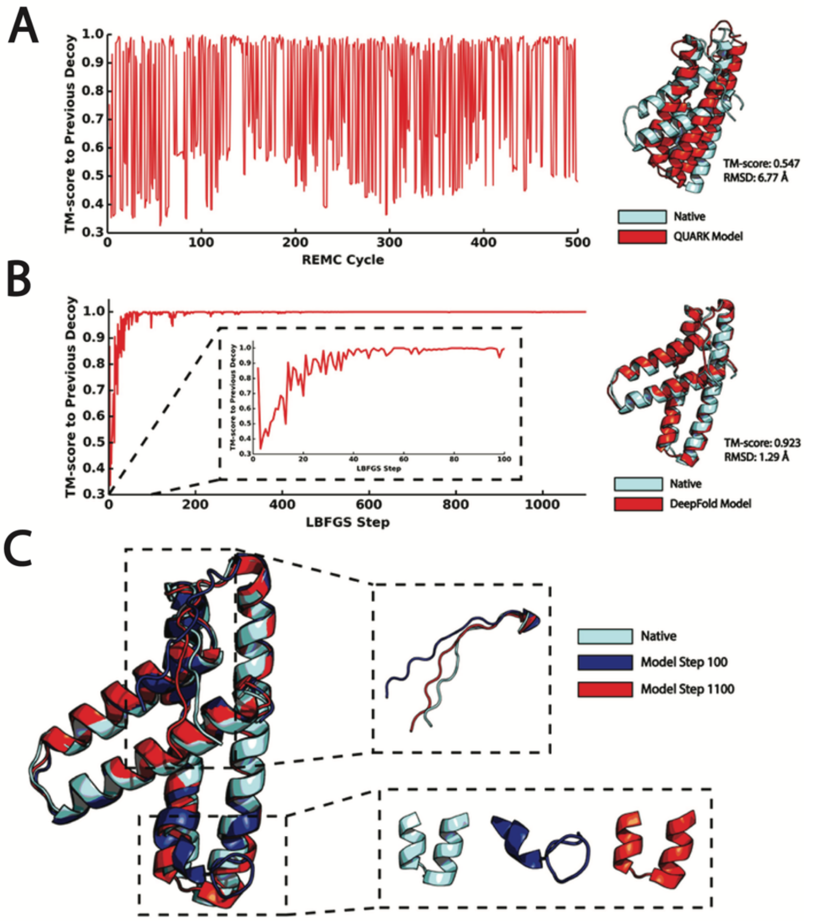


Figure 2.11 Comparison of the simulation dynamics for DeepFold and QUARK. A) Analysis of the conformational changes that occur during the QUARK fragment assembly simulations. The figure plots the TM-score of the decoy at REMC cycle *i* compared to the decoy at the previous cycle *i-1*. The right hand side shows the final QUARK model in red superposed with the native structure in cyan. B) Analysis of the conformational changes that occur during the DeepFold simulations. The figure plots the TM-score of the decoy at L-BFGS step *i* compared to the decoy at the previous step *i-1*, where the right hand side shows the final DeepFold model in red superposed with the native structure in cyan. C) Comparison between the DeepFold model at L-BFGS step 100 (blue) with the model at step 1100 (red) and the experimental structure (cyan). The insets show the areas of the structure that change the most after the 100th L-BFGS step.

2.2 Concluding Remarks

We developed an open-source program (DeepFold) to quickly construct accurate protein structure models from deep learning-based potentials. DeepFold significantly outperformed other *ab initio* structure prediction methods such as Rosetta, QUARK, I-TASSER, C-I-TASSER, DMPfold, and trRosetta on the test set of 221 Hard threading targets, and AlphaFold on the CASP13 FM targets. The impact of deep learning on DeepFold was best highlighted by the benchmark test with Rosetta, QUARK and I-TASSER, which represent the top traditional FM and TBM methods. On the benchmark dataset, Rosetta, QUARK and I-TASSER were only able to generate correctly folded models for 0.9%, 2.7% and 24.0% of the proteins, respectively, while DeepFold successfully folded 92.3% of the test proteins with an average TM-score of 0.751, compared to 0.260, 0.274, and 0.383 for Rosetta, QUARK and I-TASSER, respectively.

Furthermore, the average TM-score of DeepFold was 7.8% and 13.9% higher than the other leading deep learning-based methods, DMPfold and trRosetta, respectively, starting from the same MSAs. It was also 8.0% higher than AlphaFold on the 31 CASP13 FM targets. Of particular interest is the performance on the hardest targets in the dataset with very shallow MSAs (i.e., with normalized Neff values less than 1), where the average TM-score of DeepFold was 40.3% higher than trRosetta and 44.9% higher than DMPfold. On top of the improved accuracy, DeepFold had a similar running time as other gradient descent-based approaches such as trRosetta, but it was more than 200 times faster than the traditional fragment-assembly based approaches. The success of DeepFold is mainly due to the effective combination of the inherent knowledge-based potential with the high number of accurately predicted spatial restraints that help smooth the energy landscape, making L-BFGS optimization tractable.

Despite the success, significant improvements may still be made. For example, the use of attention-based networks (123, 177, 178), especially an end-to-end learning protocol (123), should help further improve the prediction accuracy of DeepFold. Given that the main input features to DeepPotential are derived from co-evolutionary analyses, DeepFold often requires the input MSAs contain a sufficient number of effective sequences to enable determination of the co-evolutionary relationships between protein residues. Despite the fact that the quality of the DeepFold models was considerably less dependent on the MSA quality than other methods such as DMPfold and trRosetta, the use of a transformer architecture should help further enhance the performance of DeepPotential for those targets with poor MSA quality and few homologous sequences by self-attention-based, iterative MSA refinement. This can be illustrated by the comparison of DeepFold with the most recent methods, RosettaFold and AlphaFold2, which achieved higher TM-scores on the benchmark targets. Nevertheless, when utilizing the combined RosettaFold and DeepPotential restraints, DeepFold was able to outperform both the end-to-end and distance-based versions of RosettaFold, demonstrating that it is a versatile platform that can be easily adapted for advances in the state of the art. Meanwhile, DeepFold outperformed AlphaFold2 on 4 out of the 5 targets with the least homologous sequence information (normalized Neff <0.2), revealing that there is significant room for improvement on very difficult modeling targets.

Furthermore, more efficient and precise MSA construction strategies should be developed to improve the MSA quality and reduce the time required to search the various sequence databases. The need to increase the searching efficiency is particularly important as the increase in the size of the sequence databases, mainly the metagenomics databases, is a double-edged sword. While it enables the collection of more sequences, it also greatly increases the time and computational resources necessary to search the sequence databases and the potential for false negative sequence samples due to the increase in noise. For example, searching a 150-residue protein through MetaClust, which is approximately 100 GB, using DeepMSA2 requires around 1 hour with 1 CPU; however, searching the same protein through the 5TB JGI metagenome database is dramatically more expensive, requiring approximately 4 hours using 50 CPUs. This issue is particularly important for hard modeling targets, which often require extensive homologous sequence detection. As evidence of this, in Fig. C.3, we plot the number of times each of the 7 MSAs produced by DeepMSA2 were selected for the 221 benchmark targets. From the figure, it can be seen that ~55% of the targets required searching beyond the MetaClust database, while only ~15% did not require searching through any metagenomics database. Meanwhile, incorrectly collected MSAs, despite having a high number of homologous sequences, can negatively impact the modeling results as witnessed in the CASP experiments (20). The use of a targeted MSA generation protocol that focuses on searching sequences related to the target protein's biome represents a promising strategy for improving the speed and quality of the MSA generation and the accuracy of the final 3D structure modeling (179).

2.3 Methods

DeepFold is an algorithm that can quickly construct accurate full-length protein structure models from deep learning restraints and consists of three main steps: MSA generation by DeepMSA2, spatial restraint prediction by DeepPotential, and L-BFGS folding simulations, as depicted in Fig. 2.1.

2.3.1 MSA generation by DeepMSA2

DeepMSA2 is an extension of DeepMSA (180) for iterative MSA collection, where the new components include an additional pipeline to search larger sequence databases and a novel MSA selection method based on predicted contact maps (see Fig. 2.12 below). Briefly, DeepMSA2 collects 7 candidate MSAs by iteratively searching whole-genome (Uniclust30 and UniRef90) and metagenome (Metaclust, BFD, and Mgnify) sequence databases. The first 3 MSAs are generated using the same procedure as DeepMSA (i.e., dMSA in Fig. 2.12), where the query sequence is first searched through Uniclust30 (2017_04) by HHblits2 to create MSA-1. Next, the sequences identified by Jackhmmer and HMMsearch are used to construct a custom HHblits database, against which HHblits2 is run starting from the MSA generated in the previous stage to generate MSA-2 and MSA-3, respectively. The four remaining MSAs are generated using a procedure called quadruple MSA (qMSA in Fig. 2.12), which uses HHblits2 to search the original query sequence against the Uniclust30 database (version 2020_01) to create MSA-4. Next, the sequences detected by Jackhmmer, HHblits3, and HMMsearch through the UniRef90, BFD, and Mgnify databases are used to construct custom HHblits-style databases, against which HHblits2 is used to search starting from the MSAs generated by the previous stages to create MSA-5, MSA-6, and MSA-7, respectively. To select the final MSA, a quick TripletRes contact map prediction (181) is run starting from each of the 7 MSAs, where the MSA with the highest cumulative probability for the top 10*L all-range contacts is selected as the final MSA.

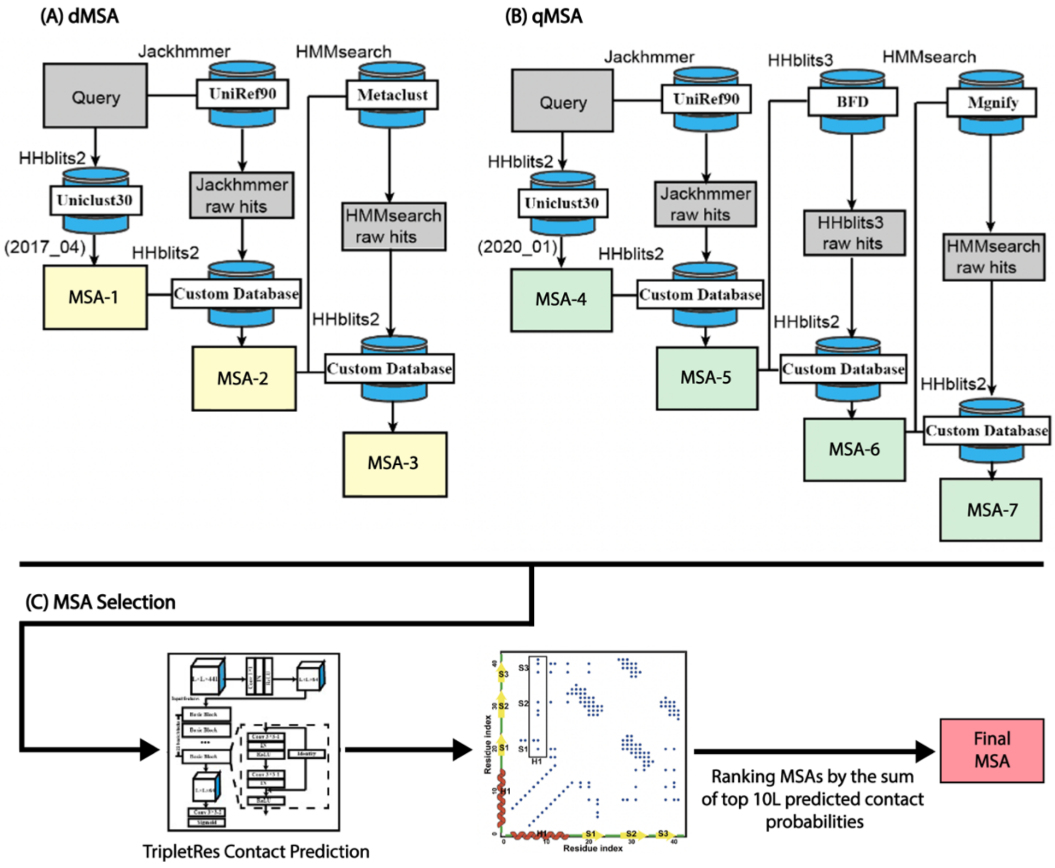


Figure 2.12 DeepMSA2 pipeline, which contains three major steps: (A) dMSA, (B) qMSA, and (C) MSA selection.

2.3.2 Spatial restraint prediction by DeepPotential

Starting from the selected MSAs, two sets of 1D and 2D features are extracted. The 2D features include the raw coupling parameters from the pseudo likelihood maximized (PLM) 22-state Potts model and the raw mutual information (MI) matrix, where the 22 states of the Potts model represent the 20 standard amino acids, a non-standard amino acid type, and a gap state. As mentioned in Chapter 1, a Potts model is a specific type of Markov Random Field (MRF) model that is widely-used in protein structure prediction (106, 107, 182, 183). Briefly, an MRF is a graphical model that represents each column of an MSA as a node that describes the distribution of amino acids at a given position (Potts model field parameters), where the edges between nodes indicate the joint distributions of amino acids at each pair of positions. The 2D coupling parameters can then be determined from the edge weights, where residue pairs that exhibit correlated mutation patterns will possess greater edge weights, which can be used to infer positions that should be closer together in 3D space. This is based off of the intuition that if two residues are in contact with each other, then when one residue mutates, the contacting residue should also mutate in order to preserve the interaction. In DeepPotential, CCMpred (183) is used to fit the Potts model. The corresponding parameters for each residue pair in the PLM and MI matrices are extracted as additional features that measure query-specific co-evolutionary information in an MSA. The 1D features contain the Potts model field parameters, Hidden Markov Model (HMM) features, and the self-mutual information, along with the one-hot representation of the MSA and other descriptors, such as the number of sequences in the MSA.

Next, these 1D and 2D features are fed into deep convolutional residual neural networks separately, where each of them is passed through a set of one-dimensional and two-dimensional residual blocks, respectively, and are subsequently tiled together. The tiled feature representations are considered as the input of another fully residual neural network which outputs the inter-residue interaction terms, including Cα-Cα distances, Cβ-Cβ distances, and the inter-residue orientations (Fig. 2.1). Here, the predicted spatial restraints are represented using various bins that correspond to specific distance/angle values, where DeepPotential predicts the probability that the spatial restraints fall within the specific bins. For example, for the Cα and Cβ distances, the predictions are divided into 38 bins, where the first bin represents the probability that the distance is <2Å and the final bin represents the probability that the distance is ≥20Å. The remaining 36 bins represent the probability that the distance falls in the range [2Å, 20Å), where each bin has a width of 0.5 Å. On the other hand, the 3 orientation features, as defined in Fig. 2.13, are predicted using a bin width of 15° with an additional bin to indicate whether there is no interaction between the two residues (i.e., Cβ-Cβ distance ≥20Å). The DeepPotential models were trained on a set of 26,151 non-redundant proteins collected from the PDB at a pair-wise sequence identity cutoff of 35%.

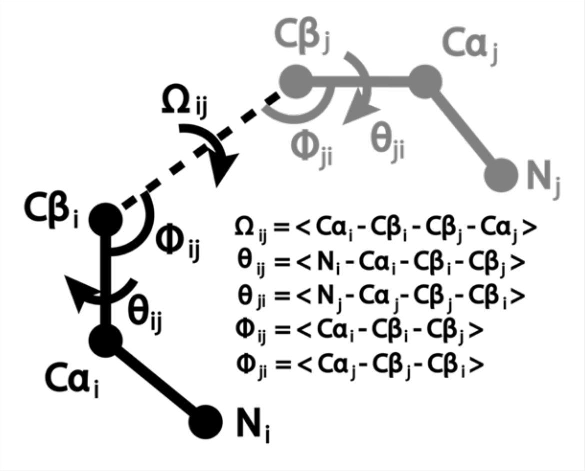

Cβj
Cαj
Ωij
Φji
θji
Nj
Cβi
Φij
θij
Cαi
Ni
Ωij = < Cαi - Cβi - Cβj - Cαj >
θij = < Ni - Cαi - Cβi - Cβj >
θji = < Nj - Cαj - Cβj - Cβi >
Φij = < Cαi - Cβi - Cβj >
Φji = < Cαj - Cβj - Cβi >

Figure 2.13 Definition of the inter-residue orientations predicted by DeepPotential. Ω and θ are inter-residue torsion angles formed by the four indicated atoms and φ is an inter-residue angle formed by three atoms.

2.3.3 DeepFold Force Field

The DeepFold energy function is a linear combination of the following terms:

$$E_{DeepFold} = (E_{C\beta dist} + E_{C\alpha dist} + E_{C\beta cont} + E_{C\alpha cont} + E_{\Omega} + E_{\theta} + E_{\varphi}) + (E_{hb} + E_{vdw} + E_{tor}) \quad (2.1)$$

where the first seven terms $E_{C\beta dist}$, $E_{C\alpha dist}$, $E_{C\beta cont}$, $E_{C\alpha cont}$, E_{Ω}, E_{θ}, and E_{φ} account for the predicted Cβ–Cβ distances, Cα–Cα distances, Cβ–Cβ contacts, Cα–Cα contacts, and three inter-residue orientation angles by DeepPotential; and the last three terms E_{hb}, E_{vdw}, and E_{tor} denote the generic energy terms for hydrogen bonding, van der Waals clashes, and backbone torsion angles, respectively.

Overall, the DeepFold force field consists of 24 weighting parameters, where the weights given to each of the deep learning restraints were separated into short (1<$|i-j| \leq 11$), medium (11<$|i-j| \leq 23$) and long-range ($|i-j| > 23$) weights, which were determined by maximizing the TM-score on the training set of 257 non-redundant, Hard threading targets collected from the PDB that shared <30% sequence identity to the test proteins. Briefly, all the weights were initialized to 0, then the weight for each individual energy term was varied one-at-a-time by an increment of 0.25 in the range from [0, 25] and the DeepFold folding simulations were run using the new weights. The weight for each term that resulted in the highest average TM-score on the training set was accepted. After the initial weighting parameters were determined, 3 more optimization runs were carried out, where the weight for each energy term was again varied in a range from [0, 25] using an increment of 0.1 and the weighting parameters that resulted in the highest average TM-score on the training set were accepted. A final optimization run was carried out, where the weights were perturbed by [-2, 2] from their previously accepted values using an increment of 0.02 to precisely fine-tune their values. The details of each energy term are further explained in Text D.2 in Appendix D. Since DeepPotential provides the bin-wise histogram probability of the spatial descriptors, these terms are further fit with cubic spline interpolation to facilitate the implementation of the L-BFGS optimization, which requires a continuously differentiable energy function.

2.3.4 L-BFGS Folding Simulations

A protein structure in DeepFold is specified by its backbone atoms (N, H, Cα, C, and O), Cβ atoms and the side-chain centers of mass (Fig. 2.14).

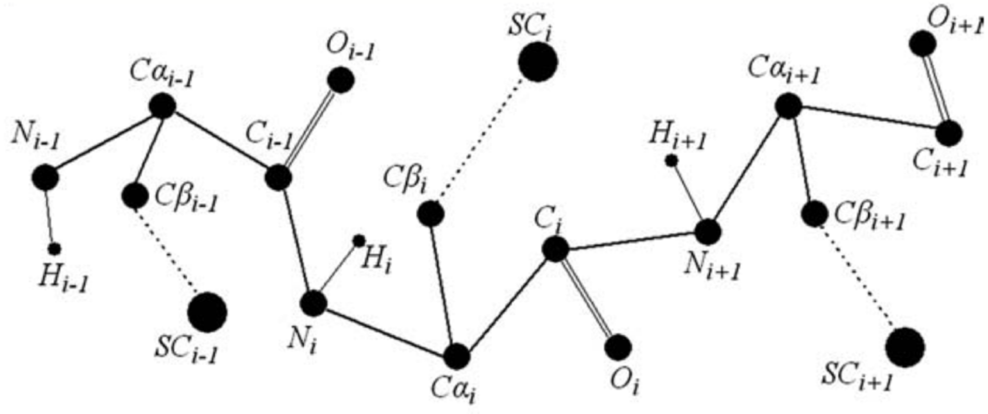


Figure 2.14 Depiction of the reduced model used to represent protein conformations during the DeepFold folding simulations.The conformations include the backbone atoms (N, H, Cα, C, and O) as well as the Cβ atoms and side-chain centers of mass for each amino acid type.

The initial conformations are generated from the backbone torsion angles (ϕ, ψ) predicted by ANGLOR through a small, fully-connected neural network (184), where the cartesian coordinates of the backbone atoms are determined using simple geometric relationships, assuming ideal bond length and angle values. The conformational search simulations are performed using L-BFGS, with bond lengths and bond angles fixed at their ideal values, and the optimization is carried out on the backbone torsion angles.

Here, L-BFGS is a gradient-descent based optimization method that is a limited memory variant of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. At each step k, the search direction d_k of the simulation is calculated by

$$d_k = -H_k^{-1} \cdot \nabla E_{DeepFold}(x) \tag{2.2}$$

where H_k^{-1} is an estimate for the inverse Hessian matrix and $\nabla E_{DeepFold}(x)$ represents the gradient of $E_{DeepFold}(x)$ with respect to the backbone torsion angles $x = (\phi, \psi)$. The value of H_k^{-1} at step $k = 0$ is set to the identity matrix, I, and the value of H_{k+1}^{-1} is obtained following the BFGS formulation

$$\begin{cases} H_{k+1}^{-1} = V_k^T H_k^{-1} V_k + \rho_k s_k s_k^T \\ V_k = I - \rho_k y_k s_k^T \\ \rho_k = {}^{1}\!/_{y_k^T s_k} \end{cases} \tag{2.3}$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla E_{DeepFold}(x_{k+1}) - \nabla E_{DeepFold}(x_k)$. H_{k+1}^{-1} can be computed recursively by storing the previously calculated values of s_k and y_k. To preserve memory, L-BFGS only stores the last m values of s_k and y_k. Thus, H_{k+1}^{-1} is calculated by

$$H_{k+1}^{-1} = \left(\prod_{i=k}^{k-\widehat{m}+1} V_i^T \right) H_0^{-1} \left(\prod_{i=k-\widehat{m}+1}^{k} V_i \right) + \sum_{j=k}^{k-\widehat{m}+1} \left(\prod_{i=k+1}^{j+1} V_i \right) \rho_k s_k s_k^T \left(\prod_{i=j+1}^{k} V_i \right) \tag{2.4}$$

where $\widehat{m} = min(k, m-1)$ and m is set to 256 in DeepFold. Once the search direction d_k is decided, the torsion angles for the next step are updated according to

$$\begin{cases}\phi_{k+1} = \phi_k + \alpha_k d_k \\ \psi_{k+1} = \psi_k + \alpha_k d_k\end{cases} \tag{2.5}$$

The value of α_k is determined using the Armijo line search technique (185) and dictates the extent to move along the given search direction. In DeepFold, a maximum of 10 L-BFGS iterations are performed with 2,000 steps each, or until the simulations converge. The final model is selected as the one with the lowest energy produced during the folding simulations.

2.4 Author Contributions

The findings of this study were published in PLOS Computational Biology (122) with myself (R.P.) as first author, co-authors Drs. Yang Li (Y.L.) and Gilbert S. Omenn (G.S.O.), and corresponding author Dr. Yang Zhang (Y.Z.). R.P. developed DeepFold, performed the experiments, analyzed the data, developed the stand-alone package, and drafted the text and figures; Y.L. developed DeepPotential; R.P., G.S.O., and Y.Z. finalized the manuscript.

CHAPTER 3

DeepFoldRNA: *Ab Initio* RNA Tertiary Structure Prediction at Atomic Resolution Using Geometric Potentials from Deep Learning

Having covered our work in protein structure prediction, we will now turn our attention towards the related field of RNA structure prediction. Like proteins, RNAs are vital macromolecules that play a fundamental role in many cellular processes in living organisms, including mediating gene translation, serving as catalysts of important biological reactions, and regulating gene expression (186). Again, as is true for proteins, RNA functions are determined by their unique three-dimensional structures, which in turn are dictated by their nucleic acid sequences. Although understanding RNA structures is fundamental to elucidating their functions, there is an enormous discrepancy between the number of known RNA sequences and the number of solved structures. For example, while ~34 million RNA sequences have been deposited in the RNAcentral database (7), there are <500 non-redundant RNA structures solved in the Protein Data Bank (PDB) at a resolution of ~2 Å and <30 are composed of >70 nucleotides. Furthermore, only 99 of the 4,192 Rfam families have members with solved structures (87). Thus, there is an urgent need to develop computational RNA structure prediction methods capable of addressing this stark disparity.

Like with protein structure prediction, the goal of RNA structure prediction is to determine the spatial location of every atom in an RNA molecule starting from its nucleic acid sequence. Some state-of-the-art methods take a physics-based approach to model RNA structures by identifying low free-energy states through Monte Carlo simulations (187), while others approach the problem by assembling homologous fragments for a given nucleic acid sequence guided by knowledge-based energy functions (94). However, even with the assistance of human expert intervention and experimental data, these methods struggle to produce accurate folds for larger, more complex RNA molecules, rarely achieving RMSDs lower than 8-12 Å in blind RNA structure prediction studies (86, 101). Moreover, the results are typically worse for automatic modeling methods, which may produce models with around 20 Å RMSDs for complex folds (187). Progress has been made by using deep learning to predict secondary structure and contact information to guide the folding simulations (188-191); however, the improvements remain unsatisfactory and current state-of-the-art methods rarely achieve atomic resolution models, i.e., <2Å RMSD (192), for complex RNA folds (86, 94, 101, 102). Recently, deep learning approaches have been successfully applied to the problem of model selection (193). Nevertheless, the success of these methods is predicated on generating conformations that are close to the native structures, where atomic resolution was only obtained after utilizing restraints from native structures, which are not available in practical modeling applications.

To improve the performance of RNA structure prediction methods, we drew inspiration from our work with DeepFold and the dramatic advances in protein structure prediction made by AlphaFold2 and other self-attention-based methods (115, 123, 125). Toward this goal, we developed DeepFoldRNA, which uses a self-attention-based neural network architecture to predict geometric restraints, where 3D RNA structures are then built using L-BFGS minimization simulations. Across multiple test experiments, DeepFoldRNA drastically outperformed other state-of-the-art modeling methods and consistently achieved atomic-level resolution for complex RNA folds. In addition, due to the rapid gradient-based folding simulations, RNAs could be folded in a tiny fraction of the time required by current methods. The speed and accuracy of DeepFoldRNA will allow for large-scale elucidation of RNA structure and function, addressing a fundamental problem in structural biology. Each component of the program, including the deep learning models and L-BFGS optimization pipeline, is integrated into a stand-alone package at https://github.com/robpearc/DeepFoldRNA and an online webserver is available at https://zhanggroup.org/DeepFoldRNA, from which users can generate structure models for their own RNA of interest.

3.1 Results and Discussion

DeepFoldRNA is a method for fully-automated RNA structure prediction that consists of two consecutive modules (Fig. 3.1). In the restraint generation module (Fig 3.1.A), multiple sequence alignments (MSAs) of RNAs are collected by iteratively searching through multiple nucleic acid sequence databases using rMSA (194), where spatial restraints, including pairwise distance and inter-residue/backbone torsion angles maps, are predicted using self-attention neural networks that are built on two transformer elements with information encoded from the sequence, MSA, and pairwise positional embeddings. In the structure construction module (Fig 3.1B), the predicted geometric restraints are converted into composite potentials by taking the negative log-likelihood of the binned probability predictions, which are then used to guide the L-BFGS folding simulations.

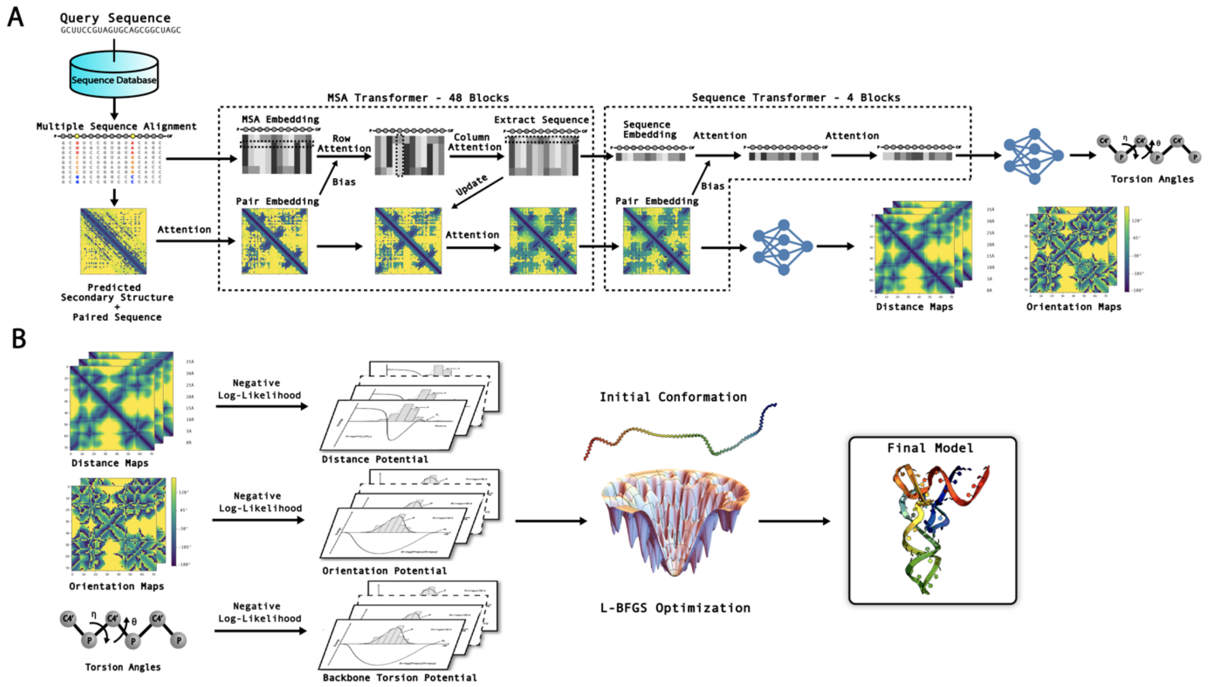


Figure 3.1 Overview of the DeepFoldRNA pipeline. A) Starting from a nucleic acid sequence, multiple RNA sequence databases are searched to create a multiple sequence alignment (MSA) for the query RNA, which is embedded into the network to initialize the MSA representation. The raw MSA is also used to derive the secondary structure prediction and initialize the pair embedding. The MSA and pair embeddings are then processed by the MSA Transformer layers, which use multiple self-attention mechanisms to refine the initial embeddings, where communication is encouraged between the two to ensure consistency. Next, the sequence embedding is extracted from the row in the final MSA embedding corresponding to the query sequence, which is further processed using self-attention mechanisms by the Sequence Transformer layers. Finally, the distance and inter-residue torsion angle maps are predicted from a linear projection of the final pair embedding, while the backbone pseudo-torsion angles are generated by a linear projection of the sequence embedding. B) The geometric restraints are converted into a negative-log likelihood potential to guide the L-BFGS simulations for final RNA model construction.

Two datasets were constructed to test DeepFoldRNA. The first was collected from Rfam families (87) with experimentally solved structures, where we curated a set of 4082 Rfam structures with complex folds and lengths between 70-250 nucleotides. From this set, we obtained 105 non-redundant RNA structures from 32 Rfam families after using a sequence identity cutoff of 80%. The second dataset was taken from the community-wide RNA-Puzzles experiment (85, 86, 101, 102) and consisted of 17 non-redundant, monomeric RNA structures where the models predicted by all groups were available to be downloaded at https://github.com/RNA-Puzzles/standardized_dataset. All targets in the test sets, together with those at >80% sequence identity to them, were held out from training the DeepFoldRNA pipeline.

3.1.1 DeepFoldRNA accurately predicts geometric restraints

Two central geometric restraints are predicted by DeepFoldRNA, including distance and orientation maps. The distance maps include the pairwise distances between the nitrogen atoms of the base bonded to the ribose sugar (N1 for pyrimidines and N9 for purines) as well as the backbone C4’ and P atoms (Fig. 3.2.A), while the inter-residue orientations include $\Omega =< C4'_i - N1/N9_i - N1/N9_j - C4'_j >$, $\lambda_i =< P_i - C4'_i - N1/N9_i - N1/N9_j >$, and $\lambda_j =< P_j - C4'_j - N1/N9_j - N1/N9_j >$, where *i* and *j* are the nucleotide indices along the sequence (Fig. 3.2.B). The network of Module-1 generates probability distributions for each of the geometric restraints, where the distances and orientations are divided into 40 and 25 bins, respectively (see Methods).

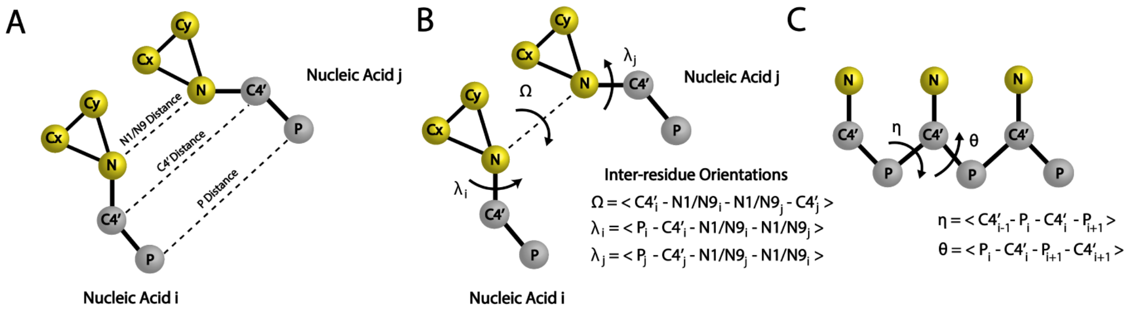

A
Cy
Cx
N
Cy
Cx
N
C4′
P
C4′
P
N1/N9 Distance
C4′ Distance
P Distance
Nucleic Acid j
Nucleic Acid i
B
Cy
Cx
N
λj
C4′
P
Ω
Cy
Cx
N
λi
C4′
P
Nucleic Acid j
Nucleic Acid i
Inter-residue Orientations
Ω = < C4′i - N1/N9i - N1/N9j - C4′j >
λi = < Pi - C4′i - N1/N9i - N1/N9j >
λj = < Pj - C4′j - N1/N9j - N1/N9i >
C
N
N
N
C4′
C4′
C4′
η
θ
P
P
P
η = < C4′i-1 - Pi - C4′i - Pi+1 >
θ = < Pi - C4′i - Pi+1 - C4′i+1 >

Figure 3.2 Definition of the geometric restraints predicted by DeepFoldRNA. These restraints include (A) inter-residue distances, (B) inter-residue torsion angles, and (C) backbone pseudo-torsion angles.

To assess the accuracy of the predicted restraints, we list in Table 3.1 the Mean Absolute Errors (MAEs) for the top L, $5L$ and $10L$ medium/long-range ($|i - j|$>12) distance and orientation restraints predicted by DeepFoldRNA for the 122 RNAs in the two test sets. Here, MAE = $(1/n)\sum_{i=1}^{n}|x_i - y_i|$, where x_i is the value of the predicted restraint with the maximum probability score for a selected residue pair, y_i is the corresponding value in the native structure, and n is the number of restraints considered. As a control, we also list the distance/orientation parameters taken from the predicted models by two state-of-the-art modeling methods: SimRNA (187) and Rosetta FARFAR2 (94), which have been among the most accurate automatic modeling servers in previous RNA-Puzzles experiments (86, 101, 102). To provide a fair comparison between the methods, the predicted secondary structures used by DeepFoldRNA were used as constraints during the SimRNA and FARFAR2 simulations, where the exact procedures used to run both programs are provided in Texts F.1 and F.2 in Appendix F. Overall, DeepFoldRNA produced accurate distance and orientation predictions, where the average MAEs for the top L, $5L$ and $10L$ N1/N9 distances were 0.72, 0.83 and 0.93 Å, respectively, which were ~9-11 times lower than those extracted from the SimRNA and FARFAR2 models. For the Ω/λ orientations, the average L, $5L$ and $10L$ MAEs were 0.17/0.14, 0.20/0.16 and 0.23/0.17 radians, respectively, which were around 4-6.5 times lower than those obtained from the SimRNA and FARFAR2 models. These data demonstrate the ability of DeepFoldRNA to create very accurate restraint predictions, which are crucial to its modeling performance.

It is noted that since the SimRNA and FARFAR2 models do not have confidence scores associated with each distance/orientation, we selected restraints based on the DeepFoldRNA confidence scores alone in the above comparisons. To remove the bias in restraint selection, we present a comparison for all medium/long-range restraints at the last column of Table 3.1. As expected, the MAE was much larger for the DeepFoldRNA restraints when all residues were considered, suggesting the sensitivity of the DeepFoldRNA confidence scores and the rationality for the 3D model construction based on a limited number of high-ranking restraints. Interestingly, the MAEs of the SimRNA and FARFAR2 models were also typically smaller (except for the λ orientation) for the top-ranked residues than for all residues; this is probably because DeepFoldRNA tends to have higher confidence scores in conserved regions where SimRNA and FARFAR2 could also generate slightly better models.

Restraint Type	**Method**	**Top-L (*p*-value)**	**Top-5L (*p*-value)**	**Top-10L (*p*-value)**	**All (*p*-value)**
N1/N9 Distance MAE (Å)	SimRNA Model	6.52 (2.7E-27)	7.47 (5.4E-30)	8.06 (2.9E-27)	10.71 (5.1E-10)
	FARFAR2 Model	8.17 (1.6E-25)	9.34 (7.0E-32)	9.91 (1.1E-32)	12.01 (5.6E-20)
	DeepFoldRNA	**0.72 (-)**	**0.83 (-)**	**0.93 (-)**	**6.37 (-)**
C4' Distance MAE (Å)	SimRNA Model	9.29 (1.8E-30)	9.38 (4.0E-35)	9.44 (9.2E-33)	10.86 (4.8E-11)
	FARFAR2 Model	11.37 (2.4E-29)	11.40 (2.8E-35)	11.27 (2.2E-36)	12.00 (1.5E-20)
	DeepFoldRNA	**0.88 (-)**	**0.97 (-)**	**1.06 (-)**	**6.80 (-)**
P Distance MAE (Å)	SimRNA Model	8.37 (7.0E-36)	8.61 (3.3E-38)	8.87 (3.4E-33)	11.49 (1.1E-12)
	FARFAR2 Model	10.25 (1.3E-28)	10.76 (7.6E-37)	11.13 (2.8E-38)	14.27 (8.6E-28)
	DeepFoldRNA	**0.87 (-)**	**0.98 (-)**	**1.1 (-)**	**6.84 (-)**
Ω Orientation MAE (radians)	SimRNA Model	0.83 (1.9E-52)	0.87 (2.1E-61)	0.88 (4.9E-61)	0.94 (4.7E-55)
	FARFAR2 Model	0.90 (2.4E-62)	0.91 (2.0E-73)	0.91 (1.0E-74)	0.98 (2.3E-65)
	DeepFoldRNA	**0.17 (-)**	**0.20 (-)**	**0.23 (-)**	**0.43 (-)**
λ Orientation MAE (radians)	SimRNA Model	0.88 (2.3E-38)	0.82 (1.2E-47)	0.78 (1.2E-51)	0.77 (6.3E-48)
	FARFAR2 Model	0.85 (2.2E-38)	0.81 (1.7E-52)	0.79 (1.1E-56)	0.79 (1.8E-51)
	DeepFoldRNA	**0.14 (-)**	**0.16 (-)**	**0.17 (-)**	**0.37 (-)**

Table 3.1 Summary of the accuracy of the DeepFoldRNA predicted restraints. The accuracy is analyzed in terms of the Mean Absolute Errors (MAEs) for the top medium/long-range ($|i-j|$>12) restraints, where L is the RNA length. The p-values were calculated between DeepFoldRNA and the control methods using paired, two-sided Student's t-tests.

3.1.2 DeepFoldRNA dramatically outperforms state-of-the-art methods on the Rfam dataset

To evaluate the modeling performance of DeepFoldRNA, Table 3.2 presents a summary of the 3D modeling results on the 105 RNAs from the Rfam dataset in terms of the average/median RMSDs and TM-scores relative to the experimental structures along with the results by SimRNA and FARFAR2. As a reminder, TM-score is a length-independent metric for assessing structural similarity that takes a value in the range (0, 1], where a TM-score=1 corresponds to an identical structural match and a TM-score >0.45 indicates that two RNAs share the same global fold (170, 195).

Method	RMSD Avg/Median (*p*-value)	TM-score Avg/Median (*p*-value)	Correct Folds*	RMSD$_{\text{DeepFoldRNA}}$ <RMSD$_{\text{Method}}$‡
SimRNA	19.37/17.23 (6.2E-44)	0.228/0.222 (1.0E-66)	0.0%	100.0%
FARFAR2	21.07/19.17 (8.4E-52)	0.228/0.219 (6.6E-64)	0.0%	100.0%
DeepFoldRNA	**2.68/2.11 (-)**	**0.757/0.779 (-)**	**99.0%**	**-**

** This column represents the percent of RNAs with TM-scores ≥0.45.*

‡ This column indicates the percent of test RNAs for which DeepFoldRNA generated a model with a lower RMSD than the control method.

Table 3.2 Summary of the structure modeling results by DeepFoldRNA compared to the control methods on the 105 test RNAs from the Rfam dataset. The RMSDs and TM-scores were calculated using the RNA-align program(195) based on sequence-dependent superposition of the C3’ atoms. The *p*-values were calculated between DeepFoldRNA and the control methods using paired, two-sided Student’s t-tests.

On average, DeepFoldRNA achieved a TM-score of 0.757, which was 232% higher than that attained by SimRNA and FARFAR2 (0.228); the differences were highly statistically significant with *p*-values of 1.0E-66 and 6.6E-64 for the comparison with SimRNA and FARFAR2, respectively. Meanwhile, the average RMSD of the DeepFoldRNA models was 2.68 Å compared to 19.37 Å and 21.07 Å for SimRNA and FARFAR2, respectively; the differences were again statistically significant with *p*-values of 6.2E-44 and 8.4E-52. When considering the median values, DeepFoldRNA produced models with a median RMSD of 2.11 Å (SimRNA: 17.23 Å; FARFAR2: 19.17 Å) and a median TM-score of 0.779 (SimRNA: 0.222; FARFAR2: 0.219), corresponding to close atomic matches between the predicted and native structures.

In Fig. 3.3, we present head-to-head RMSD and TM-score comparisons of DeepFoldRNA with SimRNA and FARFAR2. Overall, DeepFoldRNA generated models with lower RMSDs and higher TM-scores than the control methods for all of the test RNAs. Furthermore, Fig. 3.3.C and 3.3.F list the number of models produced below a specific RMSD or above a given TM-score threshold. When considering a cutoff TM-score of 0.45, for example, DeepFoldRNA generated correct global folds for 99% or all but one of the test RNAs, while the control methods were unable to generate correct global folds for any of the targets. DeepFoldRNA also consistently generated models with atomic-level accuracy, where 46 of the 105 models (43.8%) had RMSDs <2 Å to their experimental structures. When considering a more permissive RMSD cutoff of <4.0 Å to define a native-like structure, 86.7% of the DeepFoldRNA models met this criterion, while none of the models by the control methods did so.

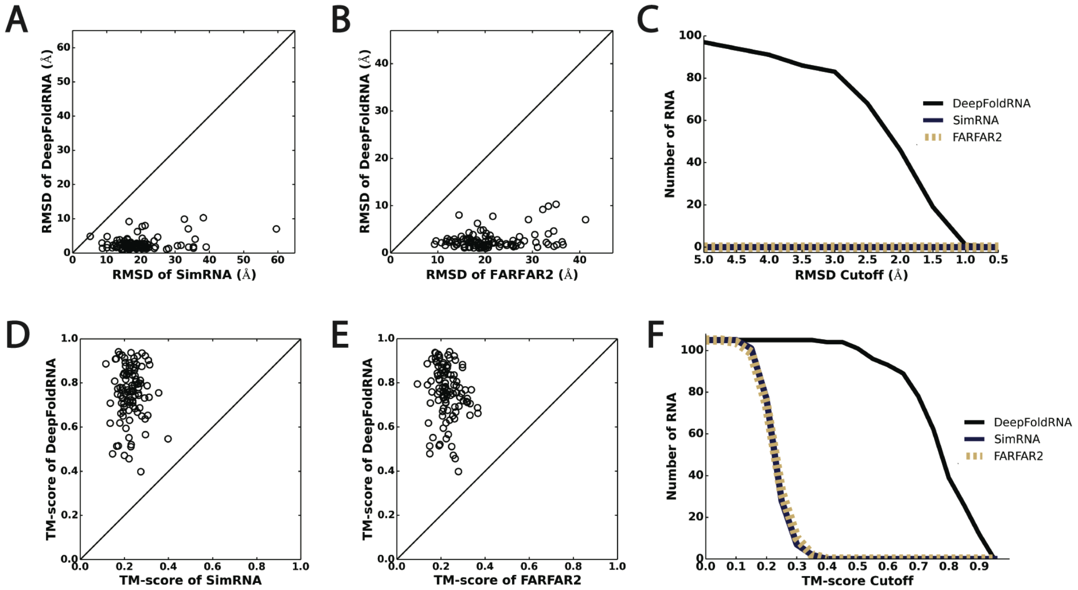


Figure 3.3 Head-to-head RMSD and TM-score comparisons of DeepFoldRNA with the selected state-of-the-art methods on the 105 Rfam RNA strucures. A) RMSD comparison with SimRNA, B) RMSD comparison with FARFAR2, C) Number of targets below a given RMSD threshold, D) TM-score comparison with SimRNA, E) TM-score comparison with FARFAR2, F) Number of targets above a given TM-score threshold.

Importantly, the success of DeepFoldRNA modeling was not limited to any specific fold type. Fig. 3.4 plots representative models across all 32 Rfam families in the test set. For 14 of the 32 families (43.8%), DeepFoldRNA generated atomic resolution models with <2Å RMSD and 100% of the models possessed correct global folds with TM-scores >0.45. Highly accurate models could be constructed for well represented families such as RF00001 (composed of 5S ribosomal RNAs) and RF00005 (made up of tRNAs), where the DeepFoldRNA models had RMSDs of 1.08 Å and 1.09 Å, respectively, corresponding to very close atomic matches between the predicted and experimental structures. Accurate models were also constructed for families with few sequence homologs. For instance, RF01689 (PDB ID 4frg, chain B, residues 1-83) is composed of AdoCbl variant RNAs and the generated MSA had relatively few homologous sequences with a Neff value (number of effective sequences) of 3.65, where DeepFoldRNA created an accurate model for this family with an RMSD of 2.12 Å and a TM-score of 0.709.

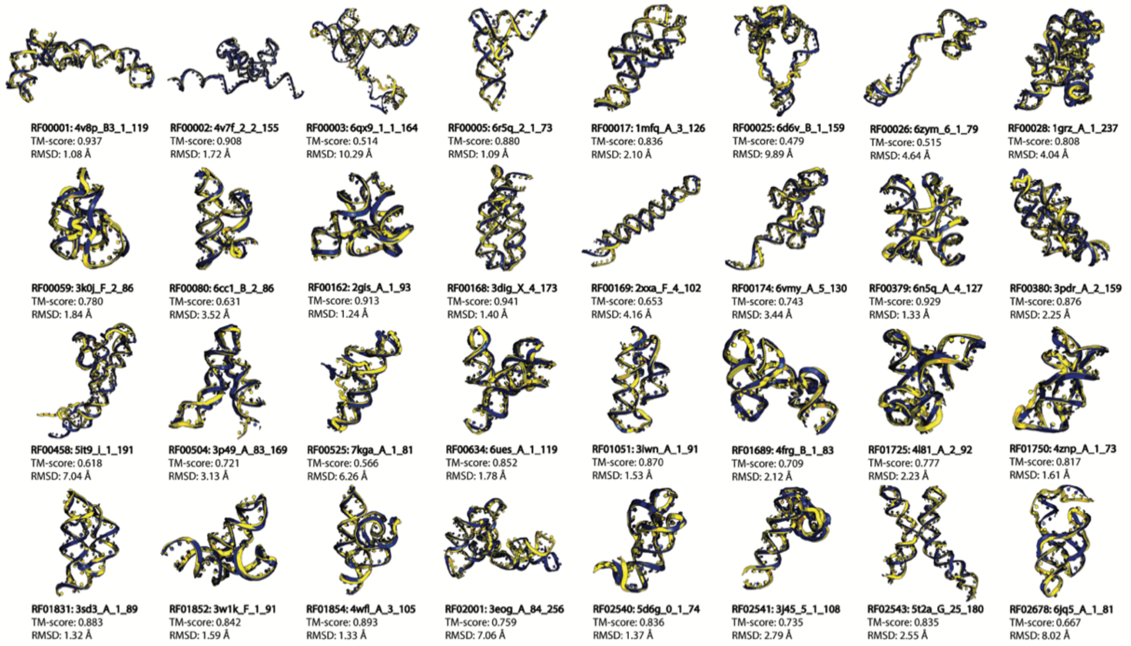


Figure 3.4 Representative models generated by DeepFoldRNA for each of the 32 Rfam families. The modeled structures in blue are superposed with the native structures in yellow. The PDB IDs, chain ids, and residue numbers are shown below each RNA together with the TM-scores and RMSDs.

Interestingly, for models with higher RMSDs, the modeling errors were often localized in flexible or unstructured regions of the RNAs. For instance, for RF02678 (PDB ID 6jq5, chain A, residues 1-81) the model generated by DeepFoldRNA had an RMSD of 8.02Å, where the deviation between the modeled and native structures was mainly confined to the unpaired region of the structure from residues 64-81 (Fig. E.1 in Appendix E). In the core region of the RNA (residues 1-63), in contrast, the RMSD between the modeled and native structures was only 1.40 Å, resulting in a correct global fold with a TM-score of 0.667. Overall, the results demonstrate that DeepFoldRNA is able to consistently generate correct global folds, frequently with atomic-level resolution, for RNAs across various complex fold types, drastically outperforming the leading Monte Carlo simulation methods.

3.1.3 DeepFoldRNA outperforms the best models by the RNA-Puzzles community by a large margin

To further examine DeepFoldRNA with the state of the art, we tested it on 17 challenging, monomeric RNA targets from the community-wide RNA-Puzzles experiment, where many of the targets lacked structural and sequence homologs (85, 86, 101, 102). The experiment is split into Human and Server Sections, where each group is allowed to submit up to 10 models for each target. Traditionally, the automated Server methods, which are given 48 hours to model a target, have been unable to achieve the same performance as Human groups, who are typically given 3-6 weeks and often utilize extensive expert intervention during the modeling process and restraints from fast-track experimental data (86, 101, 102). Fig. 3.5 summarizes the modeling results of DeepFoldRNA compared to all RNA-Puzzles participants.

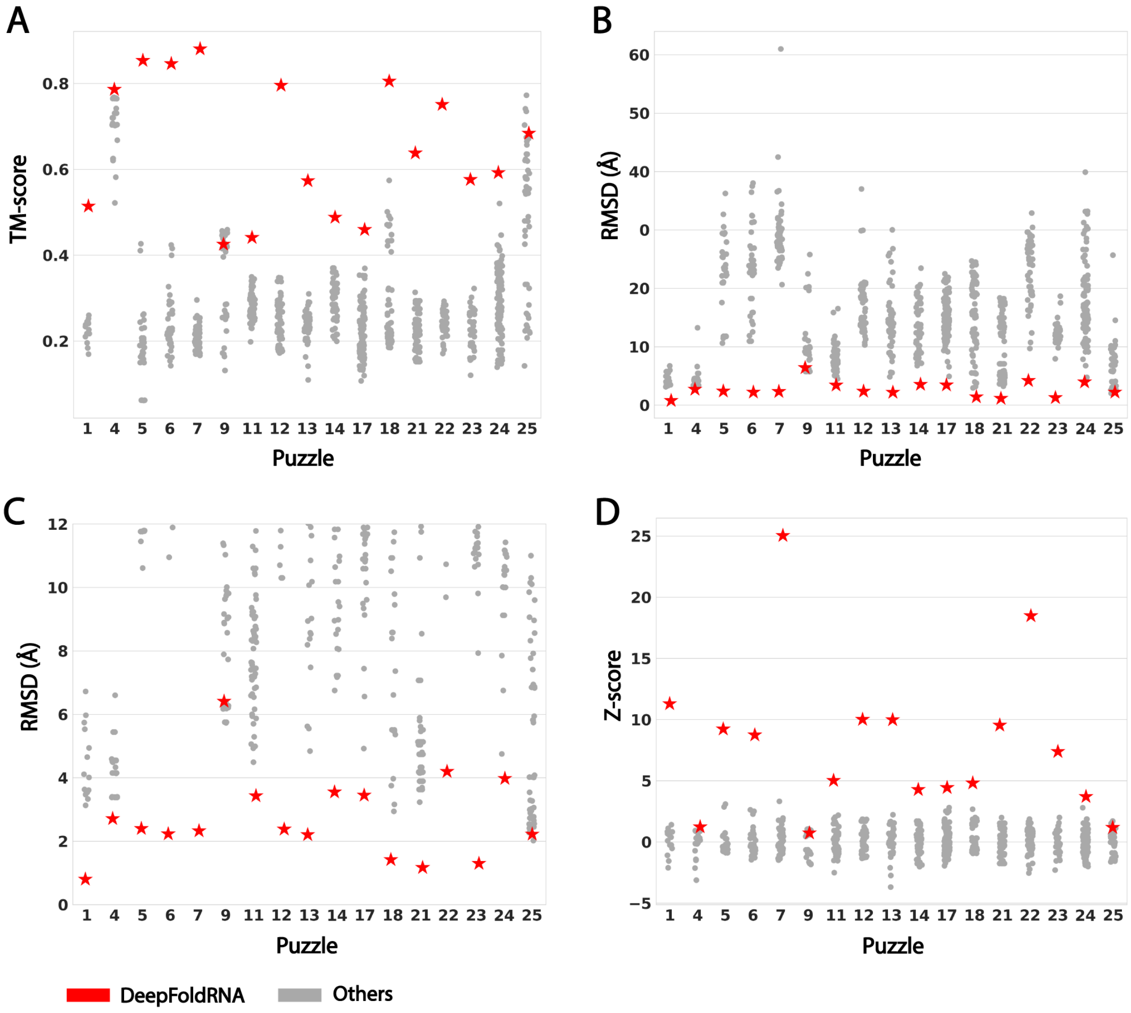


Figure 3.5 DeepFoldRNA modeling results on the 17 RNA-Puzzles targets compared to the participants. (A) TM-score; (B) RMSD; (C) Same as (B) but only for models with RMSDs below 12 Å. (D) Z-score of the TM-score for DeepFoldRNA compared to the participating groups. The RMSDs and TM-scores were calculated using the RNA-align program(195) based on sequence-dependent superposition of the C3' atoms.

Overall, DeepFoldRNA achieved an average TM-score of 0.654, which was 77.7% higher than the average TM-score of the first models generated by the best-performing group in RNA-Puzzles (Das Group, TM-score=0.368). If we select the best model submitted for each target by all of the RNA-Puzzles participants, the DeepFoldRNA TM-score was still 55.0% higher than that of the best models by the community (TM-score=0.422). Similarly, the average RMSD of DeepFoldRNA (2.72 Å) was 4.18 Å lower than the average RMSD of the best models (6.90 Å) generated by all RNA-Puzzles groups. When considering a cutoff TM-score of 0.45, DeepFoldRNA generated correct global folds for 15 of the 17 Puzzles (88.2%), while correct global folds could only be constructed for 5 of the 17 targets (29.4%) by the RNA-Puzzles community. Meanwhile, DeepFoldRNA generated models with <2.5 Å RMSD for 10 of the 17 cases (58.8%), while this accuracy was achieved for only one target (Puzzle 25) by the community.

Since many of the RNA-Puzzles targets lack sequence homologs, it is of interest to examine the modeling performance in relation to the quality of the generated MSAs. In Fig. 3.6.A, we plot the TM-score of the DeepFoldRNA models against the logarithm of the MSA Neff value on the RNA-Puzzles dataset. From the figure, it can be seen that there is essentially no correlation (ρ=-0.001) between the model TM-score and the MSA Neff value, suggesting that DeepFoldRNA is a robust method for the hardest class of RNA targets, which lack homologous sequence information. Furthermore, Fig. 3.6.B plots the model TM-scores vs. the MSA Neff values across the targets in both the RNA-Puzzles and Rfam datasets. Again, only a very weak correlation (ρ=0.013) existed between the Neff value and the model quality by DeepFoldRNA. Overall, these results demonstrate that DeepFoldRNA is capable of accurately folding very challenging modeling targets using a fully automated pipeline, significantly outperforming approaches from the RNA-Puzzles challenge, where many of the predictions were guided by human expert intervention and experimental restraints.

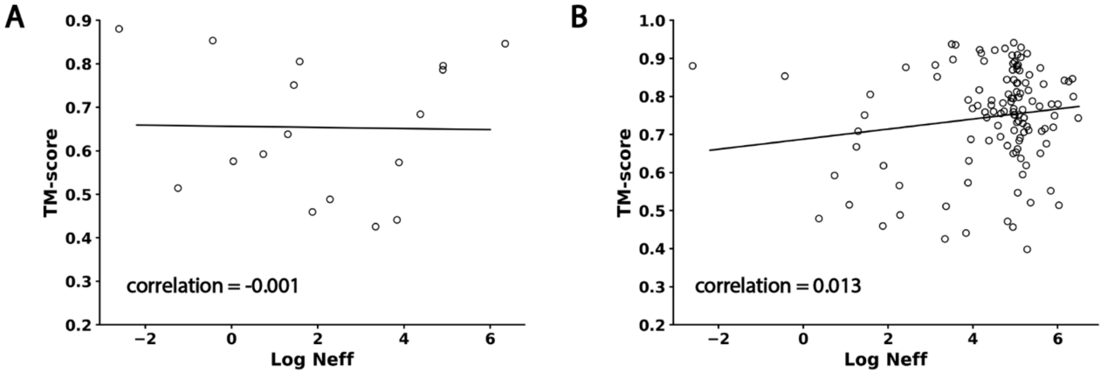


Figure 3.6 Model TM-score vs. the logarithm of the MSA Neff value for DeepFoldRNA on the RNA-Puzzles dataset (A) and the overall dataset (B). The fitted models were obtained by linear regression.

3.1.4 Case studies reveal DeepFoldRNA's ability to fold challenging targets with complex structures

A closer examination of Fig. 3.5 shows that DeepFoldRNA achieved the best models with the highest TM-scores and lowest RMSDs for 15 out of the 17 RNA targets. If we define a $\mathrm{Z-score} = (\mathrm{TM_D} - \langle \mathrm{TM} \rangle)/\sigma$, where $\mathrm{TM_D}$ is the TM-score of the DeepFoldRNA model, $\langle \mathrm{TM} \rangle$ is the average TM-score of all groups and σ is the standard deviation, DeepFoldRNA generated a model that was better than any other submitted model by a large margin for 10 cases (i.e., Puzzles 1, 5, 6, 7, 11, 12, 13, 21, 22, and 23) with Z-scores above 5 (Fig. 3.5.D). There were only two targets (PZ9 and PZ25) for which the DeepFoldRNA model was marginally worse with RMSDs that were 0.2 and 0.66 Å higher than the best models from the RNA-Puzzles community, respectively.

In Fig. 3.7, we present four case studies for which DeepFoldRNA achieved near-native quality models with RMSDs <2.5 Å, while all models submitted by the RNA-Puzzles community had RMSDs above 10 Å.

First, **Puzzle 5** is a 188-nucleotide long lariat-capping ribozyme (PDB ID: 4p9r) that catalyzes reactions involving the formation of a 3 nucleotide 2',5' lariat (196). The RNA possesses a unique open ring structure formed by the interaction between the two peripheral helical regions. The highest TM-score model submitted by the RNA-Puzzles community had a TM-score of 0.426 and an RMSD of 10.61Å, where the open ring structure was not reproduced by any of the submitted models (102). For this target, the generated MSA by rMSA contained 17 sequence homologs, where only 3 sequences were aligned to the query with a coverage >50%, resulting in a low Neff value of 0.65. Nevertheless, the deep learning module generated accurate spatial restraints with

MAEs for the top 5L N-N distances and Ω/λ orientations of 0.99 Å and 0.17/0.14 radians, respectively. Additionally, the structure produced by the folding simulations closely converged to the predicted restraints with an MAE of 0.72 Å between the predicted top 5L N-N distances and the model distances. This resulted in a high-quality 3D structure with a TM-score of 0.851 and RMSD of 2.43 Å, accurately recapitulating the open ring structure and again highlighting the ability of DeepFoldRNA to model challenging targets with few homologous sequences.

Second, **Puzzle 6** is a 168-nucleotide adenosylcobalamin riboswitch (PDB ID: 4gxy), which possesses a large ligand binding pocket that binds the adenosyl moiety to control gene expression (197). The models submitted by the RNA-Puzzles community had a wide range of TM-scores (~0.142-0.424) and RMSDs (~38.02-11.89 Å), where the best model (TM-score=0.424) was produced with the assistance of experimental SHAPE data to help elucidate important secondary structure and contact information (102). For DeepFoldRNA, a reliable MSA was collected with a high Neff of 517.9, which resulted in accurate predicted restraints with MAEs for the top 5L N-N distances and Ω/λ orientations of 0.94 Å and 0.22/0.17 radians, respectively. Moreover, the folding simulations produced a structure that closely matched the predicted restraints with an MAE of 0.73 Å between the top 5L predicted N-N distances and the model distances. Thus, the generated model possessed a near-native structure with a TM-score of 0.846 and RMSD of 2.23 Å. Importantly, the ligand binding site, which is essential to the RNA's function, was accurately recapitulated without any explicit provisions or simulations that accounted for the ligand position.

Third, **Puzzle 7** is the Varkud satellite ribozyme (PDB ID: 4r4v), which is composed of 185 nucleotides and mediates rolling circle replication of a plasmid in the *Neurospora* mitochondrion (198). The highest TM-score RNA-Puzzles model was constructed with the assistance of hydroxy radical footprinting experiments as well as mutate-and-map measurements used to determine contact information (101). Nevertheless, the resulting model had a low TM-score of 0.295 and a high RMSD of 25.33 Å, where the model possessed incorrect helical orientations and an overly compact structure. For this target, DeepFoldRNA generated a poor MSA containing only 3 sequence homologs, all of which were nearly identical to the query sequence, resulting in an extremely low Neff of 0.07, making the prediction essentially a single sequence prediction problem. Nevertheless, the deep learning module produced accurate restraints with MAEs for the top 5L N-N distances and Ω/λ orientations of 0.77 Å and 0.16/0.14 radians, respectively. This resulted in a DeepFoldRNA model with a TM-score of 0.875 and RMSD of 2.40 Å, corresponding to a 196.6% higher TM-score than the best model submitted during RNA-Puzzles.

Last, **Puzzle 12** is a medium-size (108 nucleotides) *ydaO* riboswitch (PDB ID: 4qlm) with a novel structural topology that contains two binding pockets for cyclic-di-AMP (199). It is involved in a number of important cellular functions, including sporulation, osmotic stress responses, and cell wall metabolism (199). The best RNA-Puzzles model was produced with the assistance of fast-track experimental SHAPE data and multidimensional chemical mapping (101) and had a TM-score of 0.347 and RMSD of 14.35Å. Notably, the bubble region in the structure was unable to be correctly predicted by any of the submitted models and is partially unresolved in the crystal structure, likely due to its flexibility and lack of base pairing (101). For DeepFoldRNA, the generated MSA was reliable with a Neff value of 135.5 and the resulting predicted restraints were accurate with MAEs of 0.70 Å and 0.15/0.20 radians for the top $5L$ N-N distances and Ω/λ orientations, respectively. Again, the folding simulations closely converged to the predicted restraints with an MAE of 0.72 Å between the predicted top $5L$ N-N distances and the model distances. These resulted in a high-quality model by DeepFoldRNA with an RMSD of 2.38 Å and a TM-score of 0.796 to the experimental structure, corresponding to a 129.4% improvement in the TM-score over the best model produced during the RNA-Puzzles challenge.

The results on these case studies demonstrate that DeepFoldRNA is able to produce accurate structural models for challenging RNAs that could not be folded by any traditional approach even with expert intervention and experimental restraints. It is practically encouraging that medium to higher resolution structures could be created for complex folds with few homologous RNA sequences, which has been one of the most challenging problems for deep learning-based protein structure modeling methods (123, 125, 164). This is probably due to the fact that, compared to proteins whose structural patterns are often buried in deep evolutionary profiles, RNA structures are more explicitly encoded in the individual nucleic acid sequences (e.g., the tertiary structures are highly dependent on the Watson-Crick pairing of the RNA sequence), which can be readily captured by advanced deep learning models even with relatively shallow sequence profiles.

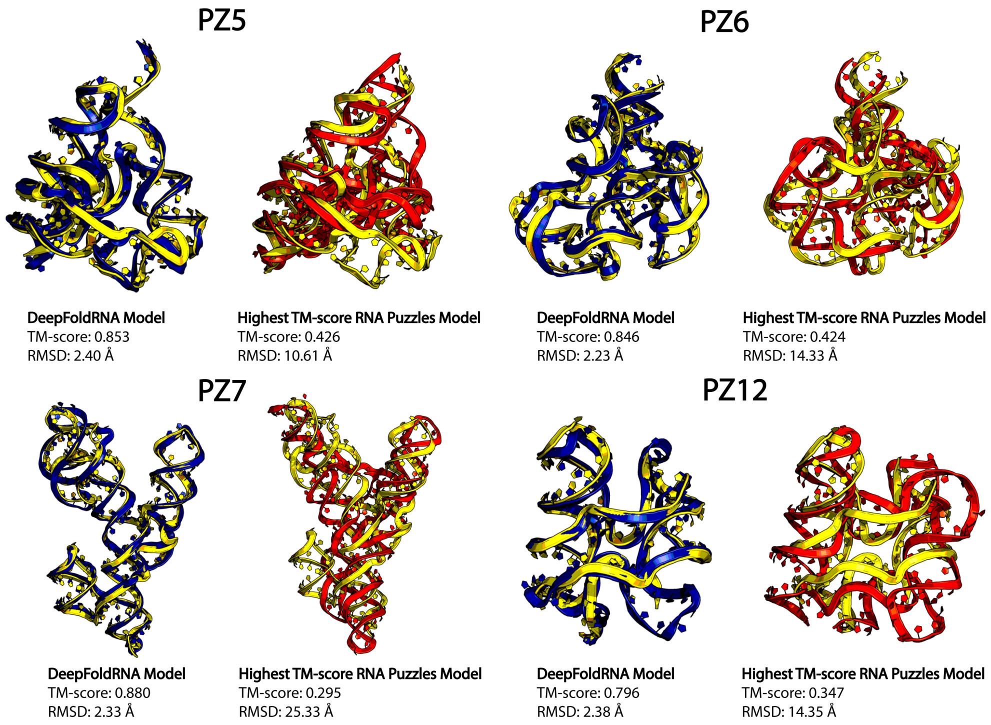


Figure 3.7 Case studies from difficult RNA-Puzzles targets. The native structures are shown in yellow, and the structures by DeepFoldRNA and the best RNA-Puzzles models are shown in blue and red, respectively.

3.1.5 DeepFoldRNA improves the speed and accuracy of RNA folding simulations for large RNAs

Monte Carlo sampling is a widely used approach in structural folding simulations and has been proven to be efficient at identifying global free-energy minima for cases with frustrated knowledge-based energy landscapes (14, 25, 187). However, these simulations typically require lengthy runtimes, which partially limits their application to large-scale modeling experiments. Given that the DeepFoldRNA energy landscape is significantly simplified by accurate and abundant spatial restraints, gradient-based L-BFGS sampling is sufficient to quickly fold RNA molecules and drastically reduce the simulation runtime.

As evidence, we plot in Fig. 3.8.A the simulation time required for DeepFoldRNA, SimRNA and FARFAR2 against the RNA length, where both SimRNA and FARFAR2 are based on Monte Carlo sampling. Overall, SimRNA required 379.3 minutes on average to fold the RNAs in the Rfam dataset, while DeepFoldRNA required 1.1 minutes, corresponding to a 345-fold reduction of the folding simulation time. The difference was even more significant when compared to FARFAR2, which required 4547.1 minutes for its folding simulations on average. Notably, DeepFoldRNA could fold the largest RNA in the dataset, which was composed of 237 nucleotides, within 7 minutes, while SimRNA and FARFAR2 required 1146 and 11615 minutes, respectively. Thus, DeepFoldRNA can be used to fold RNA molecules in seconds to minutes, significantly improving the speed at which RNAs can be modeled.

Crucially, the modeling performance of DeepFoldRNA did not deteriorate as the sequence length of the RNA increased. In Fig. 3.8.B, we plot the TM-score values for the models generated by DeepFoldRNA, SimRNA, and FARFAR2 against the RNA sequence length. As expected, there was a negative correlation between the RNA length and model TM-score for both SimRNA and FARFAR2, as larger RNAs often have more complex folds that require sampling from wider-ranging conformational space, which is more difficult for Monte Carlo sampling to cover when guided by low-resolution energy force fields. For DeepFoldRNA, however, there was actually a slight positive correlation, where the method was able to generate on average more accurate spatial restraints and reliable folds for longer RNAs in the test set. These data demonstrate that the rapid simulations do not lead to unreliable results for larger and more complex folds, making DeepFoldRNA a robust method for generating accurate models independent of the fold complexity, which is critical for applications to large-scale RNA structure modeling.

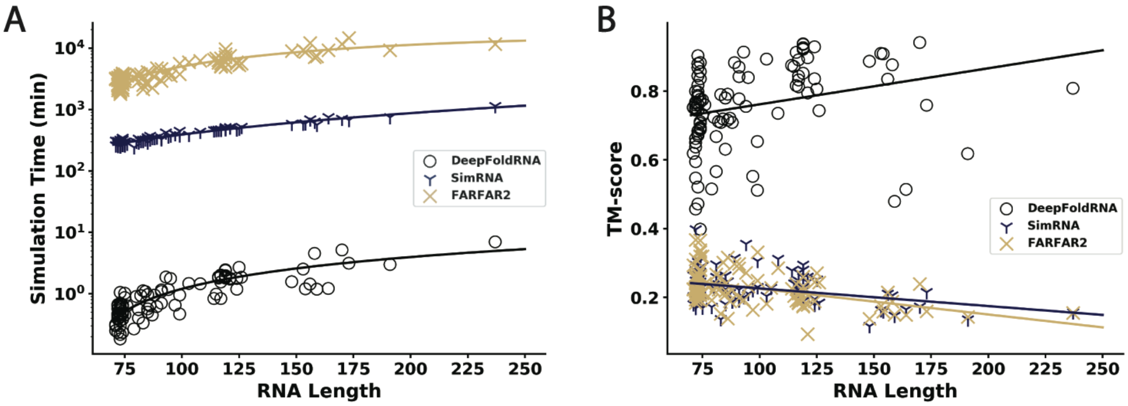


Figure 3.8 Dependence of the simulation runtime/modeling performance on the RNA length for DeepFoldRNA, SimRNA, and FARFAR2. A) Log-scale simulation runtime for DeepFoldRNA, SimRNA, and FARFAR2 in minutes plotted against the RNA length. B) Model TM-score versus RNA length for DeepFoldRNA, SimRNA, and FARFAR2. Lines are plotted to guide the eye.

3.2 Concluding Remarks

Inspired by our work with DeepFold and the latest advances in protein structure prediction, we developed a fully-automated method, DeepFoldRNA, to model RNA structures starting from sequence alone. The approach is built on deep self-attention neural networks to deduce high-accuracy spatial restraints from multiple RNA sequence alignments, followed by full-length 3D model construction through restraint-guided L-BFGS folding simulations.

The method was tested on two independent benchmark datasets. The first consisted of 105 non-redundant RNAs from 32 Rfam families with complex global folds. For these targets, DeepFoldRNA generated models with an average TM-score of 0.757 and RMSD of 2.68 Å, which was dramatically more accurate than the state-of-the-art methods, SimRNA (187) and FARFAR2 (94), which are physical and knowledge-based Monte Carlo approaches that produced models with an average TM-score/RMSD of 0.228/19.37 Å and 0.228/21.07 Å, respectively. For the second benchmark dataset containing 17 challenging targets from the community-wide RNA-Puzzles experiment, DeepFoldRNA constructed models with better quality than the best models submitted from the community for 15 cases, where there was a large margin in the TM-score/RMSD difference for 10 cases, despite the fact that many models from the community were constructed with human expert intervention and experimental restraints (86, 101, 102).

These improvements demonstrate the power and advantage of deep self-attention neural networks, which can learn more detailed structural information from evolutionary profiles than knowledge-based potentials derived from statistical analyses of PDB structures. The success of DeepFoldRNA modeling exhibited little correlation to the quality of the input MSAs, in part due to the effectiveness of self-attention networks, which are able to learn structural patterns embedded in single RNA sequences. Meanwhile, given the abundant, high-accuracy restraints produced by the deep learning modules, which can dramatically smooth the energy landscape, an additional advantage of DeepFoldRNA is its rapid model construction enabled by the gradient-based folding simulations. On average, DeepFoldRNA only required 1.1 minutes to fold the Rfam RNAs, which was 345 times faster than SimRNA and 4134 times faster than FARFAR2.

The high accuracy and speed of DeepFoldRNA, together with its fully-automated procedure, should help facilitate its usefulness and application to large-scale, atomic-level RNA structure modeling. Currently, only 2% of Rfam families have experimentally solved structures, where the application of DeepFoldRNA to model unknown Rfam families will provide critical information and insight into uncharacterized RNA structure space. Furthermore, the extension of the deep neural network models to RNA complexes and RNA-protein interactions will help elucidate the molecular and cellular functions of non-coding RNAs. Studies along these lines are in process.

3.3 Methods

DeepFoldRNA is a deep learning-based approach to full-length RNA structure modeling, which consists of three main steps: input feature generation, spatial restraint prediction, and L-BFGS folding simulations, as depicted in Fig. 3.1.

3.3.1 Input feature generation

DeepFoldRNA takes as input the nucleic acid sequence in FASTA format for the RNA of interest, from which all features used by the neural network are derived. The major input to the network is the MSA generated by rMSA (194). Briefly, rMSA constructs an MSA for a query sequence by iteratively searching multiple nucleic acid sequence databases, including Rfam (87), RNAcentral (7), and the nt database (200) using blastn (201), nhmmer (202), and cmsearch (202) (Fig. 3.9). From the MSA, nhmmer is used to generate a hidden Markov model (HMM), which serves as a succinct statistical representation of the detected homologous sequence profile. Additionally, PETfold (203) is used to predict the secondary structure from the generated MSA, where the pairwise reliability scores are used as the input of the network. This allows for a convenient embedding of the secondary structure information, while simultaneously capturing the uncertainty in the PETfold predictions.

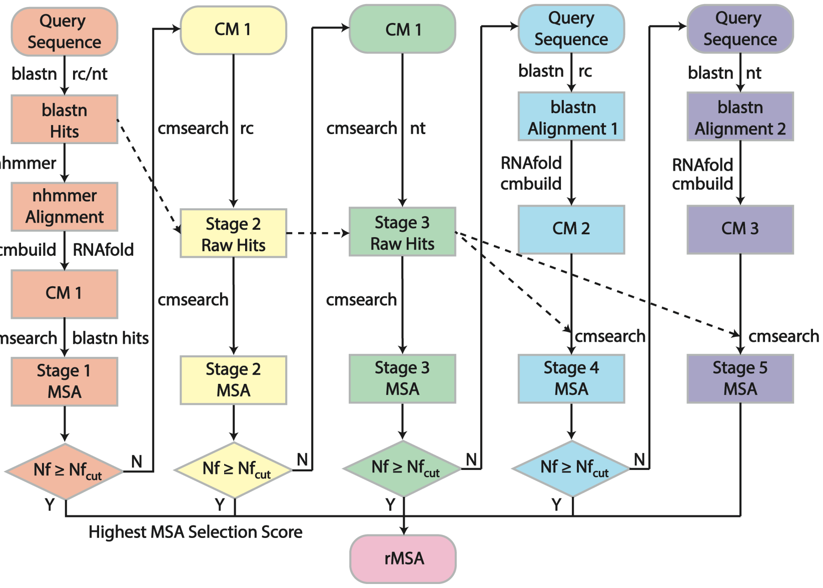


Figure 3.9 rMSA pipeline overview. rMSA generates 5 MSAs in total, where "CM" and "rc" stand for Covariance Model and the RNAcentral database, respectively, and Nf_{cut}=128. The blastn searches are performed with "-max_target_seqs 50000 -strand plus" and "-max_target_seqs -strand both" options to search only the plus strand through RNAcentral and both strands through the nt

database, respectively. This is due to the fact that RNAcentral is a transcriptomics database, while nt is a genomics database. Similarly, cmsearch is performed using the "--toponly --incE 10.0" option for the plus strand in RNAcentral and "--incE 10.0" for both strands in nt. The nhmmer search is performed using "--watson" to only consider alignments with directions that are consistent with the blastn alignments.

3.3.2 Network Architecture

Overall Architecture. The overall network architecture is depicted in Fig. 3.1.A. The generated features are first embedded into the network and passed through 48 MSA Transformer blocks, which use multiple self-attention layers to extract information encoded in the MSA to determine the spatial relationships between each pair of positions. Meanwhile, a pair representation is embedded into the network, where communication is encouraged between the MSA and pair representations using biased self-attention as well as updating the pair representation based on the processed MSA representation. Following this step, the sequence embedding is extracted from the MSA representation based on the position in the MSA embedding that corresponds to the original query sequence. The sequence embedding is then processed by 4 Sequence Transformer blocks, which use multiple self-attention layers that are biased by the pair representation to encourage consistency between the two. This process is repeated for 4 cycles, where the MSA and pair representations determined from the MSA Transformer layers of the previous cycle as well as the sequence embedding from the previous Sequence Transformer layers are added to those produced by the current iteration, allowing the network to gradually refine its predictions. Finally, the distance and orientation restraints are predicted from a linear projection of the final pair representation, while the backbone torsion angles are predicted from a linear projection of the sequence representation. A more detailed description of each component of the network is described below.

Input embedding. As shown in Fig. 3.1.A, the input features are used to generate two major representations: the MSA representation and the pairwise representation. The MSA embedding captures the evolutionary information contained in the MSA, while the pairwise representation captures the pairwise relationships between each nucleic acid in the target sequence. To initialize the MSA representation, up to 128 sequences from the MSA are randomly sampled and encoded using one-hot-encoding. Then a linear layer with an output channel size of 32 is used to embed the one-hot-encoded MSA along with the relative positional encodings of each MSA column. The nhmmer HMM is also embedded using a linear layer with an output channel dimension of 32 and concatenated to the MSA embedding to produce the initial MSA representation. The pairwise representation is initialized from the paired sequence encodings as well as the predicted secondary structure using linear layers with an output dimension of 32. Here, the predicted secondary structure is not in dot-bracket format, but rather the LxL reliability scores output by PETfold, which allows for a convenient projection of the secondary structure information as well as inherently capturing the uncertainty in the predictions. A triangular self-attention procedure (123) is applied to further refine the pair representation derived from the predicted secondary structure information. Triangular self-attention represents the pair embedding as a directed graph and performs multiple rounds of self-attention-based transformations by first updating the outgoing edges of the pair representation graph followed by the incoming edges. Then self-attention is performed around the starting nodes and around the ending nodes. Lastly, a transition block composed of two linear layers is used to project the pair embedding to an output dimension of 32*2 and back down to the original size of 32.

MSA Transformer network. The MSA Transformer network takes as input the MSA and pair embeddings, where the MSA representation contains information from homologous sequences in each row and the positional relationships in each column and the pair representation contains the pairwise distance relationships. The MSA embedding is first processed using multi-head, row-wise self-attention, which extracts positional information encoded by the different homologous sequences contained in the MSA. During the row-wise self-attention procedure, the rows of the MSA are mapped to a set of queries (q), keys (k), and values (v) using linear layers with an input dimension of 32 and an output dimension of 8x16, where 8 is the number of heads and 16 is the size of the hidden dimension.

The attention maps can be derived from the set of queries, keys, and values following the standard formulation: $att = softmax(q^T k/\sqrt{c})$, where c is the size of the hidden dimension. Bias from a linear projection of the pair representation is added to the resulting attention maps and the updated MSA row embeddings are determined by applying the attention maps to the values (v) along with a gate determined from a linear projection of the rows of the MSA representation followed by a sigmoid activation. After the row-wise self-attention layer, a similar procedure is repeated for the MSA columns using multi-head, column-wise self-attention. During this process, the columns of the MSA are mapped to queries, keys, and values using linear layers with an input dimension of 32 and an output dimension of 8x16, where 8 is the number of heads and 16 is the size of the hidden dimension. Next, the MSA columns are updated using the queries and keys to calculate the attention maps and applying these to the obtained values. Similar to the MSA row-wise self-attention, a gate is applied to the updated columns using a linear projection of the MSA columns followed by a sigmoid activation. The updated MSA rows and columns are further processed using an MSA transition block that passes the embedding through two linear layers, where the first layer projects the MSA embedding with a hidden dimension of 32 to a hidden dimension of 32x16 and the second layer projects the MSA embedding back to its original size of 32. Finally, the pairwise representation is updated by taking the outer product mean of the processed MSA representation and adding it to the pair representation. The pair representation is then processed using the same triangular self-attention scheme introduced above.

Overall, this process is repeated 48 times to gradually refine the MSA and pair embeddings, where the final output is the updated MSA and pair representations. If it is not the first cycle through the network, the MSA and pair representations from the previous pass of the network are then added to the representations from the current cycle.

Sequence Transformer network. Following the MSA Transformer layers, the position in the MSA embedding that corresponds to the original sequence is extracted and a linear layer is used to project its dimension from 32 to 64. Next, the sequence embedding is processed by two self-attention blocks. The first maps the input sequence to a set of queries, keys, and values from which the attention maps are derived. Bias from a linear projection of the pair representation is then added to the attention maps and the attention maps are applied to the values to update the sequence representation. A gate is also applied to the updated sequence representation by a linear projection of the sequence embedding followed by a sigmoid activation. The second attention block is similar to the first with the exception that it does not include bias from the pair representation. Finally, the updated sequence embedding is passed through 3 linear layers with an input and output channel dimension size of 64 to produce the final sequence representation. If it is not the first cycle through the network, the sequence representation from the previous pass through the network is added to the final sequence embedding from the current iteration.

Geometric restraint prediction. The predicted geometric restraints include the pairwise distance maps between the N1/N9 atoms, C4' atoms, and backbone P atoms, as well as the inter-residue and backbone torsion angles $(\omega, \lambda, \eta, \theta)$ specified in Fig. 3.2. The distances are divided into 40 bins, where the first and last bins indicate predicted distances <2 Å or >40 Å, respectively, while the middle 38 bins correspond to distances in the range of [2Å, 40Å] with an even bin width of 1Å. Similarly, the inter-residue torsion angles (Ω, λ) are divided into 25 bins with a width of 15°, where the first 24 bins correspond to the probability that the orientation angles fall in the range [-180°, 180°] and the last bin captures the probability that there is no interaction between a given pair of nucleotides. Non-interacting nucleotides are defined as those with N1/N9-N1/N9 distances >40 Å. The distance and orientation restraints are predicted from the final pairwise representation using linear layers with an input dimension of 32 and an output dimension of 40 for the distance restraints and 25 for the inter-residue orientation restraints, where a log softmax activation is applied to each output restraint. Lastly, the backbone pseudo-torsion angles (η, θ) are predicted from a linear projection of the sequence embedding with an input channel dimension of 64 and an output channel dimension of 24. Thus, the predicted pseudo-torsion angles are divided into 24 bins from [-180°, 180°] with a width of 15°, where a log softmax activation function is applied to the final predictions.

3.3.3 Training Data and Procedure

DeepFoldRNA was trained on 2,986 RNA chains collected from the PDB which were non-redundant (with a sequence identity <80%) to the 122 test RNAs used in this study. The labeled features from the PDB structures include the native C4', N1/N9 and P distance maps, the inter-residue Ω and λ orientations, and the backbone η and θ pseudo-torsion angles. The training features from the experimental structures were discretized into binned values with the same sizes as the predicted features. The output of the network is the probability that each feature falls within one of the given bins, thus the network was trained using the softmax cross-entropy loss between the predicted and native distributions. In addition, a BERT-style loss (204) was incorporated by randomly masking positions in the MSA and predicting the masked positions from a linear projection of the final MSA representation. The softmax cross-entropy loss was then calculated between the predicted MSA and the unmasked MSA.

Since the number of solved RNA structures is low, we also collected a non-redundant distillation set of 16,842 RNAs from the bp-RNA-1m database (205) that were predicted to have regular secondary structures. Since the RNA in the distillation set do not have solved tertiary structures, we generated predicted labels for each RNA based on the network that was trained on the PDB sample. We then trained the network by sampling from the PDB dataset at a probability of 25% and from the distillation set at a probability of 75%. The loss function for the distillation set was identical to that used for the PDB set, where the softmax cross-entropy loss was calculated between the predicted features and the labels.

The network model was trained using Adam optimization with a learning rate of 0.001 for 159,000 steps, where the distillation set was incorporated after step 99,000. The entire model was trained using a single Nividia V100 SMX2 GPU on the SDSC Expanse cluster (206).

3.3.4 DeepFoldRNA Energy Function

DeepFoldRNA uses L-BFGS simulations to quickly fold RNA based on optimization of the following energy function:

$$\mathrm{E_{DeepFoldRNA}} = \mathrm{E_{C4'dist}} + \mathrm{E_{Ndist}} + \mathrm{E_{Pdist}} + \mathrm{E_{\Omega}} + \mathrm{E_{\lambda}} + \mathrm{E_{bb\eta}} + \mathrm{E_{bb\theta}} \tag{3.1}$$

where $\mathrm{E_{C4'dist}}$, $\mathrm{E_{Ndist}}$, $\mathrm{E_{Pdist}}$, $\mathrm{E_{\Omega}}$, $\mathrm{E_{\varphi}}$, $\mathrm{E_{bb\eta}}$, and $\mathrm{E_{bb\theta}}$ are energy terms derived from the predicted C4'–C4' distances, N1/N9-N1/N9 distances, P-P distances, Ω orientations, λ orientations, backbone η torsions, and backbone θ torsions, respectively. The details of each energy term are further explained in Appendix F Text F.3. As L-BFGS optimization requires a continuously differentiable energy function, the energy terms are fit using cubic spline interpolation.

Overall, the DeepFoldRNA force field consists of 7 weighting parameters, which were determined on 184 RNAs from the training set with lengths between 60-480 nucleotides and pairwise sequence identities <50%. The weights were tuned iteratively, where the weight for each energy term was adjusted one at a time in the range [0, 10] using an increment of 0.25. The weighting parameter for each term that produced models with the highest average TM-score for the 184 RNAs were selected. Once the initial weight for each energy term was determined, this process was repeated 4 more times, varying each weight one-at-a-time using an increment of 0.1.

3.3.5 L-BFGS Folding Simulations

In DeepFoldRNA, an RNA structure is represented by its backbone P and C4' atoms as well as the N1/N9 atoms and two carbon atoms from the base (C2/C4 for pyrimidines and C2/C6 for purines) (Fig. 3.2.A). During the simulations, the bond lengths and bond angles are fixed at their ideal values, and the optimization is directly carried out on the backbone η and θ pseudo-torsion angles guided by the gradient of the energy function with respect to η and θ. L-BFGS optimization is used to find the backbone η/θ angles for each residue that minimize the energy function described in Eq. (3.1).

Here, L-BFGS is a gradient descent-based optimization method built on a limited memory variant of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, which attempts to identify the minimum of $E_{\text{DeepFoldRNA}}(\eta,\theta)$, where η/θ are vectors of length L that represent the backbone pseudo-torsion angles at each position of the simulated structure. At each L-BFGS step k, the search direction d_k is calculated by

$$d_k = -H_k^{-1} \cdot \nabla E_{\text{DeepFoldRNA}}(x) \tag{3.2}$$

where H_k^{-1} is an estimate for the inverse Hessian matrix and $\nabla E_{\text{DeepFoldRNA}}(x)$ is the gradient of the DeepFoldRNA energy function with respect to the backbone pseudo-torsion angles, that is $x = (\eta,\theta)$. H_k^{-1} at step $k = 0$ is set to the identity matrix, I, and the value of H_{k+1}^{-1} is obtained following the BFGS formulation:

$$H_{k+1}^{-1} = V_k^T H_k^{-1} V_k + \rho_k s_k s_k^T,\ \ \rho_k = {}^{1}\!/_{y_k^T s_k},\ \ V_k = I - \rho_k y_k s_k^T \tag{3.3}$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla E_{\text{DeepFoldRNA}}(x_{k+1}) - \nabla E_{\text{DeepFoldRNA}}(x_k)$. Accordingly, the value of H_{k+1}^{-1} can be computed recursively by storing the previously calculated values of s_k and y_k. However, to preserve memory, L-BFGS only stores the last m values of s_k and y_k. Thus, H_{k+1} can be calculated as follows:

$$H_{k+1}^{-1} = \left(\prod_{i=k}^{k-\widehat{m}+1} V_i^T\right) H_0^{-1} \left(\prod_{i=k-\widehat{m}+1}^{k} V_i\right) + \sum_{j=k}^{k-\widehat{m}+1} \left(\prod_{i=k+1}^{j+1} V_i\right) \rho_k s_k s_k^T \left(\prod_{i=j+1}^{k} V_i\right) \tag{3.4}$$

where $\widehat{m} = min(k, m-1)$ and m is set to 256 in DeepFoldRNA. Once the search direction d_k is calculated, the η/θ angles are updated according to:

$$\eta_{k+1} = \eta_k + \alpha_k d_k,\ \ \theta_{k+1} = \theta_k + \alpha_k d_k \tag{3.5}$$

The value of α_k is determined using the Armijo line search technique (185) and dictates the amount to move along the given search direction. In DeepFoldRNA, a maximum of 10 L-BFGS iterations are performed with 2000 steps each, or until the simulations converge. We also use 3 rounds of noisy restarts, where the optimal backbone pseudo-torsion angles from the previous simulation are perturbed by a random value in the range [-10°, 10°] to avoid becoming trapped in local minima. The final model is the lowest energy decoy produced during the folding simulations.

3.4 Author Contributions

The findings from this study are available as a bioRxiv preprint (127) with myself as first author (R.P.), co-author Dr. Gilbert S. Omenn (G.S.O.), and corresponding author Dr. Yang Zhang (Y.Z.). R.P. developed DeepFoldRNA, performed the experiments, analyzed the data, developed the standalone package, and drafted the figures and text; R.P., G.S.O., and Y.Z. finalized the manuscript.

CHAPTER 4

EvoDesign: Online Resource for Designing Protein-Protein Binding Interactions Using Evolutionary Interface Profiles in Conjunction with an Optimized Physical Energy Function

We will now transition from the problem of structure prediction to the second major topic of this thesis, protein design. In this chapter, we will specifically focus on protein sequence design and the development of an online webserver for this purpose. Despite the impressive role of natural proteins, only a tiny portion of the total possible amino acid sequences appear in nature. Computational protein design can be used to more thoroughly explore the sequence space in order to design artificial proteins with increased stability and/or enhanced functionality compared to their natural counterparts. Since many protein functions are mediated by protein-protein interactions (PPIs) (207, 208), an effective strategy to enhance the function of proteins is to redesign their interfaces to increase or alter the binding affinity and binding mode of PPIs (209). This approach has been successfully applied to the redesign of various protein systems (140, 210-213), and holds tremendous potential for the development of novel therapeutics, enzymes, and other useful proteins.

Most current protein design methods utilize physical energy functions to search for low free energy states in sequence space. This approach, however, may be limited by the inability of physical energy functions to accurately recapitulate inter-atomic interactions or recognize correct folds, which has also been manifested in various protein folding and structure prediction studies (80, 133). To partially address the inaccuracies of computational protein design using physics-based energy functions, we previously developed an evolution-based method, EvoDesign (214). EvoDesign utilizes evolutionary profiles collected from analogous protein folds to help guide the sequence search simulations. Large-scale design and folding experiments demonstrated that the combination of evolutionary profiles with physical energy terms, where the latter was included mainly to accommodate the local atomic-level packing interactions, was more effective than purely physics-based methods in terms of designing proteins that adopted a desired target fold (215). Despite the success, the former version of EvoDesign focused solely on the design of monomeric proteins, and could not be used to design PPIs, which considerably limited its usefulness in terms of functional protein design.

In this work, we extended the use of evolutionary-profile guided design to the design of PPIs. For this purpose, a new strategy was developed to extract PPI profiles from structurally analogous protein interfaces, which are then used to guide the interface design search (216). Furthermore, the former EvoDesign pipeline utilized an external program, FoldX (217), to calculate the physical energy of a protein. Although it worked reasonably well, the procedure of calling an external program was prohibitively time-consuming. We developed a new physical energy function, EvoEF (EvoDesign Energy Function), which showed an improved ability to recognize inter-molecular binding interactions, while significantly speeding up the design process. Overall, the new EvoDesign server contains two design protocols: monomer fold design and dimer interface design, each with its own online interface.

It should be noted that the focus of the new dimer interface design protocol is on the redesign of one specific chain in the complex structure, termed the scaffold, so as to increase its stability and binding affinity towards the other chain in the complex, termed the binding partner. The sequence of the binding partner is unchanged during the simulations, although its side-chain conformations are allowed to rotate in order to accommodate the designed interface residues. This interface design protocol can be used for various applications that allow for a variable scaffold protein but call for a fixed binding partner. One such application is the design of protein therapeutics, where the therapeutic can be redesigned to increase its affinity for a fixed target in the body. The EvoDesign pipeline is fully automated and freely available at https://zhanggroup.org/EvoDesign/. In addition to the online server, the source code for the newly developed physical energy function, EvoEF, can be downloaded at https://zhanggroup.org/EvoEF/.

4.1 Methods and Results

4.1.1 Overview of the EvoDesign Protocol

In order to incorporate functional protein design into EvoDesign, the evolution-based design method was extended to the design of PPIs, where an overview of the new EvoDesign pipeline is depicted in Fig. 4.1.

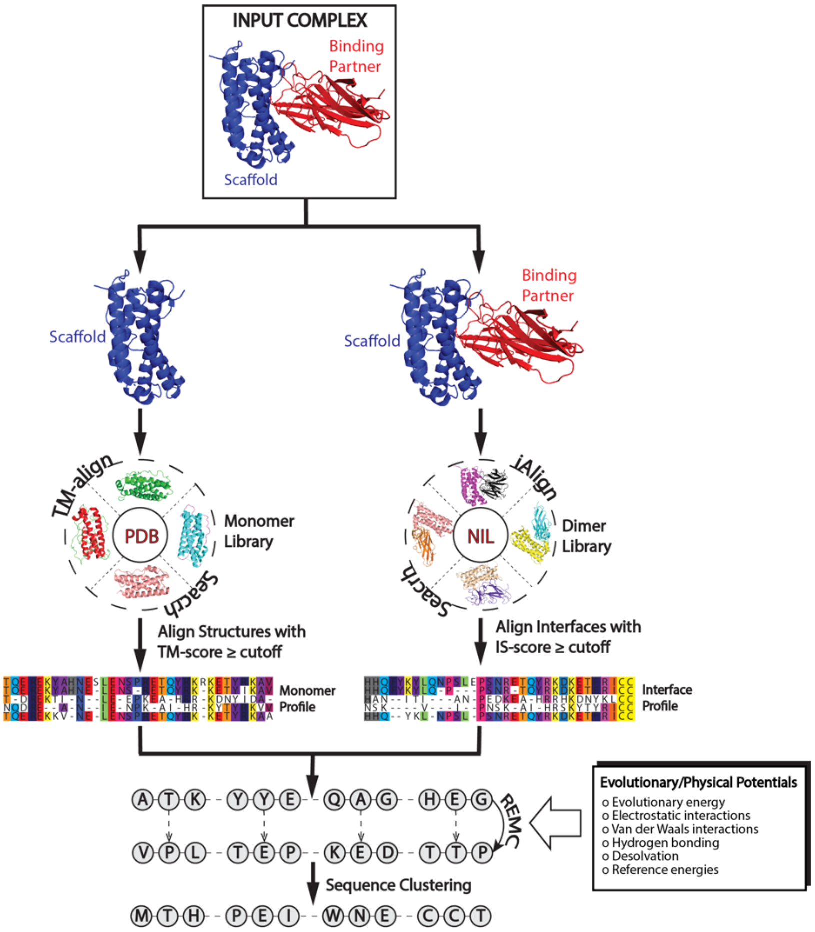


Figure 4.1 Flowchart of the EvoDesign pipeline for protein-protein interaction design. Similar monomer and interface structures are identified from monomer and dimer libraries, respectively, which are used to create the evolutionary profiles. These profiles guide the redesign simulations of the scaffold protein.

Starting from a two-chain complex structure of interest, its interface is structurally aligned to interfaces in the non-redundant interface library (NIL) (216) using iAlign (218). A profile is then constructed from the interface multiple sequence alignment (iMSA), based on the structures that have a high similarity score (IS-score (218)) to the query complex interface. Finally, the evolution-based binding affinity change for each mutation at the interface is determined by the logarithm of the relative probability of each mutant amino acid compared to the wild-type amino acid in the interface profile (216, 219). This evolutionary energy term is combined with the physical energy score calculated by EvoEF to determine the total binding energy. Complementing the interface profile, a monomer structural profile is constructed from the multiple sequence alignment of monomer proteins that have a similar fold to the scaffold chain as identified by TM-align (220) from the PDB library. Overall, the information from both the monomer and interface profiles, as well as the physical energy function, are used as the composite energy function to guide the replica-exchange Monte Carlo (REMC) simulation in order to search for low free energy sequences.

Following the REMC simulation, the generated sequence decoys are clustered by SPICKER (221) based on the distance matrix defined by their BLOSUM62 sequence similarity. The final designs are selected from the lowest free energy sequences in the largest clusters. Here, it is important to note that EvoDesign provides an option for users to specify which chain in the complex is the scaffold and which chain is its binding partner. As stated previously, EvoDesign only focuses on the redesign of the scaffold, leaving the sequence of its binding partner unchanged, although the side-chain rotamer conformations of both chains are repacked during the design simulations.

4.1.2 Evolutionary Profile-Based Potentials

The evolutionary energy is composed of two terms: $E_{evoMonomer}$ and $E_{evoInterface}$. The first term, $E_{evoMonomer}$ is used to capture the information from the multiple sequence alignment (MSA) generated by TM-align based on the scaffold structure. The derivation of $E_{evoMonomer}$ was discussed previously (215). For the webserver description, we will focus on the new evolutionary interface potential. However, a full explanation of $E_{evoMonomer}$ is provided in Text G.1 in Appendix G.

The second term, $E_{evoInterface}$, captures the information from the iMSA collected by the iAlign search:

$$\begin{aligned} E_{evoInterface}(S_{Des}, S_{Scaff}) &= -\sum_{i=1}^{L} \ln \frac{P(aa_{Des,i}, i)}{P(aa_{Scaff,i}, i)} \\ &= -\sum_{i=1}^{L} \ln \frac{N_{obs}(aa_{Des,i}, i) + N_{pseudo}(aa_{Des,i}, i)}{N_{obs}(aa_{Scaff,i}, i) + N_{pseudo}(aa_{Scaff,i}, i)} \end{aligned} \tag{4.1}$$

where $P(aa_{Des,i}, i)$ and $P(aa_{Scaff,i}, i)$ are the probabilities that the designed and scaffold amino acids, respectively, appear at position i in the interface. These terms are composed of the number of times either the designed, $N_{obs}(aa_{Des,i}, i)$, or the native scaffold, $N_{obs}(aa_{Scaff,i}, i)$, amino acids appear at the i^{th} position in the iMSA and the corresponding, position-specific pseudocount, N_{pseudo}. The pseudocount is used to help compensate for the small size of the interface library and takes into consideration gaps in the iMSA as well as amino acids that are related to the native/mutant residues in the interface alignment. A full description of the pseudocount is contained in Text G.2.

4.1.3 EvoEF Energy Terms

The energy function of EvoEF is designed to describe the atomic interactions in proteins and contains five terms:

$$E_{EvoEF} = \sum_{i,j} [E_{vdw}(i,j) + E_{elec}(i,j) + E_{HB}(i,j) + E_{solv}(i,j)] - E_{ref} \tag{4.2}$$

The first term, $E_{vdw}(i,j)$, is the van der Waals energy, which is modified from the Lennard-Jones (LJ) 12-6 potential (222, 223):

$$E_{vdw}(i,j) = w_{vdw} \begin{cases} min\left\{5.0\varepsilon_{ij}, \varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{d_{ij}}\right)^{12} - 2\left(\frac{\sigma_{ij}}{d_{ij}}\right)^{6}\right]\right\}, & if\ d_{ij} < 0.8909\sigma_{ij} \\ \varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{d_{ij}}\right)^{12} - 2\left(\frac{\sigma_{ij}}{d_{ij}}\right)^{6}\right], & if\ 0.8909\sigma_{ij} \le d_{ij} < 5.0 \\ A * d_{ij}^3 + B * d_{ij}^2 + C * d_{ij} + D, \quad if\ 5.0 \le d_{ij} \le 6.0, & if\ 5.0 \le d_{ij} < 6.0 \\ 0, & if\ d_{ij} \ge 6.0 \end{cases} \tag{4.3}$$

$$\begin{cases} A = -0.4\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{12} - 1.6\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{6} \\ B = 7.8\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{12} + 25.2\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{6} \\ C = -50.4\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{12} + 129.6\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{6} \\ D = 108\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{12} + 216\varepsilon_{ij}\left(\frac{\sigma_{ij}}{5.0}\right)^{6} \end{cases} \tag{4. 4}$$

where d_{ij} is the distance between the two atoms i and j, $\sigma_{ij} = \sigma_i + \sigma_j$ is the sum of their van der Waals atomic radii and ε_{ij} is the combined well depth parameter for atoms i and j, which is taken from the CHARMM19 force field (224). Here, the attractive and repulsive components of the van der Waals potential are split at $\mathrm{d}_{ij} = 0.8909\sigma_{ij}$. To increase the computational efficiency of EvoEF, we set a maximum distance cutoff of 6.0 Å and use a cubic function to continuously transition the LJ energy from its value at 5.0 Å to zero at the cutoff distance. For the repulsive component of the LJ potential, the maximum energy cutoff is set to $5.0\varepsilon_{ij}$. This helps alleviate possible clashes, while not overly penalizing them due to the discrete rotameric conformations used in protein design. An example of the overall shape of the van der Waals energy between an amide N and a carbonyl C is shown in Fig. 4.2.

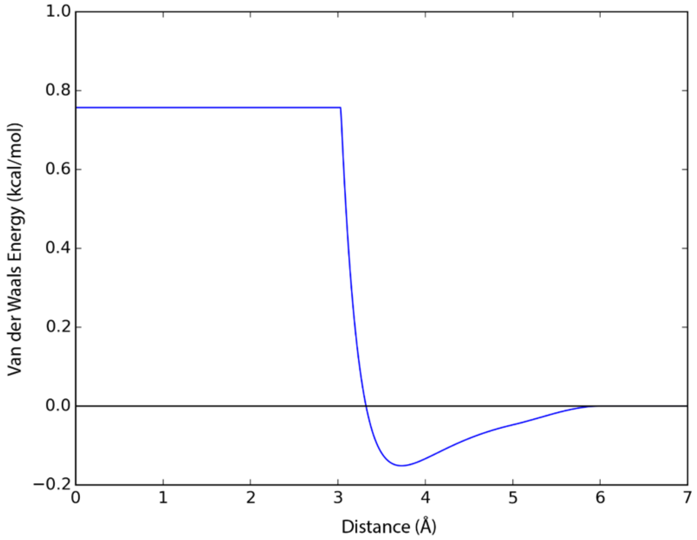


Figure 4.2 Shape of the Van der Waals Energy Between an Amide N and a Carbonyl C. The van der Waals radii for the amide N and carbonyl C are 1.7632 and 1.9649 Å, respectively, while their corresponding well-depths are 0.1617 and 0.1418 kcal/mol, respectively.

The second term in Eq. (4.2), $E_{elec}(i,j)$, is used to determine the electrostatic interactions between partially charged atoms:

$$E_{elec}(i,j) = w_{elec}\begin{cases} \frac{C_0 q_i q_j}{\varepsilon(0.8\sigma_{ij})}\frac{1}{0.8\sigma_{ij}}, & if\ d_{ij} < 0.8\sigma_{ij} \\ \frac{C_0 q_i q_j}{\varepsilon(d_{ij})}\frac{1}{d_{ij}}, & if\ 0.8\sigma_{ij} < d_{ij} < 6.0 \\ 0, & if\ d_{ij} \geq 6.0 \end{cases} \tag{4.5}$$

where q_i and q_j are the partial atomic charges, which are calculated using the PARSE method (225). Furthermore, $C_0 = 332$ Å kcal mol$^{-1}e^{-2}$, where e is the elementary charge, and $\varepsilon(d_{ij})$ is the distance-dependent dielectric constant. $\varepsilon(d_{ij})$ takes the form $\varepsilon(d_{ij}) = 40d_{ij}$. When computing the electrostatics term and dielectric constant, if the distance between two atoms, d_{ij}, is less than $0.8\,\sigma_{ij}$, d_{ij} is set to $0.8\,\sigma_{ij}$. This restricts the electrostatics energy to a reasonable, finite value. Again, for the sake of computational efficiency, a maximum distance cutoff is set to 6.0 Å, beyond which the value of the electrostatics term is zero.

The third term in Eq. (4.2), $E_{HB}(i,j)$, is used to calculate the hydrogen-bonding interactions. $E_{HB}(i,j)$ is a linear combination of three energy terms that depend on the hydrogen-acceptor distance ($\mathrm{d}_{\mathrm{ij}}^{\mathrm{HA}}$), the angle between the donor atom, hydrogen and acceptor ($\theta_{\mathrm{ij}}^{DHA}$), and the angle between the hydrogen, acceptor and base atom ($\varphi_{\mathrm{ij}}^{HAB}$):

$$E_{HB}(i,j) = w_{d_{HA}} E\left(d_{ij}^{HA}\right) + w_{\theta_{DHA}} E\left(\theta_{ij}^{DHA}\right) + w_{\varphi_{HAB}} E\left(\varphi_{ij}^{HAB}\right) \tag{4.6}$$

where

$$\begin{cases} E\left(d_{ij}^{HA}\right) = \begin{cases} -\cos\left[\frac{\pi}{2}\left(d_{ij}^{HA} - 1.9\right)/(1.9 - d_{min})\right] & d_{min} \le d_{HA} \le 1.9 \\ -0.5\cos\left[\pi\left(d_{ij}^{HA} - 1.9\right)/(d_{max} - 1.9)\right] - 0.5 & 1.9\text{ Å} < d_{HA} \le d_{max} \\ 0 & otherwise \end{cases} \\ E\left(\theta_{ij}^{DHA}\right) = -\cos^4\left(\theta_{ij}^{DHA}\right) \\ E\left(\varphi_{ij}^{HAB}\right) = \begin{cases} -\cos^4\left(\varphi_{ij}^{HAB} - 150\right), & for\ HBbb\ and\ for\ sp^2\ in\ HBsb\ or\ HBss \\ -\cos^4\left(\varphi_{ij}^{HAB} - 135\right), & for\ sp^3\ in\ HBsb\ or\ HBss \end{cases} \end{cases} \tag{4.7}$$

The optimal distance between the hydrogen and its acceptor is set to 1.9 Å, which was taken from Kortemme *et al.* (226). Additionally, $d_{min} = 1.4$ Å and $d_{max} = 3.0$ Å are the lower and upper bounds on the distance between the hydrogen-acceptor pair. The optimal φ_{ij}^{HAB} value is set to either 150° or 135°, depending on the acceptor hybridization (sp^2 or sp^3) and the locations of the donor and acceptor atoms (HBbb: backbone-backbone; HBsb: sidechain-backbone; HBss: sidechain-sidechain).

The fourth term in Eq. (4.2), $E_{solv}(i,j)$, describes the desolvation energy following the model introduced by Lazaridis and Karplus (227):

$$E_{solv}(i,j) = -V_j \frac{\Delta G_i^{free}}{2\pi^{\frac{3}{2}}\lambda_i d_{ij}^2} exp\left[-\left(\frac{d_{ij}-\sigma_i}{\lambda_i}\right)^2\right] - V_i \frac{\Delta G_j^{free}}{2\pi^{\frac{3}{2}}\lambda_j d_{ij}^2} exp\left[-\left(\frac{d_{ij}-\sigma_j}{\lambda_j}\right)^2\right] \qquad (4.8)$$

where $V_{i,j}$, $\Delta G_{i,j}^{free}$, and $\lambda_{i,j}$ are the atom volumes, reference solvation energies, and correlation lengths, respectively. These values were taken from the Lazardis and Karplus paper (227). The desolvation energy for both polar and nonpolar atoms is calculated using this method; however, the contribution from polar atoms is weighted differently from non-polar atoms. Specifically, $E_{solvPolar}(i,j) = w_{solvPolar}E_{solv}(i,j)$ and $E_{solvNonpolar}(i,j) = w_{solvNonpolar}E_{solv}(i,j)$.

The last term in Eq. (4.2), E_{ref}, is the reference energy of a protein sequence and is used to approximate the energy of the unfolded state ensemble:

$$E_{ref} = \sum_{i=1}^{L} E_r(aa_i) \qquad (4.9)$$

where L is the length of the protein sequence, $E_r(aa_i)$ is an amino acid specific parameter to be optimized. The reference energy is used to choose sequences that have a large energy gap between the folded and unfolded states.

4.1.4 EvoEF Parameter Optimization and Benchmark Tests

EvoEF contains a total of 36 weights and 20 reference energies. These parameters were decided by optimizing the energy function's ability to predict protein stability and binding affinity change upon mutation. Since EvoEF's energy calculation is split into three parts: the non-bonded atomic interactions within a residue ($E_{intraResidue}$), those between different residues within the same chain ($E_{interResidueSameChain}$), and those between different residues from different chains ($E_{interResidueDiffChain}$) (see Text G.3), the parameterization of EvoFF was performed in two steps. First, the reference energies and weighting factors for $E_{intraResidue}$ and $E_{interResidueSameChain}$

were optimized by minimizing the difference between experimental and predicted values for mutation-induced protein monomer stability change ($\Delta\Delta G_{stability}^{WT\rightarrow mut}$). The experimental data consisted of 3,989 non-redundant mutation samples from 210 monomeric proteins taken from the FoldX and STRUM datasets (44, 228). Second, the 14 weights for $E_{interResidueDiffChain}$ were determined using the mutation-induced protein-protein binding affinity change data ($\Delta\Delta G_{binding}^{WT\rightarrow mut}$), which contained 2,204 non-redundant mutant samples from 177 dimeric complexes collected from the latest version of the SKEMPI database (229). Each dataset was randomly split in half into training and test sets. A detailed description of the data construction and EvoEF optimization procedure is provided in Text G.3.

The performance of EvoEF was evaluated using the above test datasets by calculating the Pearson correlation coefficients (PCCs) and root mean square errors (RMSEs) between the experimental and predicted $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$. The results showed that the PCC between $\Delta\Delta G_{stability,pred}^{WT\rightarrow mut}$ and $\Delta\Delta G_{stability,exp}^{WT\rightarrow mut}$ for EvoEF was 0.472 with an RMSE of 1.751 kcal/mol (Fig. 4.3.A). As a comparison, FoldX obtained a PCC of 0.465 with an RMSE of 2.010 kcal/mol for the same dataset (Fig. 4.3.B). Furthermore, the PCC between $\Delta\Delta G_{binding,pred}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding,exp}^{WT\rightarrow mut}$ for EvoEF was 0.514 with an RMSE of 2.109 kcal/mol (Fig. 4.3.C), while the PCC for FoldX was 0.490 with an RMSE of 2.248 kcal/mol (Fig. 4.3.D). The data show that EvoEF slightly outperforms FoldX for both $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ prediction.

We also tested EvoEF's ability to recognize the native structure from non-native decoys using the 3DRobot Decoy Set (230), which contains 200 individual decoy sets. Among the 200 decoy sets, EvoEF was able to properly rank the native as the lowest energy in all the sets, while FoldX did so in 198 cases. In the second more stringent test, we calculated the energy gap between the near-native decoys (top 10% of decoys) and the remainder of the decoys. If we define a successful case as that with a Z-score (i.e., the energy gap normalized by the standard deviation) above 1, EvoEF successfully recognized the near-native structural decoys in 198 out of the 200 cases, while FoldX did so for 193 of the cases. Moreover, for the near-native decoy discrimination test, EvoEF had a higher average Z-score of 1.959 compared to 1.844 for FoldX. These data suggest that EvoEF has a relatively better ability to distinguish nativelike monomer structures from other structural decoys (see Text G.4 for a detailed description).

Furthermore, based on our tests on identical computational cores, EvoEF is about three times faster than FoldX at computing stability energy and approximately five times faster at computing protein-protein binding energy, indicating that using EvoEF can significantly increase the speed of our design simulations.

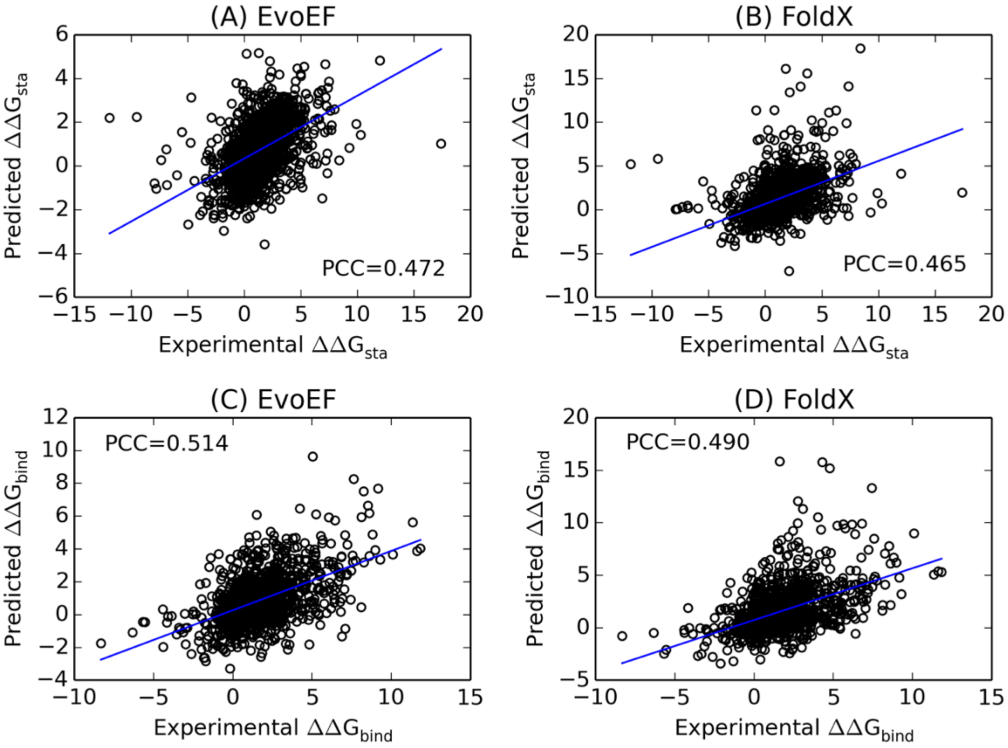


Figure 4.3 Correlation between predicted and experimental values for mutation-induced folding stability and binding affinity changes. (A, B) Folding stability changes upon mutation, $\Delta\Delta G_{stability}^{WT\rightarrow mut}$, in monomer proteins predicted by EvoEF (A) and FoldX (B) versus the experimental data for 1,994 test proteins. (C, D) Binding affinity changes upon mutation in the interface of protein-protein complexes, $\Delta\Delta G_{binding}^{WT\rightarrow mut}$, predicted by EvoEF (C) and FoldX (D) versus the experimental data for 1,102 test proteins.

4.1.5 Replica-Exchange Monte Carlo Simulation for Sequence Space Search

Starting from a random sequence, REMC is used to search the sequence space, where random mutations are made on a set of randomly selected residues at each step, which are accepted or rejected based on the Metropolis criterion (231). The composite energy function used to guide the REMC simulation is as follows:

$$E_{MC} = -E_{evoMonomer} + w_{evoInterface}E_{evoInterface} + w_{EvoEF}E_{EvoEF} \qquad (4.10)$$

where $E_{evoMonomer}$ and $E_{evoInterface}$ are the evolutionary energies from the monomer and interface profiles, while E_{EvoEF} is the physical energy calculated by EvoEF. For interface design, the weight parameters $w_{evoInterface}$ and w_{EvoEF} are set to 3.0 and 2.0, respectively. These weights were selected in order to balance the average contribution from each energy term based on design simulations for a training set composed of 625 monomers and 177 protein-protein complexes (Text G.3).

Within REMC, four parameters need to be carefully considered. First, the highest temperature (T_{max}) should be high enough to enable the simulation to overcome energy barriers, while the lowest temperature (T_{min}) should be low enough to ensure the simulation sufficiently scans the low-energy states. Second, the number of replicas (N_{rep}) should be large enough to ensure sufficient chance for the adjacent replicas to communicate with each other. Third, the number of local movements (N_{sweep}) before the global swaps should be selected to make the local Metropolis search achieve satisfactory equilibrium. After successive rounds of optimization, the final parameters were selected as follows: $T_{max} = 15$ $T_{min} = 0.5$, $N_{rep} = 40$, and $N_{sweep} = 100$.

4.1.6 Server Input

The only input to the EvoDesign server is the monomer (for monomer design) or protein-protein complex (for interface design) structures of interest in PDB format. For monomer design, the input structure may be full-atomic or a Cα trace, while for interface design, it must be full-atomic given the sensitivity of the design to the shape of the binding pocket. In addition, for interface design, the user is able to upload the scaffold structure and its binding partner as a preformed complex structure or as two separate chains. If the two chains are uploaded separately, the user is given an option to dock the two chains together using ZDOCK (232), a leading fast Fourier transform-based protein-protein docking software.

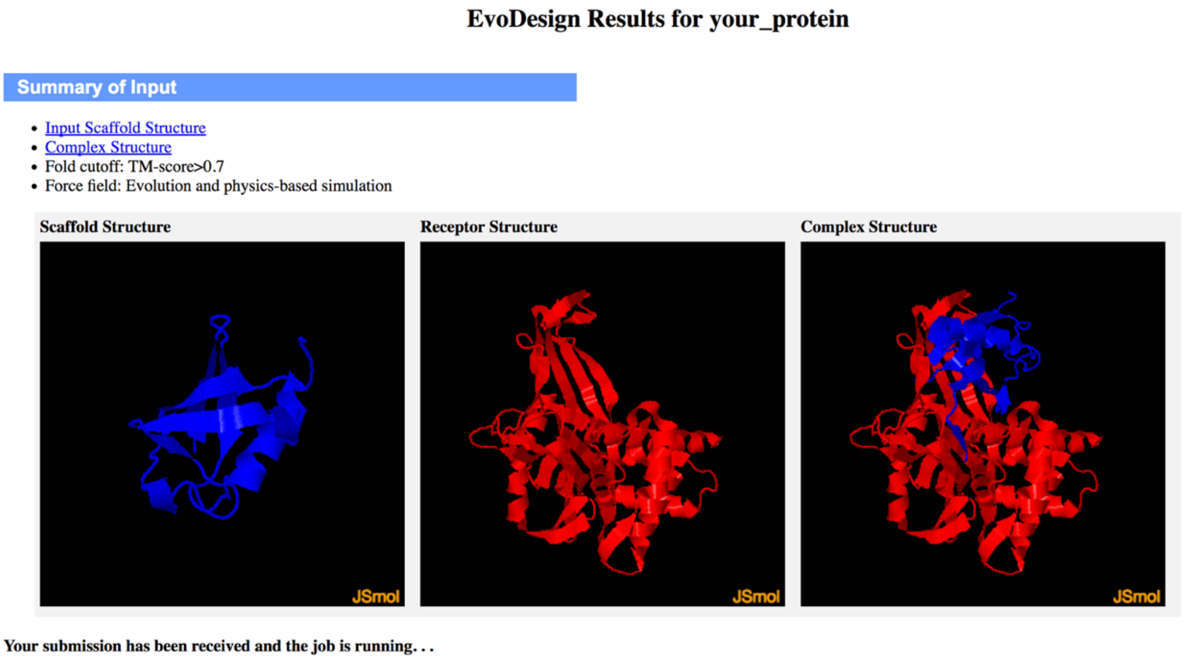
Several advanced options are provided to give users the ability to further tailor the EvoDesign simulation to suit their needs. This is achieved by allowing users (*i*) to select the structural similarity cutoff (TM-score) used during profile construction, (*ii*) to select the type of energy function used during the design simulation (either evolution-based only design or combined physics- and evolution-based design), (*iii*) to exclude residue types at specific locations, (*iv*) to prevent the mutation of residues at specific locations, and (*v*) to model the structures of the final designed sequences using I-TASSER (17).

It should be noted that the default EvoDesign setting for PPI design is to redesign the entire sequence of the scaffold chain. The rationale behind designing non-interface residues is that introducing mutations in the interface may have destabilizing effects on the whole protein or lead to suboptimal packing (233, 234). However, for some large complexes with specific folding architectures such as antibody-antigen complexes, it might be beneficial to focus the design only on the interface regions. Thus, for interface design, users are given an additional option to either redesign the entire scaffold protein or to redesign only its interface residues, which are defined as residues within 5 Å of the opposite chain.

4.1.7 Server Output

Immediately following submission of a design job, an output page with a private URL for the job is created, which users are able to bookmark for future visit. When the EvoDesign simulation finishes, users will be notified by e-mail with a link to the results page. The results in the output page contain:

(*i*) A summary of the input to the server (Fig. 4.4):



EvoDesign Results for your_protein

Summary of Input

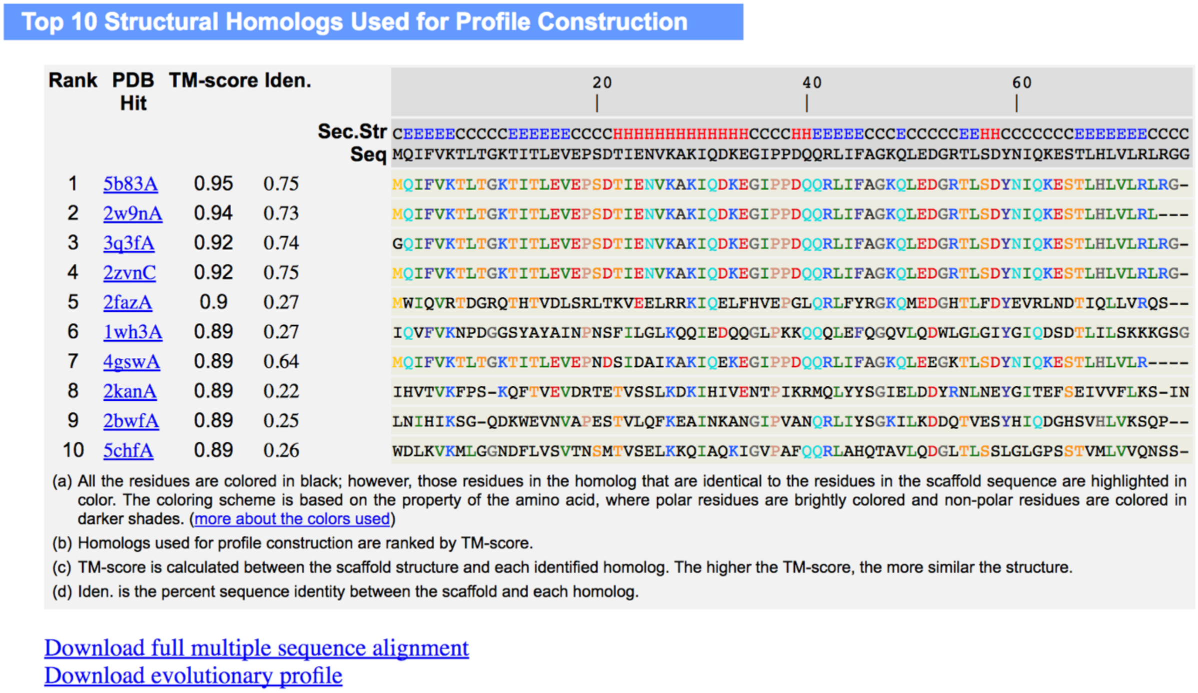
- Input Scaffold Structure
- Complex Structure
- Fold cutoff: TM-score>0.7
- Force field: Evolution and physics-based simulation

Your submission has been received and the job is running. . .

Figure 4.4 Illustration of the input summary section in the EvoDesign output page. The first section of the EvoDesign results page is a summary of the input, which is created immediately after submission of a job. It contains a description of the structural similarity cutoff (TM-score) used

during the evolutionary profile construction along with a description of the force field used by the design simulation. Additionally, a link is provided to download the input scaffold structure and, in the case of interface design, the complex structure. If the user opts to upload the scaffold and receptor structures separately and dock them together, the complex structure will be available to download upon completion of docking. For monomer design, if the input scaffold structure is a Cα trace, a full-atomic model will be generated using REMO (235). The full-atomic model is then uploaded to the server, replacing the initial Cα trace model. The scaffold, receptor and complex structures are visualized using the interactive JSmol applet.

(*ii*) The top structural homologs used for monomer and interface profile construction as well as links to download the full multiple sequence alignment and evolutionary profile (Fig. 4.5):



Top 10 Structural Homologs Used for Profile Construction

Rank	PDB Hit	TM-score	Iden.		20 40 60
				Sec.Str	CEEEEECCCCCEEEEEECCCCHHHHHHHHHHHHCCCCHHEEEEECCCECCCCCEEHHCCCCCCCEEEEEEECCCC
				Seq	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRGG
1	5b83A	0.95	0.75		MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
2	2w9nA	0.94	0.73		MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRL---
3	3q3fA	0.92	0.74		GQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
4	2zvnC	0.92	0.75		MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
5	2fazA	0.9	0.27		MWIQVRTDGRQTHTVDLSRLTKVEELRRKIQELFHVEPGLQRLFYRGKQMEDGHTLFDYEVRLNDTIQLLVRQS--
6	1wh3A	0.89	0.27		IQVFVKNPDGGSYAYAINPNSFILGLKQQIEDQQGLPKKQQQLEFQGQVLQDWLGLGIYGIQDSDTLILSKKKGSG
7	4gswA	0.89	0.64		MQIFVKTLTGKTITLEVEPNDSIDAIKAKIQEKEGIPPDQQRLIFAGKQLEEGKTLSDYNIQKESTLHLVLR----
8	2kanA	0.89	0.22		IHVTVKFPS-KQFTVEVDRTETVSSLKDKIHIVENTPIKRMQLYYSGIELDDYRNLNEYGITEFSEIVVFLKS-IN
9	2bwfA	0.89	0.25		LNIHIKSG-QDKWEVNVAPESTVLQFKEAINKANGIPVANQRLIYSGKILKDDQTVESYHIQDGHSVHLVKSQP--
10	5chfA	0.89	0.26		WDLKVKMLGGNDFLVSVTNSMTVSELKKQIAQKIGVPAFQQRLAHQTAVLQDGLTLSSLGLGPSSTVMLVVQNSS-

(a) All the residues are colored in black; however, those residues in the homolog that are identical to the residues in the scaffold sequence are highlighted in color. The coloring scheme is based on the property of the amino acid, where polar residues are brightly colored and non-polar residues are colored in darker shades. (more about the colors used)

(b) Homologs used for profile construction are ranked by TM-score.

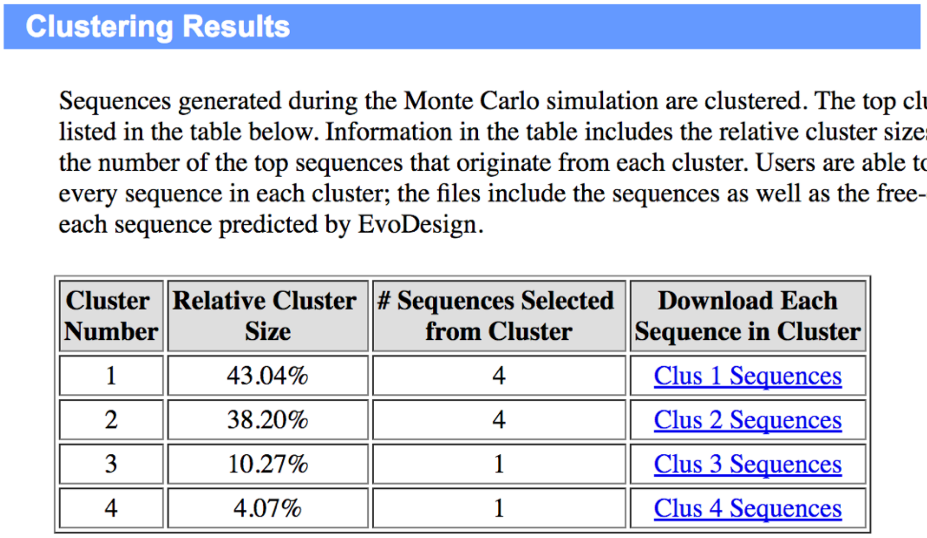
(c) TM-score is calculated between the scaffold structure and each identified homolog. The higher the TM-score, the more similar the structure.

(d) Iden. is the percent sequence identity between the scaffold and each homolog.

Download full multiple sequence alignment
Download evolutionary profile

Figure 4.5 Summary of the top homologs used for profile construction. To generate the evolutionary profiles, structural and interface homologs are identified from the PDB and protein interaction libraries. Although all homologous proteins with TM-scores higher than the specified cutoff are used for monomer profile construction, only the top ten structural homologs, which are sorted by TM-score to the scaffold structure, are displayed in this section. The information displayed for each homolog includes: (*i*) the homolog PDB ID and the link to download the structure, (*ii*) the TM-score and sequence identity to the scaffold, and (*iii*) the alignment between the scaffold and the homolog. Moreover, links are provided to download the full multiple sequence alignment and the evolutionary profile used to guide the design simulation.

(*iii*) The clustering results of sequence decoys generated during the REMC simulation (Fig. 4.6):



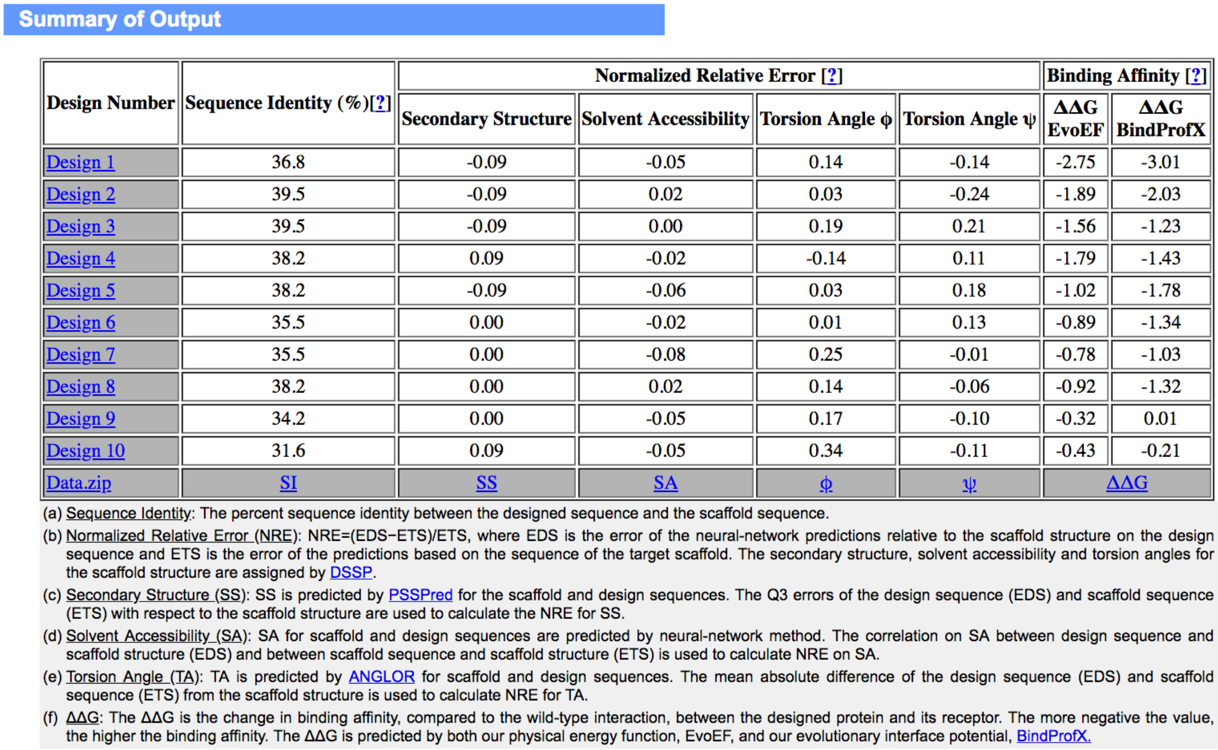
Clustering Results

Sequences generated during the Monte Carlo simulation are clustered. The top clusters are listed in the table below. Information in the table includes the relative cluster sizes as well as the number of the top sequences that originate from each cluster. Users are able to download every sequence in each cluster; the files include the sequences as well as the free-energy of each sequence predicted by EvoDesign.

Cluster Number	Relative Cluster Size	# Sequences Selected from Cluster	Download Each Sequence in Cluster
1	43.04%	4	Clus 1 Sequences
2	38.20%	4	Clus 2 Sequences
3	10.27%	1	Clus 3 Sequences
4	4.07%	1	Clus 4 Sequences

Figure 4.6 Clustering results. During the Monte Carlo simulation, many designed sequences are generated. After the simulation is completed, the generated sequences are clustered using SPICKER (221) based on the distance scaled by their BLOSUM62 sequence similarity. For each target, EvoDesign outputs ten designed sequences, which are selected after clustering. The number of sequences selected from each cluster depends on the cluster size. For example, if 70% of the sequence decoys are contained in the first cluster, 7 sequences would be selected from the first cluster. For each cluster, the sequence in the cluster center is selected first, followed by selection of the non-redundant and lowest-energy sequences. Here, the non-redundant sequence identity cutoff is equal to 70%. The first column of the table lists the cluster number. The relative size of each cluster and the number of sequences selected from each cluster are displayed in columns 2 and 3, respectively. The last column contains links to download text files containing each sequence in the cluster. The files contain the sequences and the EvoDesign calculated energy of each sequence.

(*iv*) A summary of the top ten designed sequences and the local feature assessment parameters (Fig. 4.7):



Summary of Output

Design Number	Sequence Identity (%)[?]	Normalized Relative Error [?]				Binding Affinity [?]	
		Secondary Structure	Solvent Accessibility	Torsion Angle ϕ	Torsion Angle ψ	ΔΔG EvoEF	ΔΔG BindProfX
Design 1	36.8	-0.09	-0.05	0.14	-0.14	-2.75	-3.01
Design 2	39.5	-0.09	0.02	0.03	-0.24	-1.89	-2.03
Design 3	39.5	-0.09	0.00	0.19	0.21	-1.56	-1.23
Design 4	38.2	0.09	-0.02	-0.14	0.11	-1.79	-1.43
Design 5	38.2	-0.09	-0.06	0.03	0.18	-1.02	-1.78
Design 6	35.5	0.00	-0.02	0.01	0.13	-0.89	-1.34
Design 7	35.5	0.00	-0.08	0.25	-0.01	-0.78	-1.03
Design 8	38.2	0.00	0.02	0.14	-0.06	-0.92	-1.32
Design 9	34.2	0.00	-0.05	0.17	-0.10	-0.32	0.01
Design 10	31.6	0.09	-0.05	0.34	-0.11	-0.43	-0.21
Data.zip	SI	SS	SA	ϕ	ψ	ΔΔG	

(a) Sequence Identity: The percent sequence identity between the designed sequence and the scaffold sequence.

(b) Normalized Relative Error (NRE): NRE=(EDS−ETS)/ETS, where EDS is the error of the neural-network predictions relative to the scaffold structure on the design sequence and ETS is the error of the predictions based on the sequence of the target scaffold. The secondary structure, solvent accessibility and torsion angles for the scaffold structure are assigned by DSSP.

(c) Secondary Structure (SS): SS is predicted by PSSPred for the scaffold and design sequences. The Q3 errors of the design sequence (EDS) and scaffold sequence (ETS) with respect to the scaffold structure are used to calculate the NRE for SS.

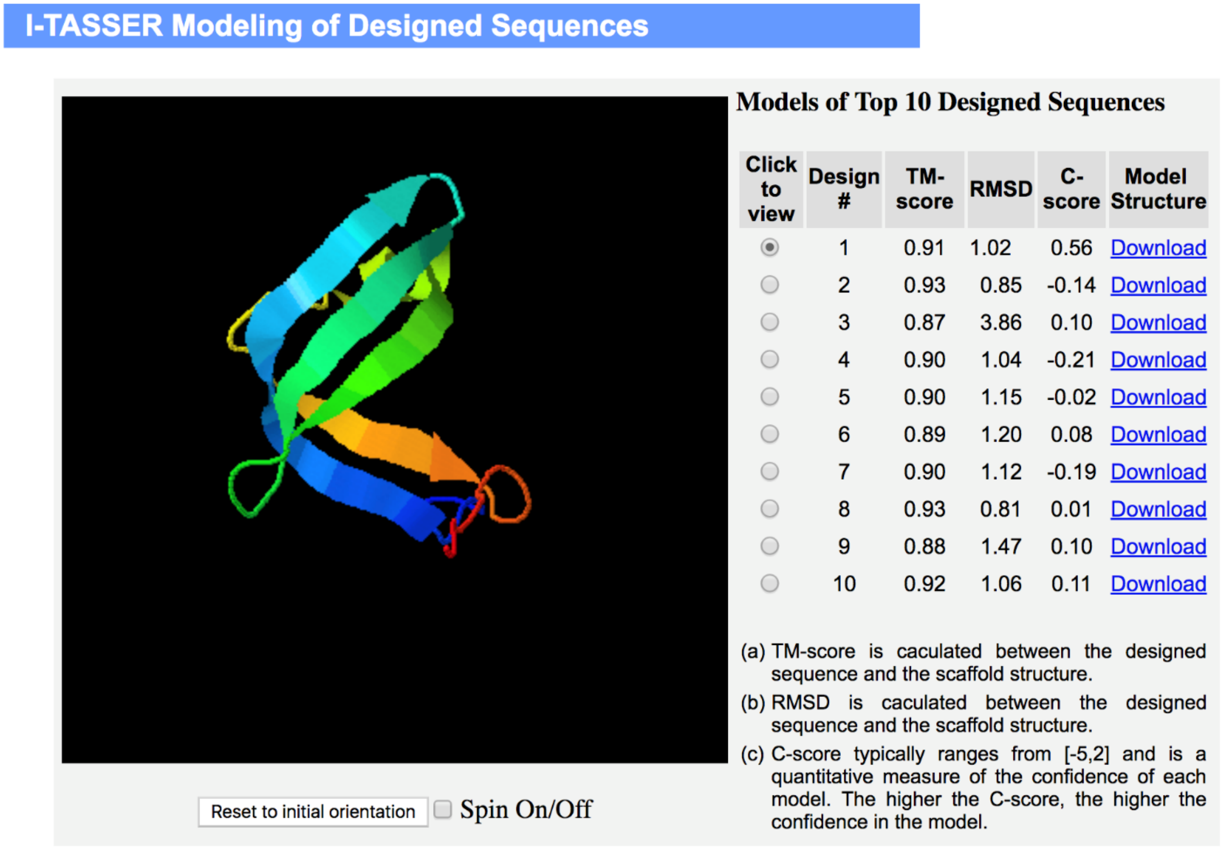
(d) Solvent Accessibility (SA): SA for scaffold and design sequences are predicted by neural-network method. The correlation on SA between design sequence and scaffold structure (EDS) and between scaffold sequence and scaffold structure (ETS) is used to calculate NRE on SA.

(e) Torsion Angle (TA): TA is predicted by ANGLOR for scaffold and design sequences. The mean absolute difference of the design sequence (EDS) and scaffold sequence (ETS) from the scaffold structure is used to calculate NRE for TA.

(f) ΔΔG: The ΔΔG is the change in binding affinity, compared to the wild-type interaction, between the designed protein and its receptor. The more negative the value, the higher the binding affinity. The ΔΔG is predicted by both our physical energy function, EvoEF, and our evolutionary interface potential, BindProfX.

Figure 4.7 Results Table. A summary of the results of the local structural analysis is provided for the top ten designed sequences in tabular form. Here, the secondary structure, solvent accessibility and torsion angles of the designed and scaffold sequences are predicted using three neural-network programs, namely, PSSpred (236), Solve (237), and Anglor (238). The normalized relative errors (NRE) are provided for each of the features (Columns 3-6) to give an approximate assessment of the local structural quality of the designed sequences. The NRE is an error measure for the local structural feature predictors for each designed sequence relative to the scaffold sequence: $NRE = (EDS - ETS)/ETS$. EDS (EDS stands for 'error of designed sequence') is the error of prediction for the designed sequence relative to that assigned by the DSSP program (239) on the scaffold structure, and ETS (ETS stands for 'error of target sequence') is the error of the prediction for the target scaffold sequence. Thus, a small (or negative) NRE value indicates that the designed sequence has a relatively small (or even smaller) prediction error than the scaffold sequence, while a large NRE usually signals a bad design due to the large prediction error relative to the scaffold sequence. Links are provided at the bottom of the table to download the secondary structure, solvent accessibility, and backbone torsional angle prediction data for the designed sequences. The first column of the table contains links so that users can download each of the designed sequences. In addition, the binding energy change of the designed proteins compared to the native scaffold are calculated by EvoEF and BindProfX and are listed in the results table. This should help provide information on how the altered interfaces affect the binding affinities compared to the wild type proteins. All of the information can be downloaded as a compressed file under the link to 'Data.zip'.

(*v*) A detailed overview of the top ten designed sequences including the sequence alignments between the scaffold and designs, and (*vi*) the I-TASSER folding results for the top 10 designs (Fig. 4.8):



I-TASSER Modeling of Designed Sequences

Models of Top 10 Designed Sequences

Click to view	Design #	TM-score	RMSD	C-score	Model Structure
●	1	0.91	1.02	0.56	Download
○	2	0.93	0.85	-0.14	Download
○	3	0.87	3.86	0.10	Download
○	4	0.90	1.04	-0.21	Download
○	5	0.90	1.15	-0.02	Download
○	6	0.89	1.20	0.08	Download
○	7	0.90	1.12	-0.19	Download
○	8	0.93	0.81	0.01	Download
○	9	0.88	1.47	0.10	Download
○	10	0.92	1.06	0.11	Download

(a) TM-score is caculated between the designed sequence and the scaffold structure.

(b) RMSD is caculated between the designed sequence and the scaffold structure.

(c) C-score typically ranges from [-5,2] and is a quantitative measure of the confidence of each model. The higher the C-score, the higher the confidence in the model.

Figure 4.8 I-TASSER modeling of the top 10 designed sequences. If the user selects the option, the structures of the top ten designed sequences are modeled with the I-TASSER pipeline. For each model, a confidence score (C-score) of the folding simulation is calculated by $C\text{-}score = \ln\left(\frac{M/M_{tot}}{\langle RMSD\rangle}\cdot\frac{1}{N}\sum_{i=1}^{N}\frac{Z(i)}{Z_{0(i)}}\right)$, where M/M_{tot} is the fraction of the structure decoys generated by I-TASSER in the largest structure cluster, and $\langle RMSD\rangle$ is the average RMSD of the decoys to the cluster center. This term corresponds to the degree of convergence of the folding simulations. $Z(i)/Z_0(i)$ is the normalized significance score of the templates by the i^{th} threading program, where there are a total of *N* threading program used by I-TASSER for template identification. The C-score is normally in the range of [-5, 2], and a C-score >-1.5 usually indicates that the I-TASSER model has a correct fold with a TM-score >0.5 (17). Since not all designs can be folded by I-TASSER with a high confidence, the C-score can be used as an approximate assessment of the foldability of the designed sequences. In a large-scale experiment that examined the folding of designed sequences (214), it was shown that there is a strong correlation between the C-score of I-TASSER simulations and the folding rate of designed proteins, where 80% (or 100%) of designed sequences are experimentally foldable for sequences with an I-TASSER C-score >0 (or >0.8). In this figure, the TM-scores and RMSDs are between the I-TASSER model and the starting scaffold. Users are able to download the I-TASSER models from the provided links.

4.2 Concluding Remarks

The EvoDesign server is a fully automated, online tool for protein design and has the ability to design new protein sequences either as a free monomer (monomer design) or as a receptor in a protein-protein complex (interface design). Starting from the structural coordinates of a monomeric protein or complex, EvoDesign first collects homologous folds and protein interfaces from the PDB, from which monomeric and complex profiles are constructed separately. Next, the evolutionary profiles are combined with a newly developed physical energy function, EvoEF, to guide the replica-exchange Monte Carlo simulation in order to design new sequences. Finally, the designed sequences are clustered, and the final designs are chosen from the lowest free energy sequences in the largest clusters.

It is important to note that the core algorithm of EvoDesign has been preserved from previous iterations of the program. This algorithm was validated in a large-scale, *in silico* redesign experiment of >300 soluble protein folds (215). Moreover, from this experiment, five designed domains with variable fold types and sequence lengths were experimentally tested through circular dichroism and NMR spectroscopy. All five proteins (including heterogeneous nuclear ribonucleoprotein K domain, thioredoxin domain, light oxygen voltage domain, translation initiation factor 1 domain, and the CISK-PX domain) were soluble and possessed secondary structure as determined by circular dichroism, and three of the designed domains had stable folds as shown by 1D NMR data. The follow-up X-ray crystallography study (240) showed that the crystal structure of the EvoDesign designed CISK-PX domain is very similar (1.32 Å) to the target model generated by I-TASSER structure prediction.

In this work, we have extended the EvoDesign pipeline to enable the design of PPIs by incorporating an evolutionary interface potential and a new physical energy function into the program. Previous benchmark studies of our evolutionary interface potential demonstrated that its predicted $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ values, binding affinity change of protein complexes upon amino acid mutation, showed superior correlation with experimental values (216). The correlation was significantly higher than that produced by leading physics- and statistical-based methods. Most recently, we applied the new EvoDesign program to the redesign of the BIR3 domain of the X-linked inhibitor of apoptosis protein (XIAP) (213), whose primary function is to suppress cell death by inhibiting caspase-9 activity. However, the suppression of cell death by wild type XIAP can be eliminated by the binding of Smac peptides. Multiple biophysical experiments such as NMR chemical shift perturbation and isothermal calorimetry binding assays demonstrated that the redesigned XIAP domains can bind the Smac peptide with dissociation constants in the low nanomolar range, but do not inhibit the caspase-9 proteolytic activity *in vitro*. Detailed mutagenesis analyses demonstrated that the major driving force behind the successful redesign of the native XIAP-Smac interaction was the interplay of the evolutionary profiles and physical potential (213).

The physical energy function utilized by the previous version of EvoDesign was FoldX. FoldX was originally developed and optimized to predict protein stability change upon mutation and has been widely used in the protein science community. Our benchmark tests show that the newly developed EvoEF generates more accurate predictions than FoldX for both stability and binding affinity change upon mutation, where the latter is critical to PPI design/engineering. In addition to the improved prediction accuracy, EvoEF is significantly faster than FoldX when it comes to energy calculation. This is particularly important in extensive protein design simulations like EvoDesign, where the physics-based energy computation is one of the most time-consuming parts of the pipeline. FoldX's inefficient energy computation is partly due to the fact that, currently, only executables are provided for the software and the computational speed cannot be fully optimized by users. Therefore, an effective and efficient physical energy function should be very helpful to the protein science community. The EvoEF source code is freely available at https://zhanggroup.org/EvoEF/, where users can optimize the code and parameters according to their own needs. Text G.5 in the SI provides a detailed description of the commands and functions implemented in EvoEF.

Despite their effectiveness and efficiency, the evolutionary components of the EvoDesign potential can be limited by the availability of structural homologs in the PDB; in particular, the number of protein interface homologs identified by iAlign may be low. In a previous study, we found that the average number of interface homologs identified for a set of test complexes was approximately five (216). To address this issue, we recently tested a new method to construct interface profiles by combining the structural iMSA with sequence homologs from sequence-based PPI databases. Based on the preliminary data, the method shows promise to significantly increase binding affinity prediction accuracy and we plan to integrate it into EvoDesign after further optimization.

As one of the essential difficulties in computational protein design is the expensive cost of experimental validations, the EvoDesign server aims to provide various transparent intermediate data to allow for detailed annotation and analysis of the confidence of the designed sequences. With the continuous effort on the development and improvement of the scope and accuracy of the methodology, we believe the new EvoDesign pipeline should be a useful tool to the community, especially for scientists who have known protein structures but want to design new sequences with enhanced foldability and biological functionality.

4.3 Author Contributions

The findings from this study were published in the Journal of Molecular Biology (134) with myself (R.P.) and Dr. Xioqiang Huang (X.H.) as co-first authors, Dr. Dani Setiawan (D.S.) as co-Author, and Dr. Yang Zhang (Y.Z.) as corresponding author. R.P. developed the online server, analyzed the data, and drafted the manuscript and figures; R.P., X.H., and D.S. co-developed EvoDesign; R.P., X.H., and Y.Z. finalized the manuscript.

CHAPTER 5

FoldDesign: *De Novo* Protein Fold Design Through Sequence-Independent Fragment Assembly Simulations

In the previous chapter we covered our work in protein sequence design; thus, in this chapter we will cover the second major topic of the design problem, protein structure design. As has been mentioned in the preceding sections, the unique and varied functions performed by proteins are made possible by the diverse structural folds adopted by different protein molecules. However, as is the case for the protein sequence space, despite the enormous conformational space available, only a tiny portion appears in nature following billions of years of evolution, probably due to the selective pressures exerted by environmental constraints upon organisms (64). For example, there have been just under 1,500 protein folds classified in the SCOPe database (65) and studies have indicated that the current PDB is nearly complete, representing the vast majority of natural folds (66, 67). Given the vital importance of proteins to living organisms, there has been growing interest in designing artificial proteins with enhanced functionality beyond their native counterparts. However, many of the attempts have focused on generating new protein sequences starting from the structures of experimentally solved proteins (134, 241-243). While this may be effective in certain cases, protein design starting from solved structures is severely limited as nature has essentially sampled from an insignificant portion of the possible structure and function space, thereby greatly limiting the number of design applications.

Given these limitations, *de novo* protein design aims to create not only artificial protein sequences, but also novel structures tailored to specific design applications, e.g., with specific fold types or binding pockets, has gained considerable traction in recent years. For instance, approaches such as Rosetta have been applied to design proteins with promising therapeutic potential (68-70), novel ligand-binding activity (71, 72), and complex logical interactions (73). The core protocol that has enabled Rosetta to design new protein folds is fragment assembly, which involves the identification of small structural fragments from experimentally solved structures that match a desired fold definition and the assembly of the identified fragments to produce full-length structural folds (14, 128, 244). Notably, fragment assembly was adapted from the related field of protein structure prediction, where it has been among the most successful classical approaches to template-free structure modeling (14, 16, 90, 125).

Despite the successes, *de novo* protein design remains somewhat of an art form, where large-scale experimental optimization is often required to generate successful designs (68, 70). In particular, extensive user-intervention during scaffold creation and selection is frequently necessary (71, 147). Nevertheless, automated fold design tailored to specific applications is highly non-trivial because traditional homologous structure assembly programs often create folds that are similar to the template structures even when distracted with strong external spatial restraints (13, 25). Although *ab initio* fragment assembly approaches, such as QUARK (16) and Rosetta (14), can create template-free models, they need to start from specific natural sequences and often create conformations that either converge to specific folding clusters or are not protein-like (245). Furthermore, as we covered in Chapter 1, most of the successful *de novo* designs have highly idealized structures with optimized SS compositions that lack the complex irregularities often present in native proteins, where a significant portion of the designed folds are well represented in nature or may be described through ideal parametric geometries (148, 156-159). Thus, development of automated algorithms capable of precisely designing any required fold type, including those without structure analogs in the PDB or idealized SS compositions, with limited human intervention is critical to improve the scope and success rate of *de novo* protein design.

Toward this goal, we proposed a new automated pipeline, FoldDesign, to create desired protein folds starting from user-specified restraints, such as the secondary structure topology and/or inter-residue contact and distance maps, through sequence-independent replica-exchange Monte Carlo (REMC) simulations. Since the designed folds do not necessarily have experimental counterparts, we designed several objective assessment criteria based on the satisfaction rate of the input requirements and the folding stability of the designs, as outlined in Fig. 5.1. The results showed that FoldDesign can produce protein-like structural folds that closely recapitulate the input features with enhanced folding stability, significantly outperforming other start-of-the-art approaches on the large-scale benchmark tests. Importantly, this was demonstrated on a set of non-idealized, complex SS topologies and roughly 1/4 of the designs possessed novel folds that were not represented in the PDB, illustrating an important ability of the program to explore the areas of protein fold space unexplored by natural evolution. The online server, which presently supports fold design targets up to 1500 residues long, and the standalone package for FoldDesign are freely available to the community at https://zhanggroup.org/FoldDesign/ and https://github.com/robpearc/FoldDesign, respectively.

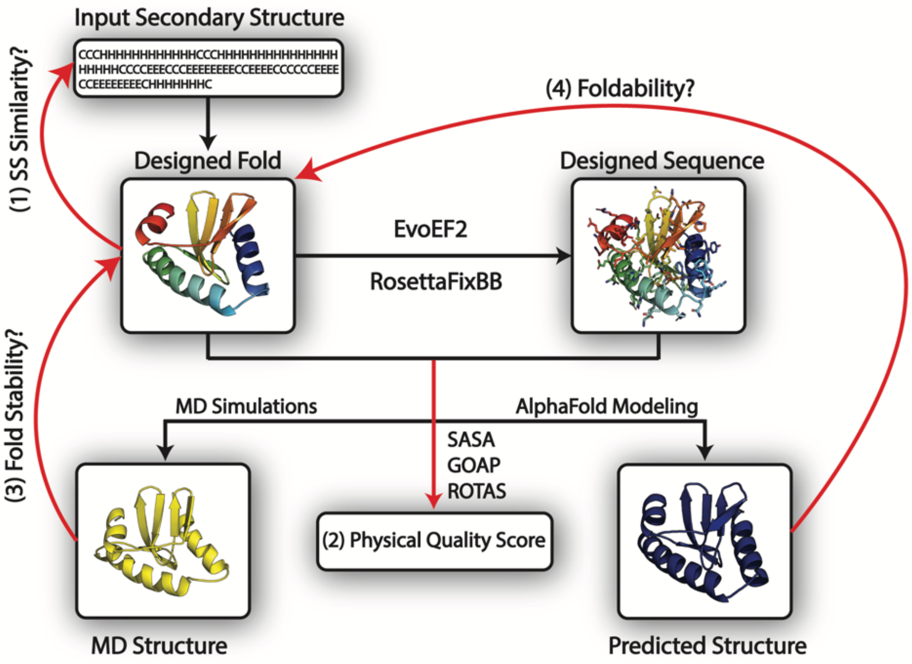


Figure 5.1 Illustrations of the strategies used to evaluate the quality of the FoldDesign scaffolds. The red lines mark the four criteria used to assess the FoldDesign scaffolds: (1) the secondary structure similarity between the input secondary structures and the secondary structures of the scaffolds designed by FoldDesign; (2) the physical quality score including hydrophobic core formation and statistical energies; (3) the fold stability assessed by the structural similarity (TM-score/RMSD) between the FoldDesign scaffolds and the final models after constraint-free molecular dynamic simulations (MD); (4) the foldability as determined by the structural similarity between the FoldDesign scaffolds and the predicted models by AlphaFold.

5.1 Results and Discussion

FoldDesign is an automated algorithm for sequence-independent, *de novo* protein fold design, where the flowchart is outlined in Fig. 5.2. The program takes as input the SS topology for a designed structure scaffold, which includes the length, order, and composition of the SS elements. A set of structural fragments with lengths between 1-20 residues is then collected from the PDB library by scoring the similarity between the input SS and the SS of the PDB fragments. These fragments are finally reassembled by REMC folding simulations to generate protein-like structural scaffolds that satisfy the input constraints, where the lowest energy structure is subjected to further atomic-level refinement to produce the final structural design (see 5.3 Methods).

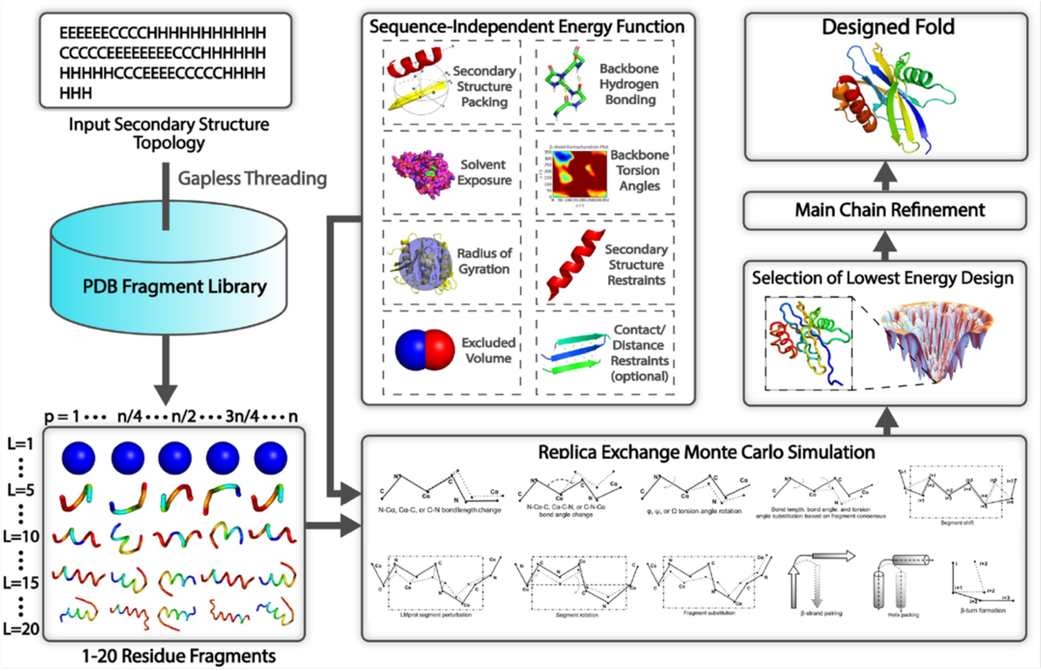

EEEEEECCCCHHHHHHHHHHHH
CCCCCEEEEEEEEEECCCHHHHHH
HHHHHCCCEEEEECCCCCHHHH
HHH
Input Secondary Structure Topology
Gapless Threading
PDB Fragment Library
p = 1 ••• n/4 ••• n/2 ••• 3n/4 ••• n
L=1
L=5
L=10
L=15
L=20
1-20 Residue Fragments
Sequence-Independent Energy Function
Secondary Structure Packing
Backbone Hydrogen Bonding
Solvent Exposure
Backbone Torsion Angles
Radius of Gyration
Secondary Structure Restraints
Excluded Volume
Contact/ Distance Restraints (optional)
Designed Fold
Main Chain Refinement
Selection of Lowest Energy Design
Replica Exchange Monte Carlo Simulation
N-Cα, Cα-C, or C-N bondlength change
N-Cα-C, Cα-C-N, or C-N-Cα bond angle change
φ, ψ, or Ω torsion angle rotation
Bond length, bond angle, and torsion angle substitution based on fragment consensus
Segment shift
LMprot segment perturbation
Segment rotation
Fragment substitution
β-strand pairing
Helix packing
β-turn formation

Figure 5.2 Overview of the FoldDesign pipeline.Starting from a user-defined SS topology as well as any further design constraints such as inter-residue contacts or distances, FoldDesign identifies 1-20 residue structural fragments from the PDB with SSs that match the input constraints. These fragments are then assembled together along with 10 other conformational movements during the replica-exchange Monte Carlo folding simulations under the guidance of a sequence-independent energy function that accounts for the fundamental forces that underlie protein folding. The lowest energy structure produced during the folding simulations is selected for further atomic-level refinement by ModRefiner to produce the final designed structure.

5.1.1 Auxiliary movements improve the folding simulation efficiency and ability to identify low energy states

Fragment substitution is the predominant movement used by FoldDesign, which involves the replacement of a selected region of a decoy structure with the structure from one of the identified fragments collected from the PDB. However, fragment substitution may cause large conformational changes that prevent the movement from being accepted. To improve the simulation efficiency, FoldDesign introduces 10 auxiliary movements, including bond length and angle perturbations, segment rotations, torsion angle substitutions, and those that form packing interactions between specific SS elements (see Texts J.1 and J.2 in Appendix J).

Fig. 5.3.A displays the FoldDesign energies of the lowest energy structures produced for each of the 354 test SS topologies (see 5.3 Methods), either using the full set of 11 conformational movements or only using fragment substitution. Of note, the 354 test SS sequences were derived from native proteins, which include irregularities and non-ideal compositions, making it a rigorous test set to determine if a method can design stable structures given non-ideal SS definitions. It can be observed that the auxiliary movements enabled the simulations to find structures with significantly lower energies than those found using fragment substitution alone. Overall, the average FoldDesign energy of the best structures produced using the full movement set was -529.5 k_BT compared to -449.7 k_BT when using only fragment substitution, where the difference was statistically significant with a p-value of 2.1E-66 as determined by a paired two-sided Student's t-test. In addition to the improved ability to sample low energy states, the auxiliary movements reduced the simulation times required to fold the proteins. Fig. 5.3.B plots the simulation time versus the protein length for each of the test topologies. From the figure, a clear reduction in the simulation time required can be seen across all protein lengths, where the average time for the simulations with the full movement set was 9.6 hours compared to 22.8 hours for the simulations that used only fragment substitution. This reduction in simulation time is due to the fact that fragment substitution is computationally expensive and requires additional loop closure to ensure that it does not cause large downstream perturbations, while the auxiliary movements are comparatively fast.

In Figs. 5.3.C-D, we further present a representative case study for the topology from the PDB protein 1ec6A, which adopts an α/β fold. Fig. 5.3.C shows the conformational dynamics of the decoys produced during the lowest-temperature replica of the simulations using only the fragment substitution movement, while Fig. 5.3.D uses the full movement set. Specifically, the figures plot the TM-score between the decoy at REMC cycle *i* compared to cycle *i-1* from cycles 50-100. In Fig. 5.3.C, there are several plateaus where no movement could be accepted, leading to identical conformations between a number of the cycles, where the most notable plateau lasted for 11 cycles (cycles 59 through 70). On the other hand, with the full movement set in Fig. 5.3.D, no such plateaus were observed. Although several cycles had very similar folds, which may be caused by subtle conformational refinements such as bond length perturbation, none of the cycles had identical structures. As a result, the simulations using the full movement set generated a structure with an energy of -346.2 k_BT in 4.7 hours compared to a structure with an energy of -224.3 k_BT in 14.2 hours using only fragment substitution.

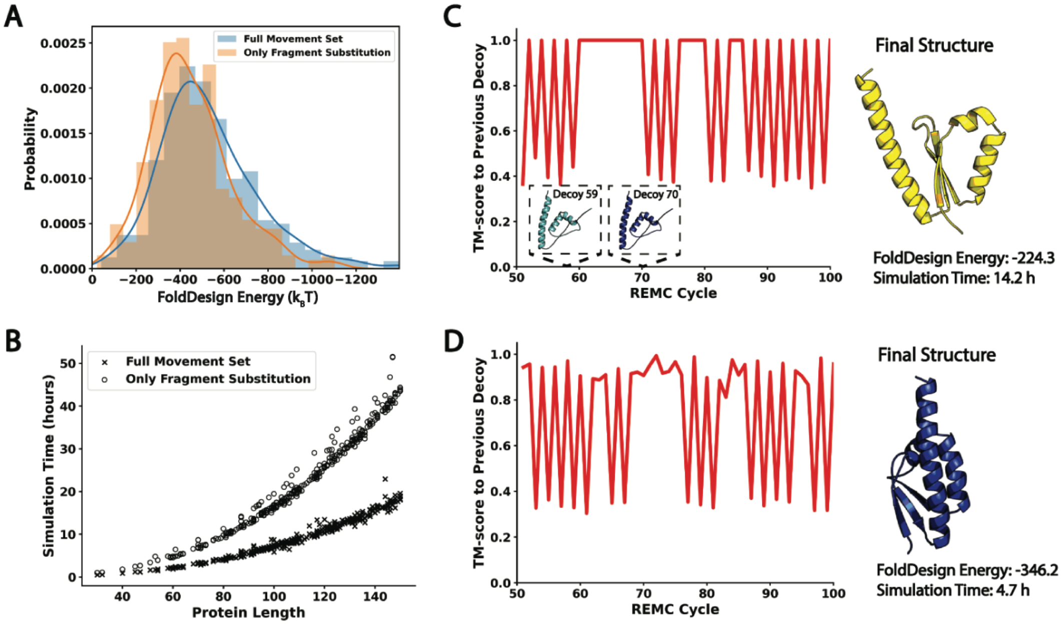


Figure 5.3 Importance of the auxiliary conformational movements. A) Energy distributions for the designs produced by the FoldDesign simulations using the full movement set and using only fragment assembly. B) Simulation time required versus protein length for FoldDesign using the full movement set and fragment assembly alone. C-D) Two representative case studies that demonstrate the dynamics of the folding simulations without (C) and with (D) the auxiliary movements. The y-axis displays the TM-score between the decoy at REMC cycle *i* compared to the decoy at cycle *i-1*.

As a comparison, Fig. I.1 in Appendix I depicts the native 1ec6A structure, which had a higher FoldDesign energy (-145.5 k_BT) than either of the simulated designs in Figs 5.3.C-D. This is expected as *de novo* protein design methods optimize the structure of a design with respect to their own energy functions and the native proteins from which an SS topology was derived will most likely never be the lowest energy conformation that the sampling procedures could/should achieve. Moreover, since many natural proteins with divergent global folds may adopt similar SS types, a given natural protein, such as 1ec6A, may not necessarily represent the most optimal fold or the lowest energy structure for a given SS composition, even with a perfect energy force field. In fact, it has been shown that many *de novo* designed proteins have increased stability compared to their native counterparts (148, 159). This is a departure from the scenario of protein structure prediction, in which the native structure, with some caveats, should lie at the global free energy minimum for a given protein sequence following Anfinsen's thermodynamic hypothesis (246); however, the same is not necessarily true for protein structure design given just the SS composition.

5.1.2 FoldDesign scaffolds closely match the input constraints

To assess its ability to design structural folds that possess the desired SS topologies, we list in Table 5.1 a summary of the FoldDesign results in terms of the average Q3 scores on the 354 test topologies. As a comparison, we also list the results from the state-of-the-art Rosetta method (156), which similarly starts from the desired SS of a designed scaffold, where a detailed description of the procedures used to run Rosetta is given in Text J.3. Here, the Q3 score is defined as the fraction of positions with SS elements that are identical to that of the input topology. Following fold generation, the SSs of the designed scaffolds for both FoldDesign and Rosetta were assigned using DSSP (239) and compared to the input for each protein.

Overall, FoldDesign achieved an average Q3 score of 0.877 compared to 0.833 for Rosetta with a p-value of 1.7E-08. When considering the Q3 scores for α-proteins, β-proteins, and α/β-proteins separately, FoldDesign achieved Q3 scores of 0.934, 0.863, and 0.875, compared to 0.828, 0.829, and 0.835, respectively, for Rosetta. Thus, across all fold types, FoldDesign was able to generate structures that more closely matched the input topologies than Rosetta. This partially reflects the advanced dynamics of the folding simulations as well as the effectiveness of the optimized energy function in FoldDesign.

Method	**Q3 Score All** **(*p*-value)**	**Q3 Score α-proteins** **(*p*-value)**	**Q3 Score β-proteins** **(*p*-value)**	**Q3 Score α/β proteins** **(*p*-value)**
FoldDesign	**0.877 (*)**	**0.934 (*)**	**0.863 (*)**	**0.875 (*)**
Rosetta	0.833 (1.7E-08)	0.828 (5.4E-05)	0.829 (0.10)	0.835 (4.5E-06)

Table 5.1 Comparison of the Q3 scores for the structures produced by FoldDesign and Rosetta on the 354 test SS topologies. Here, the Q3 score is defined as the fraction of positions in the designed structures whose SSs were identical to the input SSs. The results are further separated based on the fold type (α, β, and α/β) and the *p*-values were calculated using paired, two-sided Student's t-tests.

Although no user-defined distance restraints were included in the above tests, these are still important in many design cases where recapitulation of specific folds is desired. In Table 5.2, we extracted the pairwise Cα distances from the native structures in the test set and used them as restraints during the design simulations. From the table, it can be seen that FoldDesign was able to generate designs that closely matched the native structures with average TM-scores/RMSDs of 0.993/0.31Å, 0.993/0.27Å, 0.992/0.32Å, and 0.994/0.31Å for all, α, β, and α/β topologies, respectively. Therefore, the FoldDesign structures nearly perfectly recapitulated the desired folds when guided by user-defined distance restraints. Additionally, the Mean Absolute Errors (MAEs) between the Cα distance maps extracted from the designed folds and native structures were 0.148, 0.115, 0.130, and 0.154 Å for all, α, β, and α/β topologies, respectively, confirming that the generated structures closely satisfied the given distance restraints.

Protein Type	**MAE (Å)**	**TM-score**	**RMSD (Å)**
All	0.148	0.993	0.31
A	0.115	0.993	0.27
B	0.130	0.992	0.32
α/β	0.154	0.994	0.31

Table 5.2 Results of FoldDesign starting from distance restraints extracted from the native structures. All metrics were computed between the designed and native structures. Here, MAE is the mean absolute error between the Cα distance maps from the designed and native structures and is calculate by $MAE = \frac{\sum_{i=1}^{n}|x_i - y_i|}{n}$, where x_i is a distance from a designed structure, y_i is the corresponding distance from the native structure, and n is the number of considered distances.

5.1.3 FoldDesign generates low energy, native-like protein structures

While an important metric, the Q3 score is unable to provide a complete picture of the physical quality of the designs. In theory, a method could produce trivial or even unfavorable folds that satisfy the desired SS definitions. Thus, a more detailed analysis of the energetics and physical characteristics of the produced structures had to be performed (Fig. 5.1). As the designed scaffolds for FoldDesign and Rosetta are both sequence-independent and many of the traditional scoring and assessment tools are sequence-specific, the sequence for each scaffold had to be designed before further quantitative analysis could be conducted. To design the sequences for each scaffold, two sequence design methods were used, namely EvoEF2 (132), which is the latest iteration of the EvoEF design method described in Chapter 4, and RosettaFixBB (131), where the backbone structures of the designed scaffolds were kept fixed during the sequence design to ensure a fair comparison of the scaffolds that were directly output by FoldDesign and Rosetta. Here, RosettaFixBB and EvoEF2 are sequence design methods that perform Monte Carlo sampling in sequence space guided by combined physics- and knowledge-based energy functions. 100

sequences were designed for each scaffold, and the average results from the 10 lowest energy sequences were reported for both FoldDesign and Rosetta in the following analyses.

First, Fig. 5.4.A shows that the percent of buried residues for the FoldDesign scaffolds closely resembled the native protein structures from which the input SSs were extracted. For example, in the native structures, 19.2% of the residues were buried in the hydrophobic core, compared to 20.2% and 17.2% for the FoldDesign scaffolds whose sequences were designed by EvoEF2 and RosettaFixBB, respectively. However, for Rosetta, only 9.8% and 7.5% of the residues were buried in the hydrophobic core. Additionally, the solvent accessible surface area (SASA) for the native proteins was 7081.8 Å^2 compared to 6964.9 Å^2 and 7376.3 Å^2 for the FoldDesign scaffolds whose sequences were designed by EvoEF2 and RosettaFixBB, while the average SASA for the corresponding Rosetta scaffolds was 8721.2 Å^2 and 8944.2 Å^2, respectively. These results suggest that the FoldDesign scaffolds possessed more compact hydrophobic cores and less solvent exposed area than the Rosetta scaffolds and shared a higher similarity to the native structures for these characteristics. The difference is in part due to the fact that FoldDesign includes a number of energy terms that promote the formation of well-packed SS elements; these include specific fragment-derived distance and solvation potentials, generic backbone atom distance energy terms, and SS-specific fragment packing terms (see Text J.2). In addition, the energy weights were carefully optimized using the results of the design simulations to ensure the formation of well-folded globular proteins (see 5.3 Methods).

In Figs. 5.4.C-D, we further display the energies of the designed scaffolds by FoldDesign and Rosetta, as assessed by two leading third-party atomic-level statistical energy functions, GOAP (247) and ROTAS (248). For the sequences designed by EvoEF2 and RosettaFixBB, the FoldDesign scaffolds had average GOAP energies of -9736.9 and -10166.7, which were significantly lower than the GOAP energies of -8174.5 and -8838.8 for the Rosetta scaffolds with *p*-values of 3.4E-13 and 4.3E-10, respectively. Similar trends were observed for ROTAS. For the sequences designed by EvoEF2 and RosettaFixBB, the FoldDesign scaffolds had average ROTAS energies of -6110.3 and -4446.5 compared to -4360.8 and -3281.5 for the corresponding Rosetta designs; the differences were statistically significant with *p*-values of 6.8E-27 and 1.3E-15. Overall, the FoldDesign scaffolds possessed more tightly packed hydrophobic cores and were energetically more favorable than the Rosetta scaffolds, with GOAP energies that were 19.1% and 15.0% lower than the Rosetta scaffolds and ROTAS energies that were 40.1% and 35.5% lower than the Rosetta scaffolds depending on the sequence design method that was used. Importantly, neither FoldDesign nor Rosetta used any of the third-party energy functions for optimization.

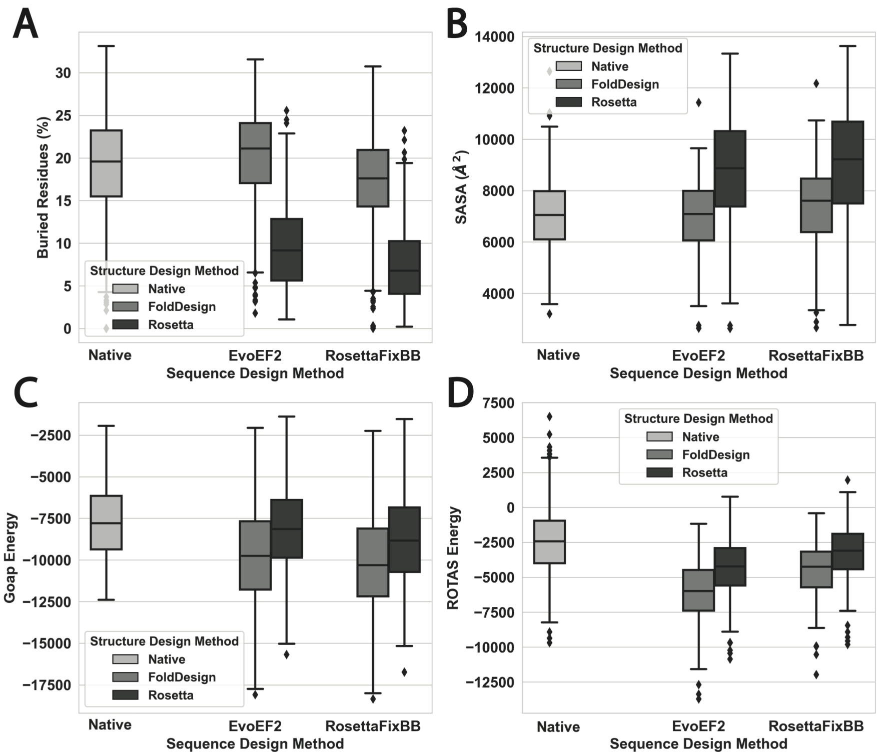


Figure 5.4 Comparison of the physical characteristics and energies for the designed folds by FoldDesign and Rosetta. The results are on the 354 test proteins, where the sequence for each scaffold was designed by EvoEF2 and RosettaFixBB, respectively. The native designation represents the proteins from which the SSs of the designed folds were derived. A) Proportion of buried residues is plotted for each protein, where a buried residue was defined as having a relevant solvent accessible surface area <5%. B) Solvent accessible surface area (SASA) for each protein. C-D) Energies for each protein calculated by GOAP and ROTAS.

Furthermore, introduction of ABEGO bias (249) during the Rosetta fragment selection protocol and enabling sub-rotamer sampling during the RosettaFixBB sequence design did not alter the above conclusions (see Text J.4 in Appendix J and Figs. I.2-3 in Appendix I). Lastly, despite the fact that Valine was used as the generic center of mass in FoldDesign and Rosetta (see 5.3 Methods), neither method demonstrated a bias towards scaffolds that favored Valine as described in Text J.5 and Fig. I.4 and all allowable regions of the Ramachandran plot were well represented in the FoldDesign scaffolds (Fig. I.5).

5.1.4 FoldDesign force field plays an important role in promoting the structural design performance

As shown in Eq. 5.1 in the Methods section, FoldDesign utilizes a number of newly introduced energy terms, including fragment-derived distance and solvation potentials ($E_{frag\_dist\_profile}$ and E_{frag_solv}) and detailed SS specific packing potentials (E_{hhpack} , E_{sspack} , and E_{hspack}), as well as generic atomic contact- and distance-based terms that promote the formation of compact, globular structures ($E_{generic\_dist}$ and $E_{contact\_num}$). Moreover, these terms were optimally combined with other more routine energy terms using an extensive weight optimization protocol based on the 107 training proteins (see 5.3 Methods).

To examine the impact of the FoldDesign force field and to probe the reason for the performance difference from the control method, we present in Fig. 5.5 the comparative results for the physical characteristics of the Rosetta designed scaffolds when the final models were selected using either the Rosetta or FoldDesign energy functions. It is noted that for this test we had to disable the fragment-derived distance and solvation potentials for FoldDesign as these are specific to the fragments generated by the FoldDesign program, which were not used to assemble the Rosetta designs given the differences in the fragment databases and identification protocols for the two methods. The data showed that selecting the Rosetta decoys according to their FoldDesign energies led to a significant improvement in the compactness of the folds as well as the GOAP and ROTAS energies compared to the designs selected using their original Rosetta energies. For example, selection using the FoldDesign energy function increased the percent of buried residues by 31.5% for the EvoEF2 sequence designs and 39.3% for the RosettaFixBB sequence designs, compared to selection by the Rosetta centroid energy function, where the differences were statistically significant with p-values of 1.6E-13 and 1.5E-14, respectively. Similarly, improvements were observed in the third-party energies of the designed scaffolds. For example, the average GOAP energy improved by 9.2% and 7.6% for the EvoEF2 and RosettaFixBB sequence designs, respectively, where the differences were significant with p-values of 3.1E-04 and 1.5E-03.

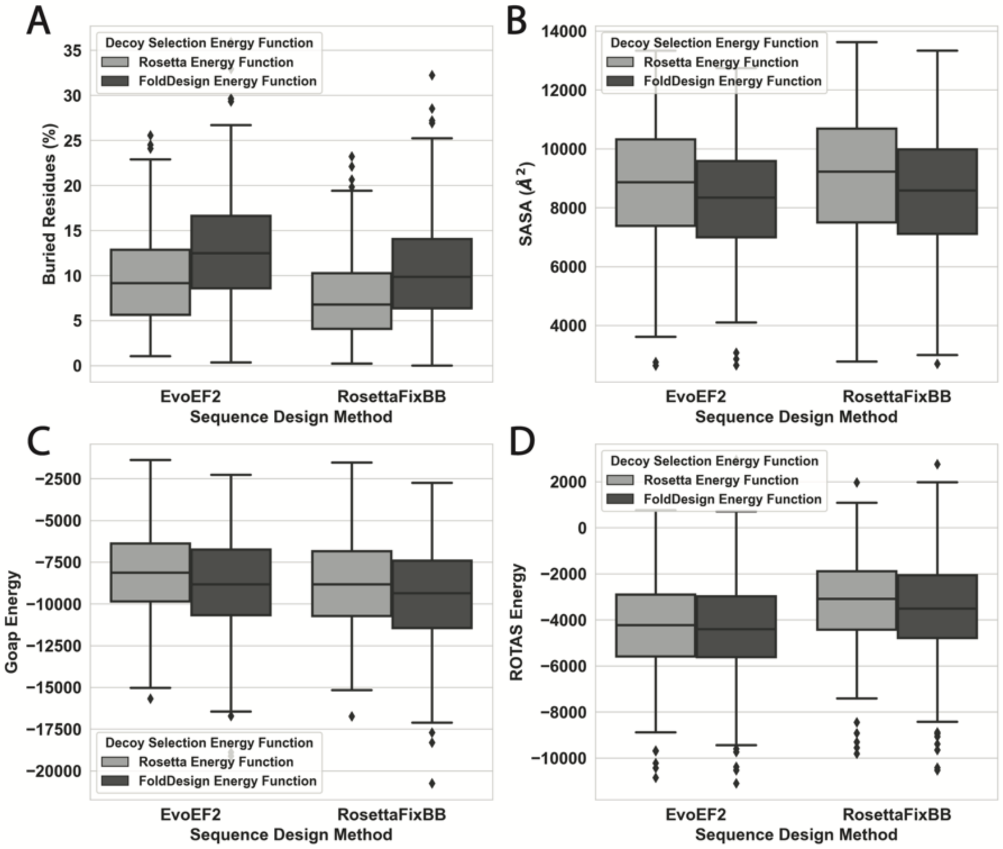


Figure 5.5 Comparison of the physical characteristics and energies for the designed folds by Rosetta. The results are on the 354 test proteins, where the final designs were selected using either the Rosetta centroid energy function or the FoldDesign energy function. A) Proportion of buried residues is plotted for each protein, where a buried residue was defined as having a relevant solvent accessible surface area <5%. B) Solvent accessible surface area (SASA) for each protein in the test set. C-D) Energies for each protein calculated by GOAP and ROTAS respectively.

In Fig. 5.6, we present a similar comparative result for the FoldDesign scaffolds when the final designs were selected by either the Rosetta or FoldDesign energy functions. For this test, an opposite trend was observed, where the selection of the FoldDesign scaffolds using the alternative force field from Rosetta resulted in a reduced performance compared to the original FoldDesign force field. For instance, the Rosetta energy-based selection led to a 43.2% and 49.4% decrease in the percent of buried residues for the EvoEF2 and RosettaFixBB sequence designs, compared to the models selected using the original FoldDesign energy function; these differences were statistically significant with *p*-values of 8.2E-79 and 5.8E-86, respectively. Furthermore, the GOAP energies were 26.7% and 25.2% worse for the EvoEF2 and RosettaFixBB sequence designs with *p*-values of 5.8E-35 and 9.8E-34, respectively. Based on the data shown in the above section, apart from the extensive REMC searching simulations, the optimized force field of FoldDesign, with newly introduced energy features, plays another critical role in creating compact and physically sound structure designs that outperform those from other state-of-the-art design methods.

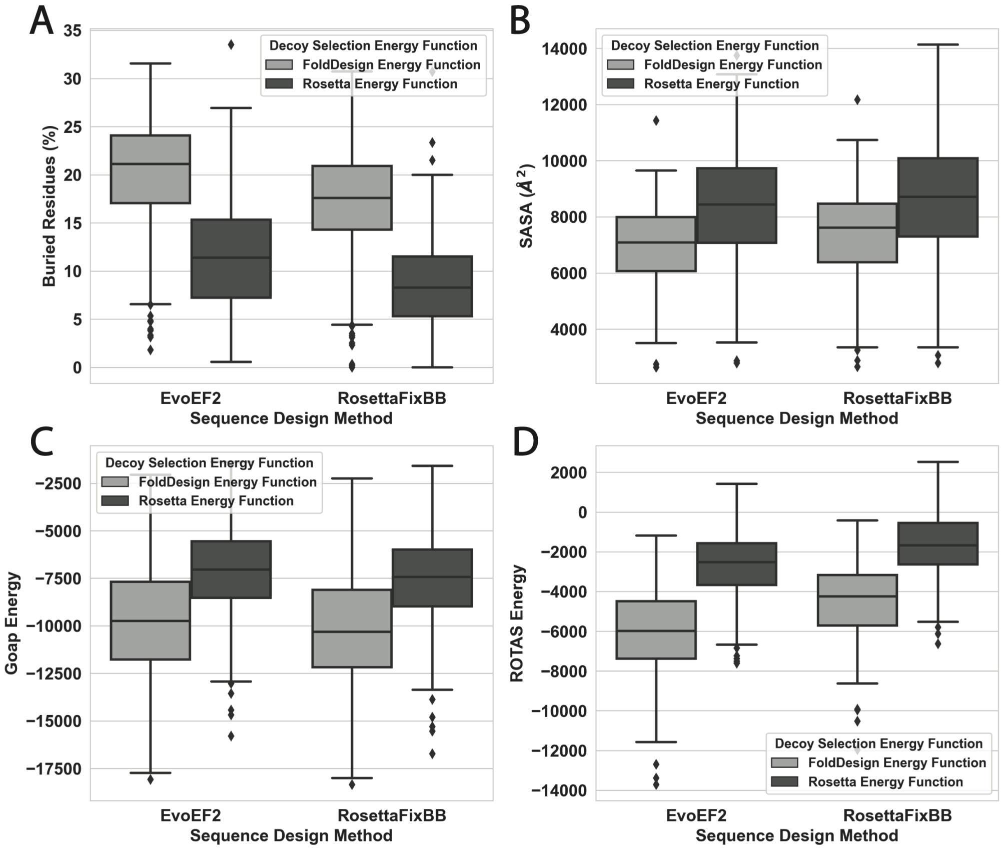


Figure 5.6 Comparison of the physical characteristics and energies for the designed folds by FoldDesign. The results are on the 354 test proteins, where the final designs were selected using either the FoldDesign energy function or the Rosetta centroid energy function. A) Proportion of buried residues is plotted for each protein, where a buried residue was defined as having a relevant solvent accessible surface area <5%. B) Solvent accessible surface area (SASA) for each protein in the test set. C-D) Energies for each protein calculated by GOAP and ROTAS respectively.

5.1.5 FoldDesign generates stable structures with novel folds

To further assess the stability of the designed structures, molecular dynamics (MD) simulations were run starting from the designed scaffolds produced by FoldDesign and Rosetta. MD is a useful tool as it allows for the study of protein motion and stability beyond static measurements such as energy calculations, where 20 ns unconstrained MD simulations were carried out using GROMACS (250) with the CHARMM36 force field (see 5.3 Methods). Following the simulations, the final MD structures were obtained by clustering the 1000 trajectories from the last nanosecond of each simulation using the GROMOS method with an RMSD cutoff of 2 Å, where the representative structure for each design was taken from the largest cluster center. To determine the stability of the structures, the TM-scores between the initial designed scaffolds and the final clustered MD structures were calculated, where the results are depicted in Figs. 5.7.A-B for the structures whose sequences were designed by EvoEF2 and RosettaFixBB, respectively.

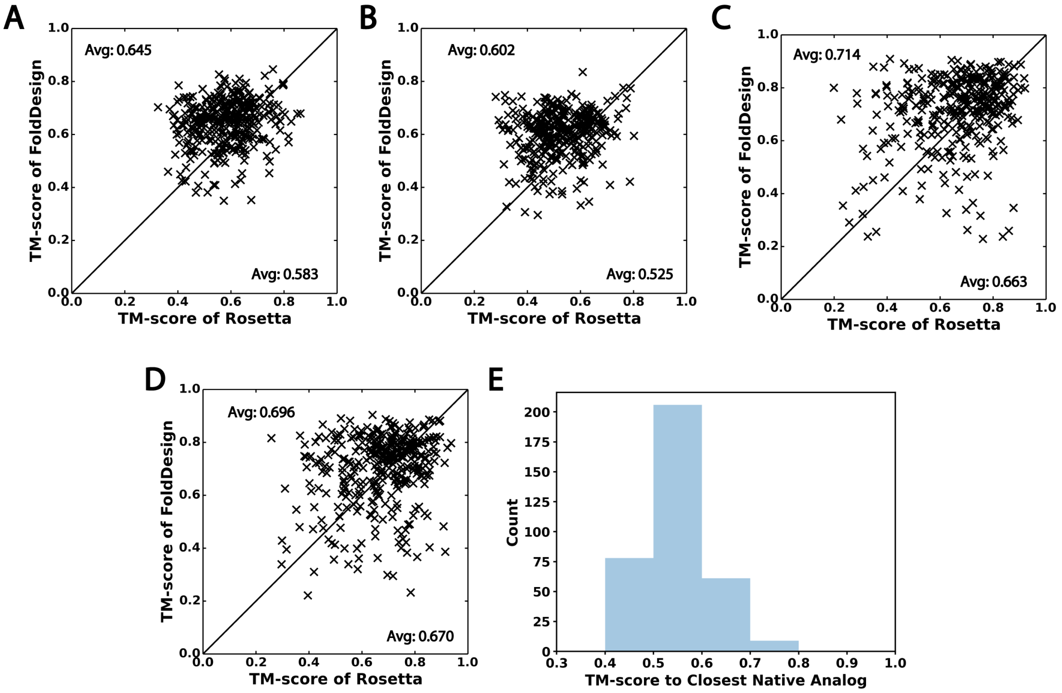


Figure 5.7 Analysis of the FoldDesign and Rosetta scaffolds using molecular dynamics (A-B) and protein structure prediction by AlphaFold2 (C-D). A-B) TM-scores of the FoldDesign and Rosetta scaffolds relative to their final structures following 20 ns MD simulations, where the sequence for each scaffold was designed by EvoEF2 (A) and RosettaFixBB (B). C-D) TM-scores of the FoldDesign and Rosetta scaffolds relative to the structures predicted by AlphaFold2 starting from the EvoEF2 (C) and RosettaFixBB (D) sequences designed for each scaffold. E) TM-score distribution between the FoldDesign structures and their closest native analogs obtained by searching the designed scaffolds through the PDB using TM-align.

From the figures, it can be seen that the TM-scores between the initial FoldDesign scaffolds and the final MD structures were higher than those for the Rosetta scaffolds, indicating a closer match and thus more stable conformations for the FoldDesign scaffolds against MD-based perturbations. For instance, the average TM-score between the FoldDesign scaffolds and final MD structures for the EvoEF2 sequence designs was 0.645 compared to 0.584 for the corresponding Rosetta scaffolds (Fig. 5.7.A), where the difference was statistically significant with a *p*-value of 7.4E-19. A similar trend was observed for the scaffolds whose sequences were designed by

RosettaFixBB, where the average TM-score between the initial FoldDesign structures and the final MD structures was 0.602 compared to 0.525 for the Rosetta scaffolds with a *p*-value of 4.6E-26 (Fig. 5.7.B). Furthermore, when considering a cutoff TM-score of 0.5, 93.7% and 87.9% of the FoldDesign scaffolds whose sequences were designed by EvoEF2 and RosettaFixBB, respectively, shared the same global folds as their final MD structures, compared to 77.1% and 54.8% of the corresponding Rosetta structures. Fig. 5.8.A shows three examples selected from among the most stable FoldDesign scaffolds, where the TM-scores were all greater than 0.8 and the RMSDs were less than 2Å, indicating a close atomic match between the designed scaffolds and the final MD structures. Overall, the vast majority of the FoldDesign scaffolds possessed stable global folds, outperforming the state-of-the-art Rosetta protocol across the test set.

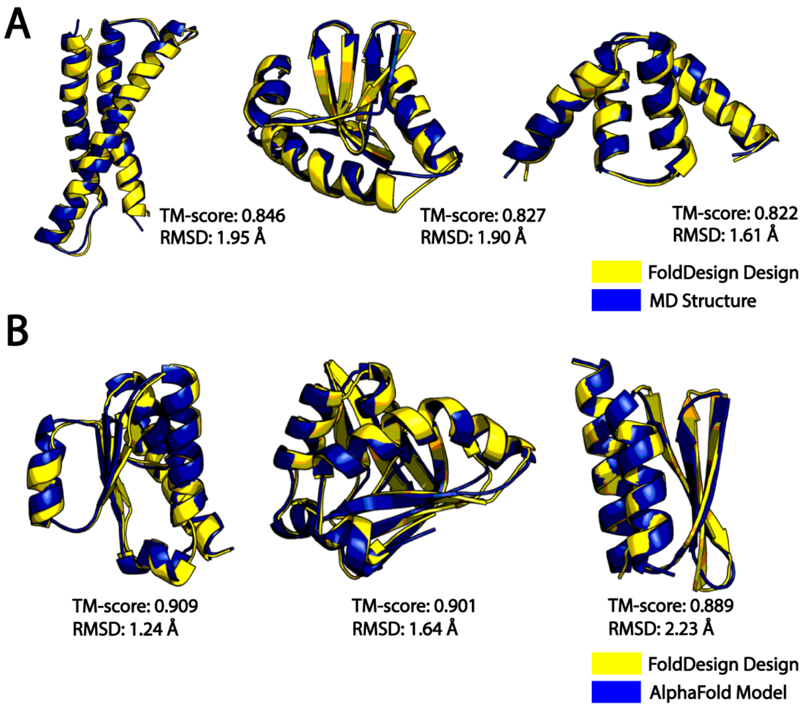


Figure 5.8 Examples of stable, well-folded FoldDesign scaffolds. The stability of the designs was assessed by molecular dynamics (A) and AlphaFold2 (B), where the sequences for each scaffold were designed by EvoEF2. A) The initial FoldDesign structures (yellow) superposed with the final MD structures (blue). B) The FoldDesign scaffolds (yellow) superposed with the AlphaFold2 models (blue).

Interestingly, despite the high fold stability with local structural features that were highly similar to the native proteins, a large portion of the FoldDesign scaffolds adopted novel folds that were different from what exists in the PDB. In Fig. 5.7.E, we present a histogram distribution of the TM-scores between the FoldDesign scaffolds and the closest structures identified by TM-align (220) from the PDB, where the average TM-score of 0.551 was relatively low given the searching power of TM-align and the near completeness of the PDB (66, 220). Of the 354 designs, 79 had a TM-score below 0.5 to any structure in the PDB, indicating they possessed novel folds, while the remaining 275 designs had analogous structures in the PDB with the same global folds (TM-scores ≥0.5). Furthermore, 74 of the 79 novel structures whose sequences were designed by EvoEF2 had stable folds with TM-scores ≥0.5 to their final structures output by the MD simulations. Moreover, there was no obvious difference between the novel folds and other folds in terms of stability, as the TM-score distributions between the designs and the final MD structures were quite similar (Fig. I.6), where their average TM-scores were 0.647 and 0.645, respectively. These results demonstrate that FoldDesign is capable of producing compact and stable scaffolds, while allowing for the exploration of novel areas of protein fold space.

5.1.6 Protein structure prediction indicates FoldDesign produces well-folded structures

As additional proof of the foldability of the designed structures, we examined the structural similarity between the designed scaffolds and the predicted models generated by the state-of-the-art AlphaFold2 program (123) starting from the designed sequences for each scaffold. As protein structure prediction is essentially the inverse problem of protein design, it would stand to reason that well-formed structure designs should be able to be recapitulated starting from their corresponding designed sequences.

However, given that AlphaFold2 is a deep learning-based modeling program, its performance largely depends on collecting meaningful MSAs (123), yet *de novo* designed proteins almost always lack natural sequence homologs. To illustrate this, in Fig. I.7 we plot the number of Blast hits that were detected from the nr sequence database (E-value <1E-5) when starting from either a single designed sequence or from jumpstarting the Blast search using an alignment of all 100 designed sequences for each FoldDesign scaffold. As shown in Fig. I.7.A, no Blast hits were detected when starting from a single EvoEF2 sequence design and jumpstarting the Blast search from the alignment of designed sequences only picked up 1-2 hits for 4 of the 354 designs. For the RosettaFixBB designs, neither the single designed sequence searches nor the jumpstarted Blast searches yielded any detectable homologs (Fig. I.7.B).

In Table H.1 in Appendix H, we also list the structure prediction results by AlphaFold2 for the 354 native protein structures starting from the MSAs generated by the DeepMSA program (180) compared to the results starting from the single designed sequences. As expected, AlphaFold2 created excellent models with an average TM-score of 0.913 when starting from the native MSAs; but starting from the single designed sequences by either EvoEF2 or RosettaFixBB produced significantly less accurate models, where the average TM-scores were only 0.506 and 0.482 for the EvoEF2 and RosettaFixBB sequence designs, respectively, and nearly (or more than) half of the cases had TM-scores below 0.5. This result is in line with previous studies that have indicated that single sequence-based modeling using deep learning approaches for non-ideal folds is significantly less accurate than that for idealized *de novo* designed folds (251). This is likely due to the fact that most of the computationally designed structures have relatively simple global folds with optimized SS compositions that lack the irregularities that exist in native proteins (148, 157, 159). Since the 354 SS topologies in the benchmark dataset were derived from native protein structures, which contain numerous irregularities, the above results indicate that single sequence based AlphaFold2 modeling may not be reliable for the FoldDesign and Rosetta scaffolds. Interestingly, when starting from artificial MSAs collected from the 100 designed sequences for the native structures, AlphaFold2 could generate reasonable folding results, where more than 97% of the cases had TM-scores >0.5, which was close to the modeling results obtained when starting from the DeepMSA MSAs (see Table H.1). This demonstrates that the MSAs collected from sequence design simulations contain some level of evolutionary information that can facilitate deep learning-based structure prediction.

Thus, given the lack of natural sequence homologs and the difficulty of AlphaFold2 to model complicated folds from single sequence designs, we constructed the input MSAs for AlphaFold2 by taking the 100 sequences designed by EvoEF2 and RosettaFixBB for each of the FoldDesign/Rosetta scaffolds. As shown in Table 5.3, when starting from the sequences designed by EvoEF2, the average TM-score between the AlphaFold2 models and the FoldDesign scaffolds was 0.714 compared to 0.663 for the Rosetta scaffolds, where the difference was statistically significant with a *p*-value of 4.6E-09. In Fig. 5.7.C, we present a head-to-head TM-score comparison, where the FoldDesign scaffolds had higher TM-scores than the corresponding Rosetta scaffolds for 211 cases, while Rosetta did so for 133 of the 354 cases. If we consider the number of designs with TM-score ≥0.5, 324 (or 91.5%) of the FoldDesign scaffolds shared the same global folds as the AlphaFold2 models compared to 301 (or 85.0%) of the scaffolds by Rosetta. These results demonstrate that the FoldDesign scaffolds more closely resembled the AlphaFold2 models than the Rosetta scaffolds did, indicating their enhanced stability/foldability. Similar patterns were observed for the sequences designed by RosettaFixBB, where the average TM-score between the FoldDesign scaffolds and AlphaFold2 models was 0.696 compared to 0.670 for Rosetta with a *p*-value of 3.0E-04 (Table S3). Moreover, 208 of the 354 FoldDesign scaffolds had higher TM-scores than the Rosetta scaffolds and 315 (or 89.0%) of the designs had TM-scores ≥0.5 (Fig. 5.7.D).

Method	TM-score (*p*-value)	RMSD (*p*-value)	# TM-score ≥ 0.5
Sequences designed by EvoEF2			
FoldDesign	**0.714 (*)**	**3.66 (*)**	**324**
Rosetta	0.663 (1.1E-07)	5.10 (4.6E-09)	301
Sequences designed by RosettaFixBB			
FoldDesign	**0.696 (*)**	**4.13 (*)**	**315**
Rosetta	0.670 (0.004)	4.95 (3.0E-4)	310

Table 5.3 Results of AlphaFold2 modeling starting from the designed sequences for the FoldDesign and Rosetta scaffolds. *P*-values were calculated using paired, two-sided Student's t-tests.

Fig. 5.8.B presents three examples from some of the closest matches between the FoldDesign scaffolds and AlphaFold2 models, where each had a TM-score greater than or close to 0.9 and RMSDs below 2.25 Å, indicating close atomic matches between the designed scaffolds and predicted models. Notably, these cases came from designs with some level of analogous structural information in the PDB, although the TM-scores between the designed scaffolds and their closest native analogs (0.517-0.611, see Fig. I.8) were much lower than those between the designed scaffolds and the AlphaFold2 predicted models (0.889-0.909, Fig. 5.7.B).

To further examine the foldability of the novel structures produced by FoldDesign, Fig. I.9 plots the AlphaFold2 TM-score distributions for the FoldDesign scaffolds that lacked or possessed native analogs, where the novel designs (with TM-score=0.723/0.718 for the EvoEF2/RosettaFixBB sequences) were found to be as foldable or even more so than those with native analogs (with TM-score=0.711/0.689 for the EvoEF2/RosettaFixBB sequences). Overall, these tests demonstrated that the FoldDesign scaffolds more closely matched the predicted models than the Rosetta scaffolds did, and the overwhelming majority of the designs shared the same global folds as the AlphaFold2 models. This structural consistency may suggest that FoldDesign captures some structural characteristics that have been integrated in the AlphaFold2 learning process.

5.1.7 Assembling uncommon structural motifs is essential to produce novel fold designs

Given the high population of novel folds produced by FoldDesign starting from native SS compositions, it was of interest to quantitatively examine the structural characteristics of these folds and determine how they deviate from native protein structures. Toward this goal, we first examined their local structural quality using MolProbity (MP) (252), where the results are summarized in Table H.2. It was observed that the novel designs possessed favorable MP-scores, with an average MP-score of 1.66 compared to 1.57 for the designs that had identifiable native analogs, where both scores were comparable to (or only slightly higher than) those of the corresponding native structures (Table H.2). Meanwhile, the novel folds had very few Ramachandran outliers, atomic clashes, or deviations in bond lengths and angles, largely comparable to (or slightly better than) the native and analogous designs. This result provides support that the novel folds possessed favorable local geometries and physical realism that resembled native proteins, although they had completely different global folds.

To further probe the source of the distinct structural folds adopted by the novel designs, following the idea of previous studies (253-255), we investigated the local geometries of the associated super-SS elements by decomposing the global folds into their local structural motifs (Smotifs). Briefly, a Smotif is composed of two adjoining regular SS elements, either helices or strands, that are linked by a loop region (253). As shown in Fig. 5.9, the geometry of a Smotif is specified by four spatial characteristics, including the distance (D) between the bracing SS elements and the three angles formed between them (hoist δ, packing θ, and meridian ρ). The overall fold of a protein can then be broken down into the basic SS building blocks, where a total of 540 Smotif types can be obtained by splitting the 4-dimensional (D-δ-θ-ρ) space into 4-3-3-6 intervals and only ~320-330 Smotif geometries can be used to describe all existing protein structures (254). In Fig. 5.10, we present the relative frequency of Smotifs in the 79 novel folds and 354 native proteins in the test set versus the normalized background frequency of the Smotifs calculated from 51,094 non-redundant full-chain structures in the I-TASSER template library (17, 256), where the relative frequency values were normalized for each protein across the four background frequency bins in the plot (see Text J.10).

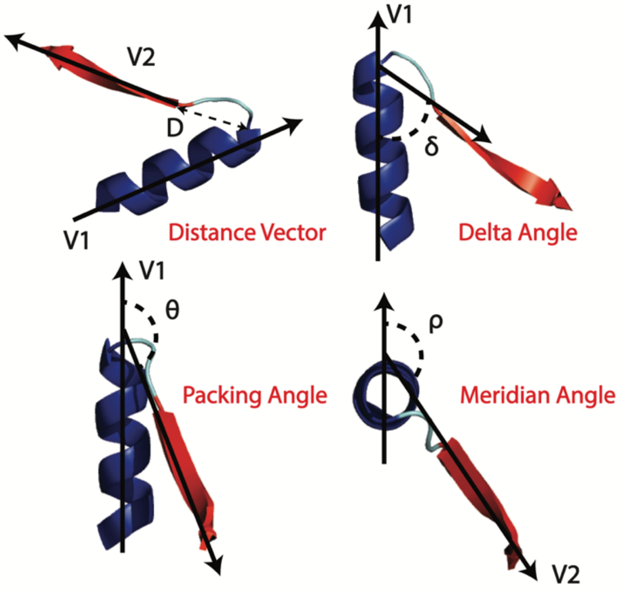


Figure 5.9 Smotif geometry definition. The axis for an α or β secondary structure is defined as the shortest of the principal moments of inertia of that structure, where V1 and V2 are the axis vectors of the secondary structure. The geometry of each motif is defined by four geometric features: (1) D, the distance between the ending points of the two secondary structure elements, (2) Hoist angle, δ, the angle between axis V1 and vector D; (3) Packing angle, θ, the angle between V1 and V2; and (4) Meridian angle, ρ, the angle between V2 and the plane that contains the vector V1.

It can be observed from Fig. 5.10 that compared to the native proteins, the novel designs by FoldDesign were highly enriched for rare or uncommon Smotifs, where 24.5% and 70.8% of the Smotifs in the novel designs had normalized background frequencies in the range [0, 1E-3] and (1E-3, 1E-2], respectively, compared to just 4.5% and 29.7% for the 354 native proteins. Additionally, 50.6% of the Smotifs from the native folds were common with background frequencies >1E-1, while just 4.1% of the Smotifs from the novel designed folds were commonly found. Of note, the vast majority of the Smotifs in the novel designs were found in nature, with the exception of one geometry that did not appear in the proteins from the PDB as shown in Fig. I.10. Thus, the novelty of the designed folds by FoldDesign may largely be a consequence of the combination of rare/uncommon local super-SS geometries, rather than the creation of new local geometries or a unique arrangement of common structural motifs. Furthermore, given the computationally assessed stability of the novel folds, these results support the claim that FoldDesign is able to produce stable designs for non-idealized SS elements, as the majority of the super-SS geometries were rarely observed in nature.

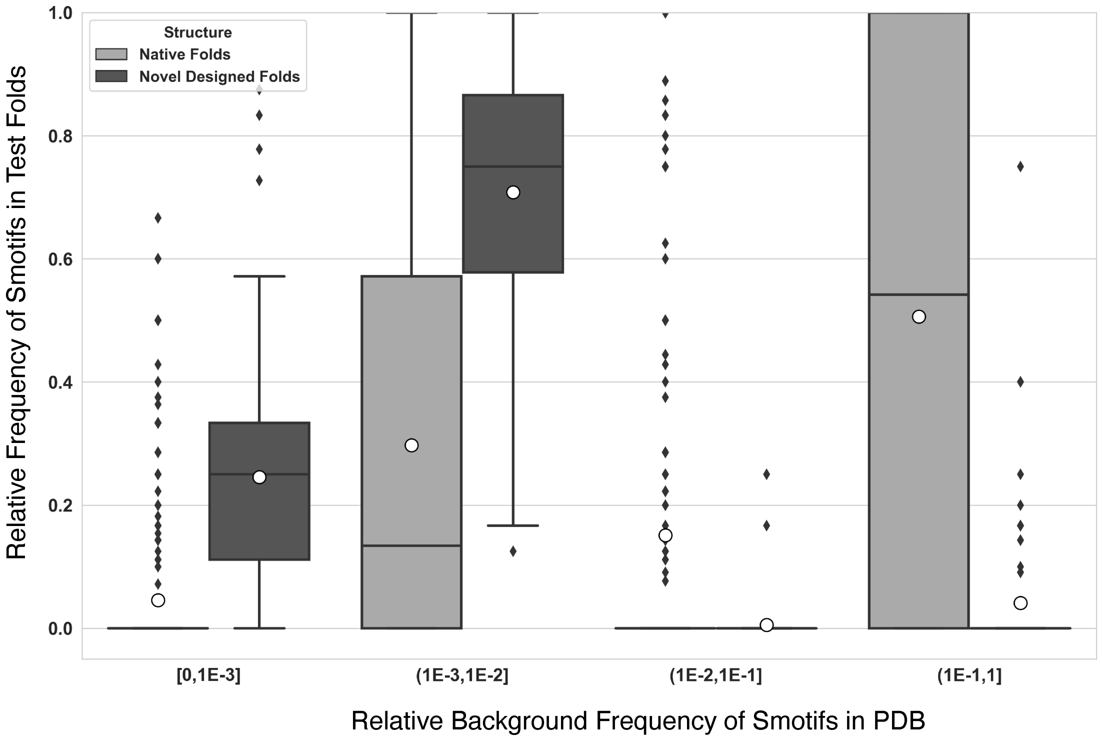


Figure 5.10 Relative frequency of Smotifs found in the 354 native protein structures and 79 novel folds produced by FoldDesign. The x-axis plots the normalized background frequency of the Smotifs calculated from the 51,094 non-redundant full-chain structures in the I-TASSER template library (see Text S10 in the SI). Two motifs are considered as identical if they fall into the same bin in the 4-dimensional (D-δ-θ-ρ) space (254). The mean values of the distributions are shown by the white circles, where a point with 0-frequency indicates that a Smotif with the indicated background frequency did not appear in one of the tested structural folds.

Fig. 5.11 highlights two design cases with novel folds whose SS compositions were taken from the PDB proteins 1id0A and 2p19A, where the designed scaffolds are shown superposed with their AlphaFold2 models and closest native analogs from the PDB. It can be observed that the AlphaFold2 models closely resembled the designed scaffolds with TM-scores of 0.809 and 0.811 for the 1id0A and 2p19A designs, respectively, indicating they were foldable by the deep learning program. Interestingly, the clusters that these designs were selected from were highly conserved with average TM-scores of 0.769/0.826 between the cluster members and 1id0A/2p19A, pointing to a clear evolutionary relationship between the SS topologies and the native folds. Despite this, FoldDesign generated novel scaffolds for these two topologies, which had low TM-scores (0.467 and 0.451) to their closest structures in the PDB, again demonstrating an ability to explore structure space unexplored by nature even for highly conserved clusters.

In the right column of Fig. 5.11, we illustrate the Smotifs that the two designs were composed of, where the Smotifs for the two native structures are shown in Fig.I.11. For the 1id0A topology design, the global structure was composed of 8 Smotifs, where all 8 were rare with a background frequency ≤1E-3, while the corresponding native structure was composed of 7 common Smotifs with a high background frequency of ~3E-1 and 1 Smotif that was less common with a background frequency of ~1E-2. Similar trends were observed for 2p19A, where the designed structure was composed of 8 uncommon Smotifs with a background frequency ≤1E-2, while the native structure was composed of 8 common Smotifs with a background frequency of ~3E-1. Thus, from these cases, it can be seen that the combination of rare or uncommon local super-SS geometries gave rise to new global folds, which was observed across the 79 novel designs.

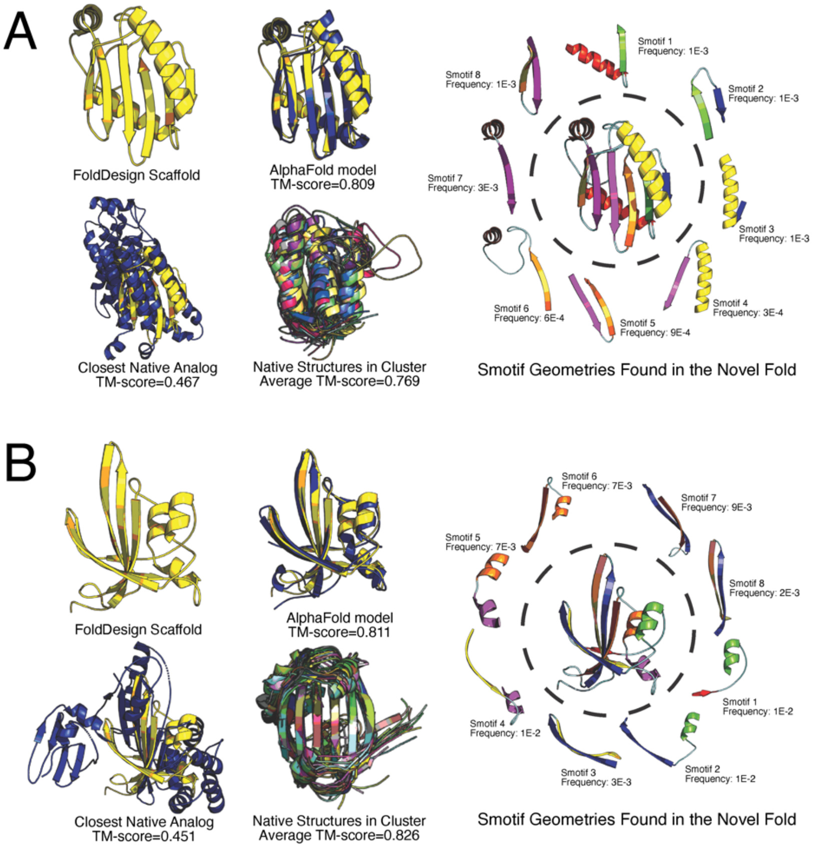


Figure 5.11 Case study of two novel designed folds for the SS topologies taken from 1id0A (A) and 2p19A (B). The designed structures are shown on the left-hand side of the figure in yellow superposed with their AlphaFold2 models and closest native analogs in blue. Additionally, each native structure in the same SS cluster as 1id0A (A) and 2p19A (B) are shown aligned with their respective cluster centers, where the average TM-scores were calculated based on the alignment of each structure in the cluster to the cluster center. Lastly, the right-hand side of the figure illustrates the Smotif geometries found in the novel folds, where the depicted frequencies for each Smotif represent the relative background frequencies calculated from the representative structures in the PDB.

5.2 Concluding Remarks

Protein design generally consists of two steps of structural fold design and sequence design. Many protein design efforts have focused on the second step of sequence design with input scaffolds taken from existing protein structures in the PDB. Despite the success, such experiments constrain design cases to the limited number of folds adopted by natural proteins, while curtailing the exploration of novel areas of protein structure and biological function.

In this work, we developed a pipeline, FoldDesign, for *de novo* protein fold design. Different from traditional protein folding simulations which start from native sequences and therefore, as expected, often result in folds that are similar to what exists in the PDB library, FoldDesign starts from structural restraints (e.g., SS assignments and/or inter-residue distance restraints) and performs folding simulations under the guidance of an optimized sequence-independent energy function. Large-scale tests on a set of 354 unique, non-ideal fold topologies demonstrated that FoldDesign could create protein-like folds with a closer Q3 score similarity to the desired structural restraints than the state-of-the-art design program, Rosetta. Meanwhile, the FoldDesign scaffolds had well-compacted core structures with buried residue rates and solvent exposed areas that more closely matched those of native proteins, while MD simulations showed that the folds were more stable than those produced by Rosetta. Importantly, FoldDesign is capable of designing folds that are completely different from the native structures in the PDB, highlighting its ability to explore novel areas of protein structure space despite the high fidelity to the input restraints and the native-like local structural characteristics. Detailed data analyses showed that the major contributions to the success of fold design lie in the optimal energy force field, which contains a balanced set of energy terms that account for fragment and SS packing, as well as the efficient exploration of conformational space through REMC simulations assisted with a composite set of efficient movements. It was also found that the ability to identify and assemble less common super-SS geometries from the PDB, rather than creating new motifs or the unique arrangement of common SS motifs, represents the key for FoldDesign to create novel fold designs.

Although the FoldDesign server outputs both the designed fold and the lowest energy designed sequences when combined with the EvoDesign/EvoEF2 programs (132, 134), the validation of the designed sequences remains to be experimentally examined. However, complete experimental validation requires both designed structures and designed sequences, where the latter is out of scope of the present study, and we leave this important work to future investigation. Nevertheless, the findings presented here have shown that FoldDesign can be used as a robust tool for generating high-quality, stable structural folds when applied to the very challenging task of completely *de novo* scaffold generation without human-expert intervention. This therefore provides a strong potential for the experimental protein design to effectively explore both structural and functional spaces which natural proteins have not reached despite billions of years of evolution.

5.3 Methods

FoldDesign aims to automatically design desired protein structure folds starting from user-specified rules such as SS composition and/or inter-residue contact and distance maps. The pipeline consists of three main steps, including fragment generation, REMC folding simulations, and main chain refinement and fold selection (see Fig. 5.2).

5.3.1 Fragment generation

Starting from a user-specified SS, high-scoring fragments are identified from a fragment library, which consists of structural fragments collected from a non-redundant set of 29,156 high-resolution PDB structures used by QUARK (16, 92). The fragments were collected from structures deposited on or before 4/3/2014 and shared <30% sequence identity to each other (16, 92). Notably, this library has been extensively validated in the related field of protein structure prediction in the most recent CASP experiments (98, 164). Gapless threading through the library is performed to generate 1-20 residue fragments, where the fragments are scored based on the compatibility of their torsion angles and SS similarity to the desired SS at each position. The top 200 fragments are generated for each overlapping 1-20 residue window. The information for each fragment includes the backbone bond lengths, bond angles, and torsion angles, as well as other useful data such as the position-specific solvent accessibility and Cα coordinates, which are later used to derive distance and solvation restraints.

5.3.2 REMC folding simulations and refinement

Following fragment generation, REMC folding simulations are performed in order to assemble full-length structural models, where each simulation uses 40 replicas and runs 500 REMC cycles (see Text J.1 for a full description of the REMC parameters and movements). The protein conformation in FoldDesign is represented with a coarse-grained model, which specifies the backbone N, Cα, C, H, and O atoms as well as the Cβ atoms and an atom that represents the side-chain center of mass (Fig. 5.12). To allow for a less biased exploration of structure space, the energy terms used by FoldDesign are sequence-independent, where the side-chain center of mass for Valine is used as the generic center of mass for each residue to minimize steric clashes.

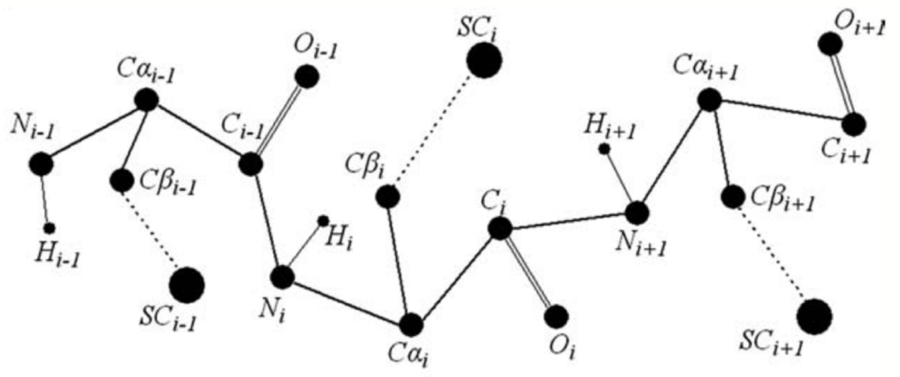


Figure 5.12 Depiction of the reduced model used to represent protein conformations during the FoldDesign simulations. The conformations include the backbone atoms (N, H, Cα, C, and O) as well as the Cβ atoms and side-chain centers of mass (SC). The center of mass for Valine is used in this study to evaluate steric clashes.

The initial conformations are produced by randomly assembling different high-scoring 9 residue fragments and then minimized using a set of 11 movements. Here, the major conformational movement is fragment substitution, which involves swapping a selected region of a decoy structure with the structure from one of the fragments randomly selected from the fragment library. Next, cyclical coordinate descent loop closure (91) is used to minimize the structural perturbations downstream. Since FoldDesign uses 1-20 residues fragments, larger fragment insertions are typically attempted during the initial REMC cycles, while smaller ones are attempted during the later steps of the simulations to improve its acceptance rate when the protein is more globular and well-folded. In addition to fragment insertion, 10 other conformational movements are attempted throughout the course of the simulations, including perturbing the backbone bond lengths, angles or torsion angles, segment rotations, segment shifts, and movements that form specific interactions between different SS elements, where these are described in Text J.1 and Fig. 5.13.

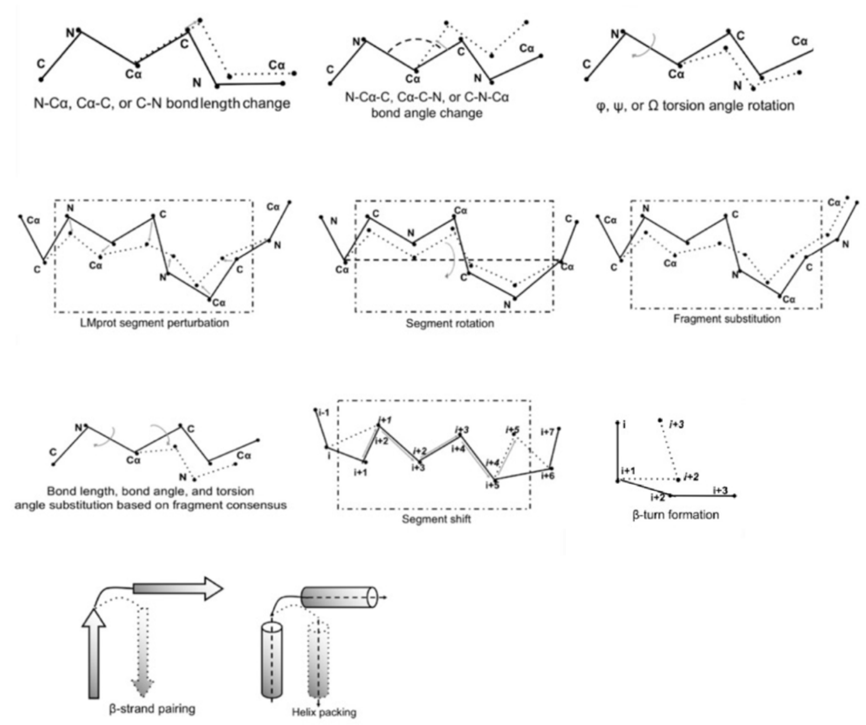

N-Cα, Cα-C, or C-N bondlength change
N-Cα-C, Cα-C-N, or C-N-Cα bond angle change
φ, ψ, or Ω torsion angle rotation
LMprot segment perturbation
Segment rotation
Fragment substitution
Bond length, bond angle, and torsion angle substitution based on fragment consensus
Segment shift
β-turn formation
β-strand pairing
Helix packing

Figure 5.13 Depiction of the conformational movements used by FoldDesign, with explanations in Text J.1 in Appendix J.

The movements are accepted or rejected using the Metropolis criterion (231), where the energy for each conformation is assessed by the following energy function:

$$E_{\text{DeepFold}} = E_{HB} + E_{ss\_satisfaction} + E_{rama} + E_{hhpack} + E_{sspack} + E_{\text{hspack}} + E_{ev} + E_{generic\_dist} + E_{frag\_dist\_profile} + E_{frag\_solv} + E_{rg} + E_{contact\_num} \quad (5.1)$$

Here, E_{HB}, $E_{ss\_satisfaction}$, E_{rama}, E_{hhpack}, E_{sspack}, E_{hspack}, E_{ev}, $E_{generic\_dist}$, $E_{frag\_dist\_profile}$, E_{frag_solv}, E_{rg}, and $E_{contact\_num}$ are terms that account for backbone hydrogen bonding, the satisfaction rate of the input SS, Ramachandran torsion angles, helix-helix packing, strand-strand packing, helix-strand packing, excluded volume, generic backbone atom distances, fragment-derived distance restraints, fragment-derived solvent accessibility, radius of gyration, and expected contact number, respectively. A more detailed explanation of these terms is given in Text J.2. After the REMC simulations are completed, the design with the lowest energy is selected for further atomic-level refinement, for which sequence design and structural refinement are performed iteratively using EvoDesign (134) and ModRefiner (257), respectively.

5.3.3 Training and test dataset collection

To test FoldDesign's ability to perform *de novo* protein fold design, we collected a non-redundant set of SS sequences. This was accomplished by extracting the 3-state SSs from 76,166 protein domains in the I-TASSER template library (17, 256) using DSSP (239). All of the pairwise SS alignments were obtained using Needleman-Wunsch dynamic programming to align the 3-state SS sequences. The target sequences were then clustered based on the distance matrix defined by their SS identities, i.e., the number of identical SSs divided by the total alignment length, where an identity cutoff=70% was used to define the clusters.

The identified clusters were further refined by eliminating atypical SS topologies (clusters with less than 10 members) and by selecting only those clusters where a clear relationship existed between the SS and the tertiary structure adopted by the cluster members. The latter requirement was accomplished by using TM-align (220) to perform structural alignment between each cluster member and the cluster center, where conserved clusters were required to have an average TM-score ≥0.5 between the members and cluster center. Finally, we obtained 461 clusters; 107 and 354 SS sequences were used for the training and test sets, respectively. The training set was composed of 22 α, 25 β, and 60 α/β topologies, while the test set was composed of 24 α, 55 β, and 275 α/β topologies.

5.3.4 FoldDesign energy function optimization

In order to ensure proper structure generation, each energy term must be carefully weighted in the FoldDesign energy function. This was done on the 107 training topologies. Briefly, a grid searching strategy was used to optimize the weights, where all weights were initially assigned as 0, except for the weight for the steric clash term, which was set to 1.0. Then the values for each weight were adjusted one-at-a-time around the grid values and the FoldDesign simulations were run to produce scaffold structures using the new weight set. After structure generation, the sequences for each scaffold were designed using EvoEF2 (132) and the designed structures were assessed based on:

$$E_{accept} = -\Delta EvoEF2 + 100 * \Delta BuriedResidues + 100 * \Delta Q3Score \qquad (5.2)$$

where, $\Delta EvoEF2$, $\Delta BuriedResidues$, and $\Delta Q3Score$ are the changes in the average EvoEF2 energy, percent of buried residues, and SS Q3 score, respectively, between the structures produced by the old and new weight sets. If the new weighting parameter increased the value of E_{accept}, the weights were accepted. Once the initial weights for each energy term were determined, many more iterations were conducted to precisely fine-tune their values based on Eq. (5.2) as well as by hand inspection of the structures. Although time-consuming, the process of directly optimizing the weights based on the results of the folding simulations resulted in high quality scaffolds with physical characteristics that resembled native proteins.

5.3.5 Molecular Dynamics simulation for examining fold stability

To examine the stability of the FoldDesign scaffolds, we performed MD simulations starting from the designed structures. For each simulation, a dodecahedron box was constructed with a distance of 10 Å from the solute and filled with TIP3P water molecules, where Na^+ and Cl^- ions were used to neutralize the charge of the system. Following this, energy minimization was carried out using steepest descent minimization with a maximum force of 10 kJ/mol. The system was then equilibrated at 300 K using 100 ps NVT simulations and 100 ps NPT simulations with position restraints (1000 kJ/mol) on the heavy atoms of the protein. After the two equilibration phases, the system was well-equilibrated at the desired temperature and pressure, and unconstrained MD simulations were performed at 300 K for 20 ns. During the simulations, non-bonded interactions were truncated at 12 Å and the Particle Mesh Ewald methods was used for long-range electrostatic interactions. Lastly, the velocity-rescaling thermostat and Parrinello-Rahman barostat were used to couple the temperature and pressure, respectively. 1000 structures were collected from the MD trajectories during the final nanosecond of the simulations. This ensemble was then clustered using the GROMOS method with an RMSD cutoff of 2 Å, and the final MD structure for each simulation was collected from the cluster center.

5.4 Author Contributions

The findings of this study were published in the Proceedings of the National Academy of Sciences (258) with myself (R.P.) as first author, co-authors Drs. Xiaoqiang Huang (X.H.) and Gilbert S. Omenn (G.S.O.), and corresponding author Dr. Yang Zhang (Y.Z.). R.P. developed FoldDesign, performed the experiments, analyzed the data, developed the stand-alone package, and drafted the text and figures; X.H. assisted with the MD simulations; R.P., X.H., G.S.O., and Y.Z. finalized the manuscript.

CHAPTER 6

Conclusion

6.1 Summary

In this thesis, we have explored several topics surrounding the sequence-structure-function paradigm, which is the cornerstone of structural biology. Specifically, we have covered methods for the prediction of protein and RNA structures from sequence and the design of new protein molecules.

For protein/RNA structure prediction, we covered two methods, namely, DeepFold (Chapter 2) and DeepFoldRNA (Chapter 3). Starting from an MSA identified for a protein sequence of interest, DeepFold uses deep ResNets to accurately predict an ensemble of contact, distance, and orientation restraints, which are then converted into a potential and minimized using gradient-descent simulations. DeepFold demonstrated significant improvements in modeling accuracy compared to contemporaneous deep learning restraint-based approaches. Of particular importance was the modeling performance for targets with shallow MSAs, where DeepFold achieved an average TM-score that was ~40-45% higher than methods such as trRosetta and DMPfold, while being 262 times faster than traditional folding simulations.

Following the groundbreaking introduction of self-attention-based networks, we extended the DeepFold pipeline to DeepFoldRNA, which was the first available self-attention-based RNA structure prediction method. Similar to DeepFold, DeepFoldRNA generates predicted distance and orientation maps, which are converted to a potential and minimized using gradient-descent. DeepFoldRNA significantly outperformed other lead RNA modeling methods across the benchmark datasets, including the RNA-Puzzles dataset, achieving an average RMSD that was 4.18 Å lower than the best models submitted by any group (2.72 Å vs 6.90 Å). These methods demonstrate the advantage of deep neural network over human-engineered potentials at capturing the fundamental principles that underly the folding paradigm.

For protein design, we covered two approaches, EvoDesign (Chapter 4) and FoldDesign (Chapter 5). EvoDesign is an online webserver for functional protein sequence design. The method combines evolutionary monomer/interactions profiles with a physics-based energy function. The webserver is freely available to the community and aims to provide various transparent intermediate data to allow for detailed annotation and analysis of the confidence of the designed sequences. Lastly, FoldDesign is a program for *de novo* protein structure design through sequence-independent fragment assembly simulations. On a large benchmark dataset of non-idealized, complex SS topologies, FoldDesign was able to consistently generate stable structure designs, where ~1/4 of the designs possessed novel folds that were not represented in the PDB. This illustrates FoldDesign's ability to explore areas of protein fold space unexplored by natural evolution.

6.2 Future Directions

6.2.1 End-to-End Protein-RNA complex structure prediction

In Chapters 2 and 3, we covered the fields of protein and RNA structure prediction for monomeric input sequences. However, the functions of these molecules often involve interactions with different chains and the formation of complex structures. Therefore, the prediction of protein-protein, protein-RNA, and RNA-RNA complex structures is a critical problem in the field. Recently, AlphaFold2 was extended to AlphaFold2-multimer for the prediction of protein complex structures (259). The overall network architecture of AlphaFold2-multimer is nearly identical to that of the original AlphaFold2 pipeline for monomeric structure prediction. The main differences are the authors include an extra embedding that indicates the relative chain positions and which chains are homomers vs heteromers, as well as updating the pair embedding prior to the MSA embedding in the trunk layers in order to allow the processed intra-chain features to evolve independently. Besides these minor changes, the existing architecture was readily adapted to protein complex prediction and demonstrated excellent accuracy (259).

Currently, no such methods exist for RNA or protein-RNA complexes. Thus, we are working on developing these through the inclusion of a structure module in DeepFold/DeepFoldRNA. As with AlphaFold2, the structure module, which replaces the gradient-descent folding simulations, is composed of 3D equivariant transformer neural networks that treat each amino acid as a gas of 3D rigid bodies and allows for the direct generation of structure models. The network is trained end-to-end using the Frame-Aligned Point Error (FAPE) loss (123). The preliminary data are promising, and we hope to provide a method for atomic-level RNA and protein-RNA complex prediction.

6.2.2 Development of pretrained RNA language models

Although the use of the latest self-attention-based networks has reduced the influence of the MSA alignment depth on the structure modeling accuracy, there is still a noticeable effect. This is particularly apparent for orphan sequences with no homologous sequence information, or at least very little information (123). We witnessed this in Chapter 2, where DeepFold was able to achieve higher structure modeling accuracy than AlphaFold2 for the five targets in the dataset with normalized Neff values <0.2. For these targets, the average TM-score of DeepFold was 0.528 compared to 0.398 for AlphaFold2. Thus, the modeling performance of deep self-attention-based models remains to be improved for such cases.

One method to address this issue is through the incorporation of self-attention-based pre-trained language models into prediction approaches. Pre-trained language models are a powerful tool as they may be pre-trained for a task, i.e., masked token prediction, for which abundant data are available, and then effectively applied to another task for which little data exist. This has already been demonstrated for protein structure prediction, where methods such as the ESM models developed by Facebook AI Research were pretrained on large sequence databases to predict the masked amino acid identities, and then successfully applied to the problem of contact map prediction (177, 178). Although these models were not trained to perform contact map prediction, structural features like inter-residue contacts and distances are inherently encoded in the pairwise attention weights (177, 178). Recently, the ESM language models were incorporated into ESMfold, which is able to predict protein structures from single sequences with high accuracy (260). However, the ESM models were specifically trained for proteins, thus, the development of such networks for RNA may help greatly improve the modeling accuracy for orphan sequences or very shallow alignment depths.

6.2.3 Deep Learning-based Protein Design

Although protein design has witnessed less involvement of deep learning-based methods compared to structure prediction, this is beginning to change. For example, as mentioned in Chapter 1, Anishchenko *et al.* set out to answer the question if the information stored in deep neural networks used to predict inter-residue distances and orientations could be applied to design new protein sequences and structures (149). To address this, they used deep network hallucination, where they performed Monte Carlo sampling in sequence space, at each step feeding the sequences into the trRosetta deep neural network architecture in order to predict distance maps and comparing them against a background distance map distribution. Mutations were accepted or rejected based on the Metropolis criterion, where the objective of the simulations was to maximize the information gain (Kullback-Leibler divergence) between the predicted distance maps and the background distribution. The developed method was then extended in two additional studies, where the procedure was either completely constrained to design sequences for a fixed fold (261) or to design sequences that recapitulated native interfaces (150), while allowing the remainder of the protein to be hallucinated freely. The newest iteration of the approach incorporates RosettaFold into a diffusion model in order generate new protein sequences and structures (262).

These studies demonstrate that the information encoded in structure prediction networks may be applied to design new protein sequences and structures. Thus, we are currently working on developing deep generative models for protein design, specifically variational auto encoders. The networks are composed of an encoder-decoder scheme, where the backbone of each layer is an equivariant structure module similar to that used by AlphaFold2. During training, the encoder takes as input the sequence and structure of a native protein monomer or complex and projects it to a latent space dimension, the decoder then samples from the latent space and tries to recover the input. During inference, the latent space can be randomly sampled from for unconditional design or conditioned on a specific binding partner or other factor for functional protein design. The latent vector is then fed to the decoder and a new protein sequence/structure is generated, allowing for the robust generation of artificial proteins in a deep learning framework.

APPENDIX A

Supplementary Figures for Chapter 1

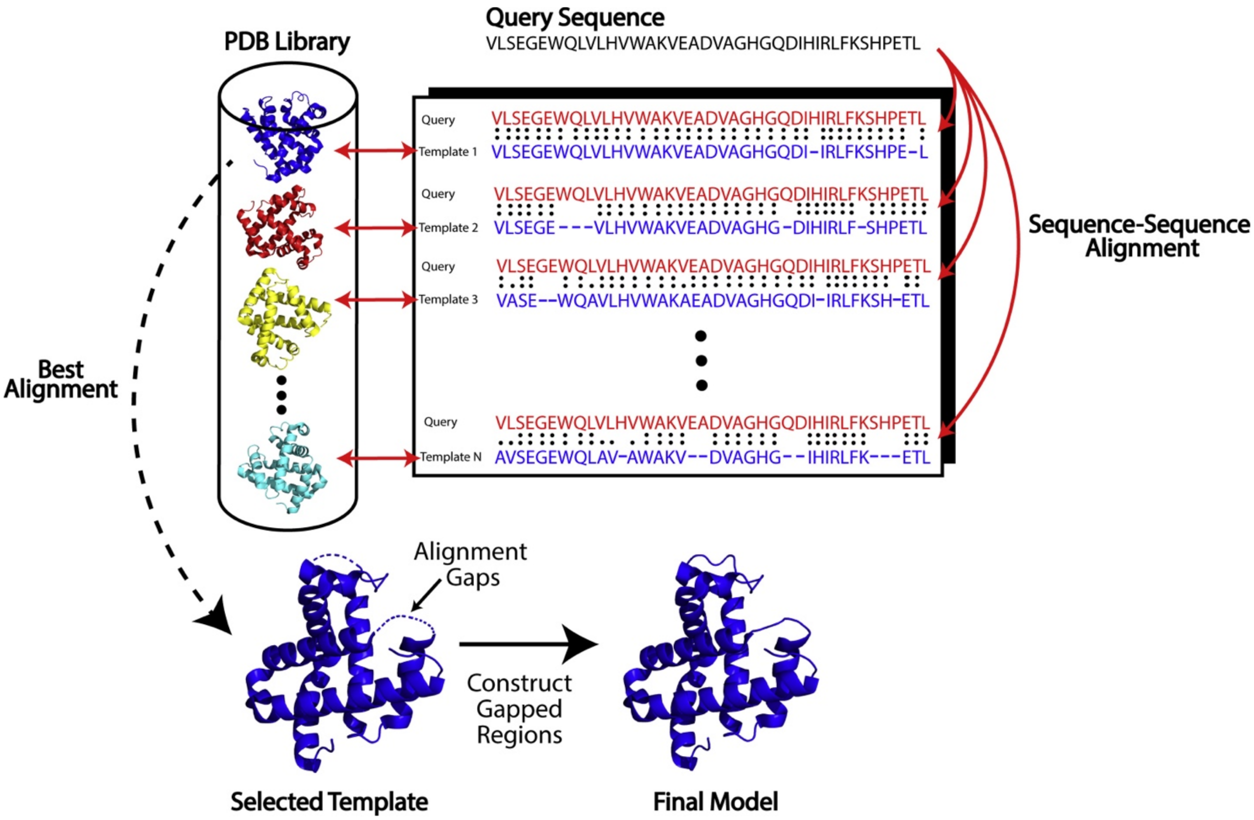

PDB Library
Query Sequence
VLSEGEWQLVLHVWAKVEADVAGHGQDIHIRLFKSHPETL
Query
VLSEGEWQLVLHVWAKVEADVAGHGQDIHIRLFKSHPETL
Template 1
VLSEGEWQLVLHVWAKVEADVAGHGQDI-IRLFKSHPE-L
Query
VLSEGEWQLVLHVWAKVEADVAGHGQDIHIRLFKSHPETL
Template 2
VLSEGE---VLHVWAKVEADVAGHG-DIHIRLF-SHPETL
Query
VLSEGEWQLVLHVWAKVEADVAGHGQDIHIRLFKSHPETL
Template 3
VASE--WQAVLHVWAKAEADVAGHGQDI-IRLFKSH-ETL
Query
VLSEGEWQLVLHVWAKVEADVAGHGQDIHIRLFKSHPETL
Template N
AVSEGEWQLAV-AWAKV--DVAGHG--IHIRLFK---ETL
Sequence-Sequence Alignment
Best Alignment
Alignment Gaps
Construct Gapped Regions
Selected Template
Final Model

Figure A.1 Typical steps in a homology-based modeling pipeline. Starting from a query sequence, templates are identified using sequence-based alignment algorithms. Then the structural framework of the best template alignment is copied, and the unaligned regions are constructed to produce the final model.

APPENDIX B

Supplementary Tables for Chapter II

Table B.1 Impact of the different components of the DeepFold energy function on the structure modeling accuracy. The accuracy is measured in terms of the average TM-score and the percent of correctly folded models (TM-scores ≥0.5) for the 221 benchmark proteins. The *p*-values were calculated using paired, two-sided Student's t-tests.

Energy Function	TM-score (*p*-value)	Correct Folds
GE	0.184 (8.4E-127)	0.0%
GE+Cont	0.263 (1.3E-118)	1.8%
GE+Cont+Dist	0.677 (1.9E-14)	76.0%
GE+Cont+Dist+Orien	**0.751 (*)**	**92.3%**

Table B.2 Impact of orientation restraints on folding convergence. Mean absolute error (MAE) between the distance maps predicted by DeepPotential and the distance maps of the 3D models built without (GE+Cont+Dist) and with (GE+Cont+Dist+Orien) inter-residue orientations. Here, the top *n**L* long-range distance restraints were sorted by their DeepPotential confidence scores. The *p*-values were calculated using paired, two-sided Student's t-tests.

Method	L/2 (*p*-value)	L (*p*-value)	2L (*p*-value)	5L (*p*-value)	10L (*p*-value)
GE+Cont+Dist	0.692 (2.3E-09)	0.707 (5.9E-10)	0.738 (1.0E-10)	0.857 (9.1E-10)	1.074 (1.5E-06)
GE+Cont+Dist+Orien	**0.562 (*)**	**0.577 (*)**	**0.606 (*)**	**0.704 (*)**	**0.887 (*)**

Table B.3 Impact of the orientation restraints on β-protein folding. DeepFold results on the 38 β-proteins in the test set with and without orientation restraints in terms of the average TM-score/RMSD and the percent of correctly folded models (TM-scores ≥0.5) for the 221 benchmark proteins. The *p*-values were calculated using paired, two-sided Student's t-tests.

Method	**TM-score (*p*-value)**	**RMSD (*p*-value)**	**Correct Folds**
GE+Cont+Dist	0.590 (1.5E-04)	8.42 (3.4E-04)	60.5%
GE+Cont+Dist+Orien	**0.706 (*)**	**6.12 (*)**	**86.8%**

Table B.4 Modeling results for trRosetta using DeepPotential's spatial restraints vs DeepFold. The *p*-value for the mean TM-score was calculated using a paired, two-sided Student's t-tests.

Method	**Mean TM-score (*p*-value)**	**Correct Folds**
trRosetta+DeepPotential	0.735 (3.9E-09)	90.5%
DeepFold	**0.751 (*)**	**92.3%**

Table B.5 Modeling results for DeepFold and AlphaFold on the 31 CASP13 FM targets which the AlphaFold team submitted models for. The *p*-value for the mean TM-score was calculated using a paired, two-sided Student's t-tests.

Method	**Mean TM-score (*p*-value)**	**Correct Folds**
AlphaFold	0.589 (0.025)	64.5%
DeepFold	**0.636**	**80.6%**

APPENDIX C

Supplementary Figures for Chapter II

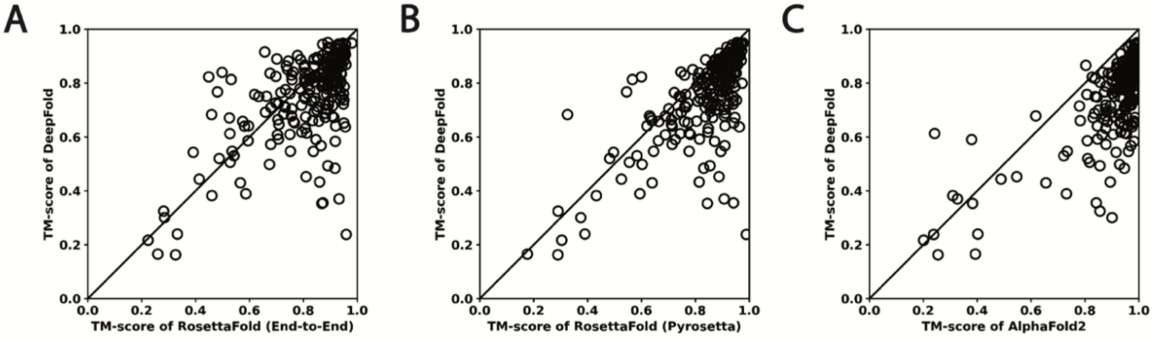


Figure C.1 Head-to-head TM-score comparisons between DeepFold using the restraints from DeepPotential (A-C) or the combined restraints from RosettaFold and DeepPotential (D-F) with other protein structure prediction methods on the 221 Hard benchmark proteins. A/D) RosettaFold (End-to-End); B/E) RosettaFold (Pyrosetta); C/F) AlphaFold2.

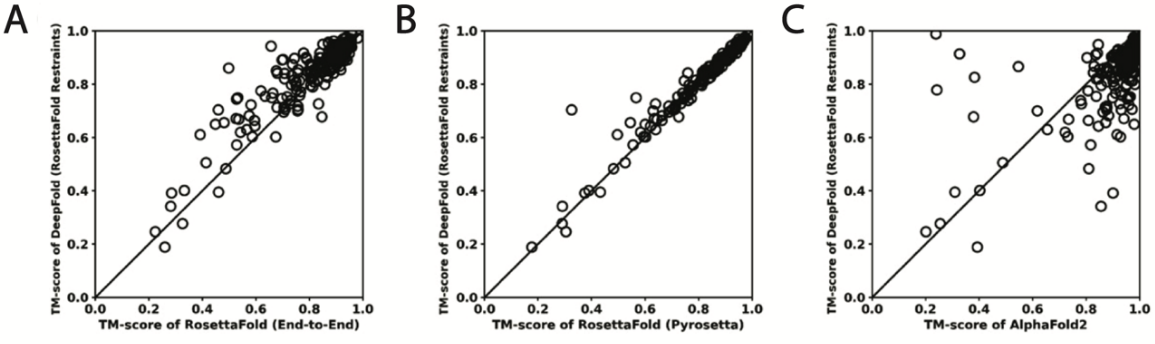


Figure C.2 Head-to-head TM-score comparisons between DeepFold using the restraints from DeepPotential (A-C) or the combined restraints from RosettaFold and DeepPotential (D-F) with other protein structure prediction methods on the 221 Hard benchmark proteins. A/D) RosettaFold (End-to-End); B/E) RosettaFold (Pyrosetta); C/F) AlphaFold2.

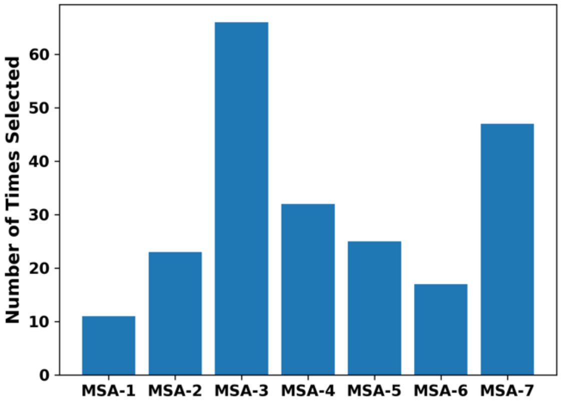


Figure C.3 Histogram distribution of the number of times each of the 7 MSAs were selected by DeepMSA2 for the 221 benchmark targets. The MSA numbers correspond to those depicted in Fig. 2.12.

APPENDIX D

Supplementary Texts for Chapter II

Text D.1 Calculation of the MSA Neff value.

In order to quantify the quality of an MSA, we define the number of effective sequences (Neff) as follows:

$$Neff = \frac{1}{\sqrt{L}} \sum\nolimits_{n=1}^{N} \frac{1}{1 + \sum_{m=1,m\neq n}^{N} I[S_{m,n} \geq 0.8]}$$

where L is the length of a query protein, N is the number of sequences in the MSA, Sm,n is the sequence identity between the m-th and n-th sequences, and I[ ] represents the Iverson bracket, which means $I\big[S_{m,n\geq 0.8}\big] = 1$ if $S_{m,n} \geq 0.8$ or 0 otherwise.

Text D.2 DeepFold energy function.

The energy function used to guide the DeepFold simulations is a combination of 10 energy terms:

$$E_{DeepFold} = (E_{C\beta dist} + E_{C\alpha dist} + E_{C\beta cont} + E_{C\alpha cont} + E_{\Omega} + E_{\theta} + E_{\varphi}) + (E_{hb} + E_{vdw} + E_{tor}) \qquad (D.1)$$

where $E_{C\beta dist}$, $E_{C\alpha dist}$, $E_{C\beta cont}$, and $E_{C\alpha cont}$ are the predicted Cβ–Cβ distances, Cα–Cα distances, Cβ–Cβ contacts, and Cα–Cα contacts generated by DeepPotential; E_{Ω}, E_{θ}, and E_{φ} are the predicted inter-residue orientations by DeepPotential as defined in Fig. 2.13; and E_{hb}, E_{vdw}, and E_{tor} are the hydrogen bonding, van der Waals and backbone torsion angle potentials. All of the energy terms are based on pairwise interactions between residues i and j in a protein molecule, with the exception of E_{tor}, which is a single-body potential. Thus, the cumulative terms are derived from the summation over all residue pairs *i* and *j* as follows:

$$E_{C\beta dist} = \sum_{i,j} w_1 E_{d_{ij}}(i,j) \tag{D.2}$$

$$E_{C\alpha dist} = \sum_{i,j} w_2 E_{d_{ij}}(i,j) \tag{D.3}$$

$$E_{C\beta cont} = \sum_{i,j} w_3 E_{con_{ij}}(i,j) \tag{D.4}$$

$$E_{C\alpha cont} = \sum_{i,j} w_3 E_{con_{ij}}(i,j) \tag{D.5}$$

$$E_{\Omega} = \sum_{i,j} w_4 E_{\Omega_{ij}}(i,j) \tag{D.6}$$

$$E_{\theta} = \sum_{i,j} w_5 E_{\theta_{ij}}(i,j) + \sum_{j,i} w_5 E_{\theta_{ji}}(j,i) \tag{D.7}$$

$$E_{\varphi} = \sum_{i,j} w_6 E_{\varphi_{ij}}(i,j) + \sum_{j,i} w_6 E_{\varphi_{ji}}(j,i) \tag{D.8}$$

$$E_{\mathrm{hb}} = \sum_{i,j} w_7 E_{hb_{ij}}(i,j) \tag{D.9}$$

$$E_{vdw} = \sum_{i,j} \sum_{ii,jj} w_8 E_{vdw}(i,j,ii,jj) \tag{D.10}$$

$$E_{\mathrm{tor}} = \sum_{i} w_9 E_{\phi_i}(i) + w_9 E_{\Psi_i}(i) \tag{D.11}$$

Note, the inter-residue θ and φ orientations are not symmetric, thus they must be summed over residues pairs i, j as well as the opposite direction j, i. Furthermore, the van der Waals potential also involves the interactions between each atom ii and jj from residues i and j. The detailed description of each energy term is described below.

$$E_{d_{ij}}(i,j) = \begin{cases} -\log\left(\frac{P(d_{ij}) + \epsilon}{P(d_{cut}) + \epsilon}\right), & d_{ij} < d_{cut} \\ 0, & d_{ij} \geq d_{cut} \end{cases} \qquad (D.12)$$

where d_{ij} is the distance between two Cβ atoms for the Cβ distance restraints or two Cα atoms for the Cα distance restraints from residues i and j, $P(d_{ij})$ is the predicted probability by DeepPotential associated with the distance d_{ij}, and $P(d_{cut})$ is the probability for the final distance bin which corresponds to a distance between 19.5Å and 20Å. The pseudo count $\epsilon = 1E - 4$ is used to avoid issues when $P(d_{cut})$ is small. Cubic spline interpolation is used to interpolate between the energy at the different distance bins in order to make the potential differentiable for L-BFGS optimization.

$$E_{con_{ij}}(\mathrm{i},\mathrm{j}) = \begin{cases} -U_{ij}, & d_{ij} < 8\text{Å} \\ -\frac{1}{2}U_{ij}\left[1 - sin\left(\frac{d_{ij}-(\frac{8+D}{2})}{d_b}\pi\right)\right], & 8\text{Å} \leq d_{ij} < D \\ \frac{1}{2}U_{ij}\left[1 + sin\left(\frac{d_{ij}-(\frac{D+80}{2})}{(80-D)}\pi\right)\right], & D \leq d_{ij} \leq 80\text{Å} \\ U_{ij}, & d_{ij} > 80\text{Å} \end{cases} \qquad \text{(D.13)}$$

where d_{ij} is the Cβ or Cα distance between the residue pair i and j. The depth of the potential, U_{ij}, is the predicted contact probability by DeepPotential. Overall, the potential is centered with a negative well at an 8 Å cutoff, with a strong force from 8 Å to D (=8 Å + d_b), followed by a weaker force from D to 80 Å, which is used to push the target residue pairs towards the well when they are far apart. Here, the gradient width (d_b) of the contact well is the only free parameter of the potential, which depends on the protein size and determines the convergence speed and satisfaction rate of the contact maps. d_b is typically narrow, e.g., 6 Å, when the length of the target is relatively small, e.g. < 100 residues. On the other hand, the well width increases to 12 Å when the length is >200 amino acids, since residue pairs from larger proteins are more difficult to draw together, a wider well is used to draw the candidate residue pairs that are further apart in distance close together. It is important that the contact potential is designed in a way that the potential curve is continuous and smooth (with $\partial E/\partial d = 0$) at all three transition points of $d_{ij} = 8$, D and 80 Å, so that the contact restraints can guide the gradient-based folding simulations.

$$E_{\Omega_{ij}}(i,j) = \{-\log\big(P(\Omega_{ij}) + \epsilon\big) \qquad (D.14)$$

$$E_{\theta_{ij}}(i,j) = \{-\log\big(P(\theta_{ij}) + \epsilon\big) \qquad (D.15)$$

$$E_{\theta_{ji}}(j,i) = \{-\log\big(P(\theta_{ji}) + \epsilon\big) \qquad (D.16)$$

$$E_{\varphi_{ij}}(i,j) = \{-\log\big(P(\varphi_{ij}) + \epsilon\big) \qquad (D.17)$$

$$E_{\varphi_{ji}}(j,i) = \{-\log\big(P(\varphi_{ji}) + \epsilon\big) \qquad (D.18)$$

where Ω_{ij}, θ_{ij}, and φ_{ij} are the inter-residue orientations predicted by DeepPotential between residues i and j defined in Fig. 2.13. Furthermore, given that θ and φ are not symmetric for a residue pair, θ_{ji}, and φ_{ji} are the inter-residue orientations between residues j and i. The pseudo count $\epsilon = 1E - 4$ is used to avoid issues when the predicted probability is small. Cubic spline interpolation is used to interpolate between the energy at the different orientation bins in order to make the potential differentiable for L-BFGS optimization.

$E_{hb}(i,j)$ was adapted from EvoEF (132) and is used to calculate the hydrogen-bonding interactions between potential hydrogen bond donor/acceptor pairs for atoms i and j, one of which should be a polar hydrogen. $E_{hb}(i,j)$ is a linear combination of three energy terms that depend on the hydrogen-acceptor distance (d_{ij}^{HA}), the angle between the donor atom, hydrogen and acceptor (θ_{ij}^{DHA}), and the angle between the hydrogen, acceptor and base atom (φ_{ij}^{HAB}):

$$E_{hb}(i,j) = w_{d_{HA}} E\left(d_{ij}^{HA}\right) + w_{\theta_{DHA}} E\left(\theta_{ij}^{DHA}\right) + w_{\varphi_{HAB}} E\left(\varphi_{ij}^{HAB}\right) \qquad (D.19)$$

where:

$$\begin{cases} E\left(d_{ij}^{HA}\right) = \begin{cases} -cos\left[\frac{\pi}{2}\left(d_{ij}^{HA} - 1.9\right)/\left(1.9 - d_{min}\right)\right], & d_{min} \leq d_{HA} \leq 1.9 \\ -0.5\,cos\left[\pi\left(d_{ij}^{HA} - 1.9\right)/\left(d_{max} - 1.9\right)\right] - 0.5, & 1.9\text{ Å} < d_{HA} \leq d_{max} \\ 0\,, & otherwise \end{cases} \\ E\left(\theta_{ij}^{DHA}\right) = -\,cos^{4}\left(\theta_{ij}^{DHA}\right), \;\; \theta_{ij}^{DHA} \geq 90° \\ E\left(\varphi_{ij}^{HAB}\right) = -\,cos^{4}\left(\varphi_{ij}^{HAB} - 150°\right), \;\; \varphi_{ij}^{HAB} \geq 80° \end{cases} \qquad (D.20)$$

$$E_{vdw}(i,j,ii,jj) = \begin{cases} (vdw(ii) + vdw(jj))^{2} - {d_{ij,ii,jj}}^{2}, if\ d_{ij,ii,jj} < vdw(ii) + vdw(jj) \\ 0,\ otherwise \end{cases} \qquad (D.21)$$

Here, $E_{vdw}(i,j,ii,jj)$ is the van der Waals energy between atoms ii and jj from residues i and j, respectively, where $vdw(ii)$ and $vdw(jj)$ are the van der Waals radii of atoms ii and jj and $d_{ij,ii,jj}$ is the distance between atoms ii and jj from residues i and j, respectively. The atoms ii/jj that are accounted for are the backbone atoms (N, Cα, C, and O) and the Cβ atoms/side-chain centers of mass.

$$E_{\phi_i}(i) = 1 - \cos\left(\phi_i - \phi_{i,pred}\right)\ and\ E_{\psi_i}(i) = 1 - \cos\left(\psi_i - \psi_{i,pred}\right) \qquad (D.22)$$

$E_{\phi_i}(i)$ and $E_{\psi_i}(i)$ are the energy for the backbone torsion angles, where ϕ_i and ψ_i are the phi/psi torsion angles at residue i and $\phi_{i,pred}$ and $\psi_{i,pred}$ are the predicted torsion angles by Anglor (238).

Overall, the DeepFold force field consists of 24 weighting parameters, where the weights given to each of the deep learning restraints were separated into short ($|i - j| > 1$ and $|i - j| \leq 11$, where i is the residue index for residue i and j is the residue index for residue j), medium ($|i - j| > 11$ and $|i - j| \leq 23$) and long-range ($|i - j| > 23$) weights, which were determined by maximizing the TM-score on the training set of 257 non-redundant, Hard threading targets collected from the PDB that shared <30% sequence identity to the test proteins. Briefly, all the weights were initialized to 0, then the weight for each individual energy term was increased one-at-a-time and the DeepFold folding simulation were run using the new weights. Following this initial optimization, the weights were carefully fine-tuned by adjusting their values using a grid-searching technique around the optimized values.

APPENDIX E

Supplementary Figures for Chapter III

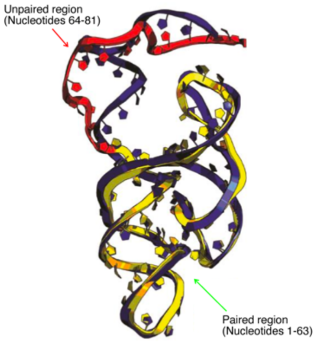


Figure E.1 Case study from Rfam RNA RF02678 where the DeepFoldRNA predicted model (blue cartoons) is superimposed on the experimentally solved structure (PDB ID: 6jq5, chain A, nucleotides 1-81). The unpaired region in the experimental structure is shown in red and the paired region in yellow.

APPENDIX F

Supplementary Texts for Chapter III

Text F.1 SimRNA procedure.

SimRNA was run using the following command:

```
<SimRNA_Directory>/SimRNA -s seq.fasta -c config.dat -S SecondaryStructure.txt
```

The default configuration file was used which runs 16,000,000 folding iterations, where the contents of the file are below:

```
NUMBER_OF_ITERATIONS 16000000
TRA_WRITE_IN_EVERY_N_ITERATIONS 16000

INIT_TEMP 1.35
FINAL_TEMP 0.90

BONDS_WEIGHT 1.0
ANGLES_WEIGHT 1.0
TORS_ANGLES_WEIGHT 0.0
ETA_THETA_WEIGHT 0.40
```

The final model was selected from the lowest energy decoy generated from each simulation.

Text F.2 FARFAR2 procedure.

FARFAR2 was run using the following command:

```
<Rosetta_bin>/rna_denovo.static.linuxgccrelease -fasta seq.fasta -native native.pdb -out:file:silent out.txt -nstruct 100 -minimize_rna true -fragment_homology_rmsd 1.2 -secstruct <Secondary Structure>
```

The default number of cycles were run for each simulation (10,000) and the final model was selected following clustering of the 100 generated structures using the default cluster radius of 3 Å and selecting the first cluster as the representative model.

Text F.3 DeepFoldRNA energy function.

The energy function used to guide the DeepFoldRNA simulations is a linear combination of 7 energy terms:

$$\mathrm{E}_{\mathrm{DeepFoldRNA}} = \mathrm{E}_{\mathrm{C4'dist}} + \mathrm{E}_{\mathrm{Ndist}} + \mathrm{E}_{\mathrm{Pdist}} + \mathrm{E}_{\Omega} + \mathrm{E}_{\lambda} + \mathrm{E}_{\mathrm{bb\eta}} + \mathrm{E}_{\mathrm{bb\theta}} \quad (F.1)$$

where $\mathrm{E}_{\mathrm{C4'dist}}$, $\mathrm{E}_{\mathrm{Ndist}}$, $\mathrm{E}_{\mathrm{Pdist}}$, E_{Ω}, E_{λ}, $\mathrm{E}_{\mathrm{bb\eta}}$, and $\mathrm{E}_{\mathrm{bb\theta}}$ are energy terms derived from the predicted C4'–C4' distances, N1/N9-N1/N9 distances, P-P distances, Ω orientations, λ orientations, backbone η torsions, and backbone θ torsions, respectively. All of the energy terms are based on pairwise interactions between residues *i* and *j* in an RNA molecule, with the exception of $\mathrm{E}_{\mathrm{bb\eta}}$ and $\mathrm{E}_{\mathrm{bb\theta}}$, which are single-body potentials. Thus, the cumulative terms are derived from the summation over all residue pairs i and j as follows:

$$E_{\mathrm{C4'dist}} = \sum_{i,j} E_{d_{ij}}(i,j) \quad (\mathrm{F.2})$$

$$E_{\mathrm{Ndist}} = \sum_{i,j} E_{d_{ij}}(i,j) \quad (\mathrm{F.3})$$

$$E_{\mathrm{Pdist}} = \sum_{i,j} E_{d_{ij}}(i,j) \quad (\mathrm{F.4})$$

$$E_{\Omega} = \sum_{i,j} E_{\Omega_{ij}}(i,j) \quad (\mathrm{F.5})$$

$$E_{\lambda} = \sum_{i,j} E_{\lambda_{ij}}(i,j) + \sum_{j,i} E_{\lambda_{ji}}(j,i) \quad (\mathrm{F.6})$$

$$E_{\mathrm{bb\eta}} = \sum_{i} E_{\mathrm{bb\eta_i}}(i) \quad (\mathrm{F.7})$$

$$E_{\mathrm{bb\theta}} = \sum_{i} E_{\mathrm{bb\theta_i}}(i) \quad (\mathrm{F.8})$$

Note, the inter-residue λ orientation is not symmetric, thus it must be summed over residues pairs (i, j) as well as the opposite direction (j, i).

The detailed description of each energy term is defined as:

$$E_{d_{ij}}(i,j) = \begin{cases} -\log\left(\frac{P(d_{ij}) + \epsilon}{P(d_{cut}) + \epsilon}\right), & d_{ij} < d_{cut} \\ 0, & d_{ij} \geq d_{cut} \end{cases} \quad (F.9)$$

where d_{ij} is the distance between two C4' atoms for the C4' distance restraints, two N1/N9 atoms for the N1/N9 distance restraints, or two P atoms for the P distance restraints from residues *i* and *j*, $P(d_{ij})$ is the predicted probability by DeepFoldRNA associated with the distance d_{ij}, and $P(d_{cut})$ is the probability for the final distance bin which corresponds to a distance between 39-40 Å. The pseudo count $\epsilon = 1E - 4$ is used to avoid issues when $P(d_{cut})$ is small. Cubic spline interpolation is used to interpolate between the energy at the different distance bins in order to make the potential differentiable for L-BFGS optimization.

$$E_{\Omega_{ij}}(i,j) = \{-\log(P(\Omega_{ij}) + \epsilon) \quad (F.10)$$

$$E_{\lambda_{ij}}(i,j) = \{-\log(P(\lambda_{ij}) + \epsilon) \quad (F.11)$$

$$E_{\lambda_{ji}}(j,i) = \{-\log\big(P\big(\lambda_{ji}\big) + \epsilon\big) \qquad (F.12)$$

where Ω_{ij} and λ_{ij} are the inter-residue orientations predicted by DeepFoldRNA between residues i and j defined in Figure S1. Furthermore, given that λ is not symmetric for a residue pair, λ_{ji} is the inter-residue orientation between residues j and i. The pseudo count $\epsilon = 1E - 4$ is used to avoid issues when the predicted probability is small. Cubic spline interpolation is used to interpolate between the energy at the different orientation bins in order to make the potential differentiable for L-BFGS optimization.

$$E_{bb\eta}(i) = \{-\log(P(\mathrm{bb\eta_i}) + \epsilon) \ \textit{and} \ E_{\mathrm{bb\theta}}(i) = \{-\log(P(\mathrm{bb\theta_i}) + \epsilon) \qquad (F.13)$$

where bbη_i and bbθ_i are the backbone pseudo-torsion angles predicted by DeepFoldRNA for residue i. The pseudo count $\epsilon = 1E - 4$ is used to avoid issues when the predicted probability is small. Cubic spline interpolation is used to interpolate between the energy at the different torsion bins in order to make the potential differentiable for L-BFGS optimization.

APPENDIX G

Supplementary Texts for Chapter IV

Text G.1 EvoDesign monomer evolutionary energy calculation.

The monomer evolutionary energy, $E_{evoMonomer}$, is calculated as the best match between the designed sequence and the scaffold structure using the Needleman-Wunsch dynamic programming (DP) algorithm (263). More specifically, a 2D DP matrix, $D(i,j)$, is defined where i and j are the positions along the designed and scaffold sequences, respectively. The value of $D(i,j)$ is equal to the ending value of the best path with the highest matching score towards the lattice (i,j). Here, a path in the matrix corresponds to an alignment between the designed and scaffold sequences. Thus, the $E_{evoMonomer}$ is the value at $D(L_1, L_2)$, where L_1 and L_2 are the lengths of the designed and scaffold sequences, respectively. Note, in EvoDesign L_1 and L_2 are equivalent. The DP procedure allows gaps in the alignment between the designed and scaffold sequences, depending on the alignment score.

Given a gap penalty scheme of $w(k) = g_o + (k-1)g_e$, where k is the gap length, and g_o and g_e are the gap opening and gap extension penalties, respectively, the initialization of the DP matrix can be written as

$$\begin{cases} D(0{,}0) = 0 \\ D(0,j) = j * w(j) \quad for\ 0 < j \ \leq L_2 \\ D(i,0) = i * w(i) \quad for\ 0 < i \ \leq L_1 \end{cases} \tag{G.1}$$

The remaining elements in the DP matrix are calculated by the recurrence equation:

$$D(i,j) = \max \begin{cases} D(i-1,j-1) + E_{match}(i,j) \\ \max\limits_{1 \le k \le i}[D(i-k,j) + w(k)] \\ \max\limits_{1 \le k \le j}[D(i,j-k) + w(k)] \end{cases} \qquad (G.2)$$

where the matching score between i and j is defined by

$$E_{match}(i,j) = M(j,aa_i) + \ w_1\Theta_{SS}(i,j) + w_2\Theta_{SA}(i,j) + w_3\Theta_{\phi}(i,j) + w_4\Theta_{\psi}(i,j) \qquad (G.3)$$

Here, aa_i is the amino acid for the i^{th} residue of the designed sequence and $M(j,aa_i)$ is the structural profile, represented by an $L_2 \times 20$ matrix, specifically, $M(j,aa_i) = \sum_{x=1}^{20} B(aa_i,x)H(j,x)$. Here, $B(aa_i,x)$ is the BLOSUM62 mutation score for mutating aa_i to amino acid x (264). Additionally, $H(j,x) = \sum_{m=1}^{f_x^j} h(m)$, where f_x^j is the frequency with which amino acid x appears at the j^{th} position of the multiple sequence alignment (MSA) that was constructed by TM-align (220) by structurally searching the scaffold against the PDB library. Lastly, $h(m)$ is the Henikoff weight of the m^{th} template sequence in the MSA. The higher (more positive) the value of $M(j,aa_i)$, the more favorable the mutation is between residue i of the designed sequence and residue j of the scaffold protein.

The terms in Eq. G.3 measure the local structural similarities between the designed sequence and the scaffold protein. The secondary structure (SS), solvent accessibility (SA), and backbone torsional angles (ϕ,ψ) for the designed sequence are predicted using the fast machine learning-based methods described previously (214), while those for the scaffold structure are assigned by DSSP (239). More specifically, these terms are defined as follows:

$$\begin{cases} \Theta_{SS}(i,j) = \begin{cases} 1, & \text{if } SS(i) = SS(j) \\ 0, & \text{else if } SS(i)\ or\ SS(j) \text{ is coil} \\ -1, & \text{otherwise} \end{cases} \\ \Theta_{SA}(i,j) = \begin{cases} 1, & \text{if } SA(i) = SA(j) \\ 0, & \text{else if } SA(i)\ or\ SA(j) \text{ is intermediate} \\ -1, & \text{otherwise} \end{cases} \\ \Theta_{\phi}(i,j) = \dfrac{-|\phi(i) - \phi(j)|}{180} \\ \Theta_{\psi}(i,j) = \dfrac{-|\psi(i) - \psi(j)|}{180} \end{cases} \qquad (G.4)$$

Here, SS is divided into three states: α-helix, β-strand or coil. Additionally, SA is categorized into three states: buried, intermediate or exposed based on its depth in the protein structure. The values for the weights w_1, w_2, w_3, and w_4 are 1.58, 2.45, 1.00, and 1.00, respectively, which are proportional to the relative accuracy of the SS, SA, and ϕ/ψ feature predictors for a set of 625 non-redundant training proteins (215).

Text G.2 Interface evolutionary energy pseudocount.

To offset the smaller size of the interface library, a pseudocount was introduced into the evolution-based interface potential by the BindProfX approach (216):

$$N_{pseudo}(aa_i, i) = N_{fix} + N_{gap} + N_{evo} = 5 + 15n_{gap}(i) + 5\sum_{x=1}^{20} \frac{N_{obs}(x,i)}{N_{tot}} M(x, aa_i) \quad (G.5)$$

where the first term, N_{fix}, is a constant parameter whose value is set to 25. The second term, N_{gap}, is a gap dependent pseudocount that is proportional to the number of gaps, $n_{gap}(i)$, at the i^{th} position of the iMSA. The final term is the evolutionary pseudocount, N_{evo}, which takes into account amino acids that are related to the wild-type and mutant residues in the interface alignment. $\frac{N_{obs}(x,i)}{N_{tot}}$ is the frequency with which an amino acid x appears at position i in the iMSA and $M(x, aa_i)$ is the interface probability transition matrix score for amino acid x mutating to residue aa_i.

Since the iMSA contains homologous sequences only from the PDB, its depth depends on the number of interface structural homologs detected. We previously found (216) that the average number of interface structural homologs was around five. This is much smaller than the size of the pseudocounts, indicating that the pseudocounts are quite important to calibrate the amino acid occurrence probabilities. The overall Pearson correlation coefficient (PCC) between experimental and predicted $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ values was 0.685 for the BindProfX benchmark on the overall dataset. However, for those targets with only one or two structurally similar interfaces, the PCC was 0.207 without pseudocounts, indicating that the amino acid occurrence probabilities were unreliable when there were too few interface homologs. With pseudocounts applied, the PCC increased from 0.207 to 0.323.

Text G.3 Dataset construction and EvoEF parameter optimization.

To compute the energy of a protein, EvoEF splits the total energy into the sum of three parts: the non-bonded atomic interactions within a residue ($E_{intraResidue}$), between different residues within the same chain ($E_{interResidueSameChain}$), and between different residues from different chains ($E_{interResidueDiffChain}$), i.e.,

$$\begin{aligned} E_{EvoEF} &= E_{intraResidue} + E_{interResidueSameChain} + E_{interResidueDiffChain} - E_{ref} \\ &= \{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{intraResidue} \\ &\quad + \{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{interResidueSameChain} \\ &\quad + \{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{interResidueDiffChain} \\ &\quad - E_{ref} \end{aligned} \qquad (G.6)$$

where E_{vdw}, E_{elec}, E_{HB} and E_{solv} are the same as defined in Eqs. (4.3-4.8) in Chapter 4. Overall, EvoEF uses eight energy terms each for $E_{interResidueSameChain}$ and $E_{interResidueDiffChain}$, and only six terms for $E_{intraResidue}$, as intra-residue E_{HBss} and E_{HBbb} do not exist. Thus, there are a total of 56 parameters that need to be optimized in EvoEF, including 8 weights for $E_{intraResidue}$, 14 weights for $E_{interResidueSameChain}$, 14 weights for $E_{interResidueDiffChain}$, and 20 amino acid reference energies.

G.3.1 Dataset construction.

We used two types of experimental data, based on the mutation-induced protein stability and binding affinity changes, to train and test EvoEF. The mutation-induced protein stability change data were collected from the FoldX (228) and STRUM (44) datasets, which contain 1,056 and 3,421 mutants, respectively. After filtering out the duplicated mutants in identical structures, a total of 3,989 non-redundant mutants from 210 proteins were retained, where 3,978 were single mutations and 11 were multiple mutations. Half of the 3989 mutants were randomly selected as the training set (with 1995 mutants) and the other half as the testing set (with 1994 mutants).

Here, we note that the FoldX dataset has an overrepresentation of mutations from larger residues to smaller ones. Out of the 1,056 data samples, 1,015 are from larger-sized amino acids to smaller ones, while only 41 are from smaller to larger-sized amino acids. This trend is much less obvious in the STRUM dataset, where 2,568 out of 3,421 mutation samples are from larger to smaller amino acids and 853 are from smaller to larger amino acids. The bias present in the FoldX dataset results in overestimation of the mutation correlations. For example, the Pearson correlation coefficient between the predicted and experimental stability change data for the FoldX potential on the FoldX dataset is 0.688, which is reduced to 0.446 for the STRUM dataset.

For the second set of benchmark data, experimental mutation-induced binding affinity changes were collected from the SKEMPI v2.0 database (229), which contains 7,085 mutation entries in total. The training and test datasets were constructed as follows. First, we discarded mutants whose corresponding structures contained three or more chains. Second, we removed mutants with non-interface residues. Here, an interface residue is defined as a residue that has at least one heavy atom within 5.0 Å of the other chain in a protein complex. When there were multiple entries for the same mutant, the average $\Delta\Delta G_{binding}^{WT\to mut}$ value was calculated. After filtering the dataset, a total of 2,204 mutants from 177 protein-protein interfaces were retained. Again, half the 2,204 mutants were randomly selected as the training set (with 1102 mutants) and the other half as the test set (with 1102 mutants).

In order to predict the binding affinity and stability change upon mutation, the native structures were minimized, and the mutant models were generated using the following steps (the information for each command can be found in Text G.5):

Step 1: For $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ predictions, we extracted the single target chain or the two target amino acid chains, respectively, from the PDB file of the crystal structure and discarded water molecules and ligands that were not amino acids.

Step 2: We optimized the structure of the wild-type protein/complex using EvoEF's "RepairStructure" command as follows:

```
./EvoEF --command=RepairStructure --pdb=wildtype.pdb
```

Following this command, the minimized wild-type protein/complex was output into a file named 'wildtype_Repair.pdb' and this minimized model was used as the initial structure to build the mutant model.

Step 3: We built a structural model of the mutant protein/complex using EvoEF's "BuildModel" command as follows:

```
./EvoEF --command=BuildMutant --pdb=wildtype_Repair.pdb --mutant-file=individual_list.txt
```

The file "individual_list.txt" contained the list of mutation(s). Following this command, a new file "wildtype_Repair_Mutant_1.pdb" was generated, which contained the modelled mutant structure.

Step 4: We computed the stability of the wild-type and mutant proteins using EvoEF's "ComputeStability" command as follows:

```
./EvoEF --command=ComputeStability --pdb=wildtype_Repair.pdb
./EvoEF --command=ComputeStability --pdb=wildtype_Repair_Mutant_1.pdb
```

Or

We computed the binding affinity of the wild-type and mutant complexes using EvoEF's "ComputeBinding" command as follows:

```
./EvoEF --command=ComputeBinding --pdb=wildtype_Repair.pdb
./EvoEF --command=ComputeBinding --pdb=wildtype_Repair_Mutant_1.pdb
```

The above steps were used to minimize/construct the models and predict either the stability or binding affinity during EvoEF's training/testing. However, to benchmark EvoEF against FoldX and to avoid potential bias in the scoring, for the FoldX tests, we minimized the structures using FoldX. The following steps were used to build the structural models and predict the stability/binding affinity energies for FoldX:

Step 1: For $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ predictions, we extracted the single target chain or the two target amino acid chains, respectively, from the PDB file of the crystal structure and discarded water molecules and ligands that were not amino acids.

Step 2: We optimized the structure of the wild-type protein/complex using FoldX's "RepairPDB" command:

```
./foldx --command=RepairPDB --pdb=wildtype.pdb
```

After this step, the minimized wild-type protein/complex was output into a file named 'wildtype_Repair.pdb' and this minimized model was used as the initial model to build the mutant model.

Step 3: We built a structural model of the mutant protein using FoldX's "BuildModel" command:

```
./foldx --command=BuildModel --pdb=wildtype_Repair.pdb --mutant-file=individual_list.txt
```

Here, “individual_list.txt” was a text file that contained the specified mutation(s). After this step, two files “WT_wildtype_Repair_1.pdb” and “wildtype_Repair_1.pdb” were generated. The former file was the wild-type structure with additional structural optimization, while the latter one was the mutant structure. Normally, “WT_wildtype_Repair_1.pdb” was the same as “wildtype_Repair.pdb”, and if not, their difference were quite small.

Step 4: We computed the stability of the wild-type and mutant proteins using FoldX’s “Stability” command:

```
./foldx --command=Stability --pdb=WT_wildtype_Repair_1.pdb
./foldx --command=Stability --pdb=wildtype_Repair _1.pdb
```

Or

We computed the binding affinity of the wild-type and mutant complexes using FoldX’s “AnalyseComplex” command:

```
./foldx --command=AnalyseComplex --pdb=WT_wildtype_Repair_1.pdb
./foldx --command=AnalyseComplex --pdb=wildtype_Repair_1.pdb
```

The stability and binding affinity change datasets, as well as the predicted ΔΔGs by FoldX and EvoEF can be found at: https://zhanglab.ccmb.med.umich.edu/EvoDesign/EvoEFBenchmark.tar.gz.

G.3.2 Optimization of reference energies and weights for $E_{interResidueSameChain}$ and $E_{intraResidue}$.

The amino acid reference energies and the weighting factors for $E_{intraResidue}$ and $E_{interResidueSameChain}$ were determined based on the stability change data ($\Delta\Delta G_{stability}^{WT\rightarrow mut}$) of monomeric proteins upon mutation. The protein stability change due to mutation is computed by

$$\Delta\Delta G_{stability}^{WT\rightarrow mut} = \Delta G_{stability}^{mut} - \Delta G_{stability}^{WT} = E_{EvoEF}^{mut} - E_{EvoEF}^{WT} \qquad (G.7)$$

where the wild-type and mutant structural models are required to compute the physical energies. To this end, we first performed local energy minimization on the native crystal structures using the EvoEF energy minimizer and then built mutant models based on the minimized wild-type structures using the steps described above. For doing so, EvoEF first scans the wild-type structure in the order of amino acid occurrence and then optimizes the amino acid side-chains one-by-one. Several minimization cycles can be performed for the sake of convergence, but the default number of cycles is set to one. Based on our test, there's not a large difference in the minimized structures when we set the number of cycles to two or more. To remove the possible steric clashes during the minimization procedure, EvoEF searches alternative rotameric conformations from a backbone-independent rotamer library obtained from Xiang and Honig (265). The rotamer library contains 984 rotamers for the 20 amino acid types, and 1,007 rotamers if two tautomers of histidine are considered. In the library, the hydroxyl groups of serine, threonine and tyrosine are rotated to expand their rotamers by six, six, and two folds, respectively. Asparagine, histidine and glutamine are also flipped to construct better hydrogen bonding networks during energy minimization. The details of energy minimization, model building and $\Delta\Delta G$ computation can be found in Text G.5.

Finally, the reference energies and parameters for $E_{intraResidue}$ and $E_{interResidueSameChain}$ were optimized by minimizing the objective function $F = \sum_i \left(\Delta\Delta G_{i,stability,pred}^{WT\rightarrow mut} - \Delta\Delta G_{i,stability,exp}^{WT\rightarrow mut}\right)^2$ over a set of experimental protein stability change data, where $\Delta\Delta G_{i,stability,pred}^{WT\rightarrow mut}$ and $\Delta\Delta G_{i,stability,exp}^{WT\rightarrow mut}$ were the predicted and experimental data for the i^{th} mutation in the dataset. More specifically, the objective function can be written as:

$$\begin{aligned} F &= \sum_i \left(\Delta\Delta G_{i,stability,pred}^{WT\rightarrow mut} - \Delta\Delta G_{i,stability,exp}^{WT\rightarrow mut}\right)^2 \\ &= \sum_i \left[\left(\sum_j \omega_j \Delta\Delta G_{i,stability,pred}^{WT\rightarrow mut}(j) + E_{ref}^{WT} - E_{ref}^{mut}\right) - \Delta\Delta G_{i,stability,exp}^{WT\rightarrow mut}\right]^2 \end{aligned} \qquad (G.8)$$

where $\Delta\Delta G_{i,stability,pred}^{WT\rightarrow mut}(j)$ was the EvoEF predicted stability change upon mutation for the j^{th} energy term, not considering the reference energy. $E_{ref}(WT)$ was the summed reference energy for the wild-type sequence and $E_{ref}(mut)$ was the summed reference energy for the mutant sequence. This is essentially a least squares optimization problem, which can be easily solved using simple algorithms such as least squares fitting, gradient descent and conjugated gradient methods. However, we found that the optimal weights for some terms decided by these methods could be negative and theoretically meaningless. Therefore, we implemented a Metropolis Monte Carlo procedure to re-optimize the parameters. During the procedure, the movement consisted of random changes to the parameters while the weights were restricted to be greater than or equal to zero. Parameter changes were accepted and rejected based on the Metropolis criterion, where *F* was the energy. The final reference energies and weights were chosen from the parameter set with the lowest *F* value over the training set.

G.3.3 Optimization of weights for $E_{interResidueDiffChain}$.

One of the major goals of the work described in Chapter 4 was to extend the EvoDesign pipeline to design protein-protein interactions. To achieve the best performance in computing the physical interactions in protein-protein interfaces, we used experimental binding affinity change ($\Delta\Delta G_{binding}^{WT\rightarrow mut}$) data to train the weights for $E_{interResidueDiffChain}$. In EvoEF, the binding energy of a protein complex for scaffold A and its binding partner B is computed by

$$\Delta G_{binding} = E_{AB} - E_A - E_B \qquad (G.9)$$

where E_{AB}, E_A and E_B are the stability scores for the complex and component monomers, respectively. The binding free energy change due to mutation is then written as

$$\Delta\Delta G_{binding}^{WT\rightarrow mut} = \Delta G_{binding}^{mut} - \Delta G_{binding}^{WT} \qquad (G.10)$$

The parameters for $E_{interResidueDiffChain}$ were decided by minimizing the objective function $\sum_i \left(\Delta\Delta G_{i,binding,pred}^{WT\rightarrow mut} - \Delta\Delta G_{i,binding,exp}^{WT\rightarrow mut}\right)^2$ over the training set of experimental binding affinity change data, where $\Delta\Delta G_{i,binding,pred}^{WT\rightarrow mut}$ and $\Delta\Delta G_{i,binding,exp}^{WT\rightarrow mut}$ were the predicted and experimental data, respectively, for the i^{th} mutation in the SKEMPI training set described above. The same Metropolis Monte Carlo procedure was used to decide the parameters for $E_{interResidueDiffChain}$ as was used to train the parameters for $E_{interResidueSameChain}$ and $E_{intraResidue}$ as well as the reference energies. During the Monte Carlo search, the 42 previously optimized parameters were fixed.

Text G.4 EvoEF decoy discrimination.

In order to further validate EvoEF, we assessed its ability to discriminate native structures from decoy structures for the 200 non-redundant monomeric proteins in the 3DRobot Decoy Set (245). For each protein, 300 decoys were generated by 3DRobot. The root mean squared deviations (RMSDs) of the structural decoys to the native ranged from 0 to 12 Å. We did two types of decoy discrimination tests: (1) discriminating the native structures from decoys according to the folding stability energy, and (2) discriminating near-native decoy structures (low RMSD decoys) from those with high RMSDs.

In the first test, EvoEF correctly ranks the native protein as the lowest energy for each of the 200 individual decoy sets, while FoldX does so in 198 cases. We also computed the Z-score of the native structure in each decoy set:

$$Z_{native} = \frac{\langle E\rangle - E_{native}}{\delta E} \qquad (G.11)$$

where E_{native} is the energy of the native structure, and $\langle E\rangle$ and δE are the average and standard deviation of the energy function for all the structures in the decoy set. For EvoEF, the average Z-score was 4.434 with value ranging from 2.25 to 8.09 for the 200 structures, while the average Z-score for FoldX was 4.484 with value ranging from 2.41 to 7.60. Since the Z-score was >1 for all the cases, both FoldX and EvoEF were able to discriminate the native from non-native decoys with a sufficient gap for all the decoy sets, although FoldX had a slightly higher average Z-score.

For the second test, we computed the Z-score of the near-native decoy structures, i.e., those with low RMSDs:

$$Z_{nnative} = \frac{\langle E\rangle_h - \langle E\rangle_l}{\delta E_h} \qquad (G.12)$$

where $\langle E\rangle_l$ is the average energy for the 10% of decoys that have the lowest RMSD; $\langle E\rangle_h$ and δE_h are the average and standard deviation of the energy function for all the rest of the structures in the decoy set. Since both low and high RMSD decoys are generated *in silico* and thus have similar local structural errors, it is much harder to recognize the near-native structures than to recognize the native structure which was determined experimentally and usually has idealized local structural features and side-chain packing. The average $Z_{nnative}$ for EvoEF was 1.959 with values ranging from 0.32 to 3.57, while that for FoldX was 1.844 with values ranging from 0.40 to 2.83. In 198 cases, EvoEF has a $Z_{nnative} > 1$, while FoldX has a $Z_{nnative} > 1$ in 193 of the cases. These data suggest that EvoEF has a relatively better ability to recognize near-native structures from high RMSD structural decoys.

Here, the decoys datasets were taken directly from the work of Deng et al. (245), which can be downloaded at: https://zhanggroup.org/3DRobot/decoys/. The decoy recognition data for EvoEF and FoldX can be found at: https://zhanglab.ccmb.med.umich.edu/EvoDesign/EvoEFBenchmark.tar.gz.

Text G.5 Commands in EvoEF.

We have implemented several commands such as “RepairStructure”, “BuildMutant”, “ComputeStability”, “ComputeBinding”, and “OptimizeHydrogen”, to make it easy to use EvoEF’s functions. Generally, these commands are performed using the following syntax:

```
 EvoEF --command=commandName --pdb=your.pdb [other options]
```

Here, we describe the details of each of these commands.

G.5.1 Energy minimization.

Energy minimization in EvoEF is performed using the command “RepairStructure”. Usually, the user-provided structural model or even the crystal structure can have steric clashes or bad hydrogen bonding networks. Moreover, sometimes side-chain atoms can be missing from the structural model. Therefore, it is important to fix the structure and do energy minimization to optimize the rotameric side-chain conformations for the clashed amino acids. Essentially, the global optimization of the amino acid side-chain conformations requires complete repacking of the side-chains, but this is not trivial. Instead of doing full side-chain repacking, EvoEF does fast local optimization of the initial model, either a crystal structure or a model predicted by structure modeling software, to remove steric clashes as much as possible. To do so, EvoEF first scans the user-input structure in the order of amino acid occurrence and then optimizes the amino acid side-chains one-by-one. To remove the possible steric clashes in the user-provided structural model, EvoEF searches rotameric conformations from a backbone-independent rotamer library obtained from the work of Xiang and Honig (265). The rotamer library contains 984 rotamers for 20 amino acid types, and 1,007 rotamers if two tautomers of histidine are considered. In the library, the hydroxyl groups of serine, threonine and tyrosine are rotated to expand their rotamers by six, six, and two folds, respectively. The asparagine, histidine and glutamine amino acids are also flipped for better hydrogen bonding evaluation. Several cycles of energy minimization can be performed for the sake of convergence and the best minimization results, but the default number of cycles is set to one. Based on our benchmarking, the difference between one or two or more minimization cycles is not that significant. The syntax to do energy minimization in EvoEF is:

```
EvoEF --command=RepairStructure --pdb=model.pdb
```

Successful execution of this command will generate a new structure file named "*model_Repair.pdb*". In the minimized model, the optimized polar hydrogen coordinates are also shown.

G.5.2 Model builder.

To compute the protein stability and binding affinity changes due to mutation, we need the experimental structure and the mutant model. Here, the experimental structure should be minimized as mentioned in the above section. We build a mutant model starting from the minimized wild-type structure and mutate the amino acids at the specified positions one-by-one. During the mutation process, the amino acid side-chain conformations within 6 Å of each mutated position are repacked to alleviate possible steric clashes and optimize the local energies. The rotameric conformations for repacking and mutation are also taken from the above Xiang and Honig rotamer library (265). The mutated structure is first built, then three cycles of local energy minimization are performed. The syntax to build mutant models in EvoEF is:

```
EvoEF --command=BuildMutant --pdb=model_Repair.pdb --mutant-file=individual_list.txt
```

where "*model_Repair.pdb*" is the minimized initial model, and "*individual_list.txt*" is a text file that specifies the desired mutation(s). In "*individual_list.txt*", the mutations must be presented in the following format:

```
CA171A,DB180E;
```

Each mutation is written in one line ending with ";", and multiple mutants are divided by ",". Note that there are no gaps/spaces between single mutations. For each single mutation, the first letter is the native amino acid, the second is the identifier for the chain that the amino acid appears on, the number is the amino acid's position in the chain, and the last letter is the mutant amino acid. Running the command successfully should generate a new structure file named "*model_Repair_Mutant_1.pdb*". In the mutant model, the optimized polar hydrogen coordinates are also shown.

G.5.3 Energy computation.

In EvoEF, the protein stability energy can be calculated using the following command:

```
EvoEF --command=ComputeStability --pdb=your.pdb
```

Furthermore, binding affinity for protein-protein complexes can be calculated using the command:

```
EvoEF --command=ComputeBinding --pdb=complex.pdb
```

The energies for each term and the total energy will be output if the command is run successfully.

G.5.4 Other commands

In the initial protein structures, such as the crystal structure or models obtained by different structure modelling software, polar hydrogens are usually not provided. However, the positions of polar hydrogens are important to model and calculate hydrogen bonding energy, which is crucial for the structural specificity that underlies protein folding, function, and interactions. Although backbone or side-chain polar hydrogens of some amino acid types can be determined by standard topologies from force fields such as CHARMM19 (224) and AMBER (266), the hydroxyl groups of serine, threonine and tyrosine are rotatable and the hydrogen positions cannot be decided by the topologies. In EvoEF, we implemented another command "OptimizeHydrogen" to find the hydrogen positions that optimize the hydrogen bonding network. Specifically, we build the rotamers for serine, threonine and tyrosine using their native conformations and expand the number of rotamers considered by rotating the hydroxyl groups.

APPENDIX H

Supplementary Tables for Chapter V

Table H.1 Results of AlphaFold2 modeling using different MSA generation methods for the 354 native protein structures. P-values were calculated using paired, two-sided Student's t-tests between the results by DeepMSA and the other approaches. In the table, the 'DeepMSA MSA' option refers to the results obtained by AlphaFold2 starting from the MSAs identified by searching the original native sequences using the DeepMSA program, the 'Designed MSA' option refers to the results obtained by AlphaFold2 when starting from the alignment of 100 designed sequences by EvoEF2 or RosettaFixBB, and the 'Single Sequence' option refers to the results for AlphaFold2 modeling starting from the single lowest energy designed sequence produced by EvoEF2 or RosettaFixBB.

AlphaFold2 Input	**TM-score (*p*-value)**	**RMSD Å (*p*-value)**	**#TM-score ≥ 0.5[a]**
Native sequences			
DeepMSA MSA	0.913 (*)	1.99 (*)	350
Sequences designed by EvoEF2			
Designed MSA	0.852 (3.8E-13)	2.48 (1.8E-02)	345
Single Sequence	0.506 (7.7E-113)	12.45 (3.4E-91)	179
Sequences designed by RosettaFixBB			
Designed MSA	0.837 (2.5E-18)	2.72 (3.3E-04)	344
Single Sequence	0.482 (1.3E-120)	12.08 (5.4E-94)	161

[a] This column indicates the number of AlphaFold2 models with correct global folds (i.e., TM-score ≥0.5).

Table H.2 Local structure characteristics of the designed folds by FoldDesign. The table illustrates the overall Molprobity scores (MP-score) and additional structure quality metrics output by the Molprobity program for the 354 native structures (Native) as well as the 354 FoldDesign scaffolds (All Designs), the 79 novel designs (Novel Designs), and the 275 designs with native fold analogs (Analogous Designs).

Structures	MP- Score	Rama Outliers (%)	Rama Favorable (%)	Rotamer Outliers (%)	Clash Score	RMS Bonds	RMS Angles
Native	1.19	1.19	93.95	5.53	0.00	0.01	1.48
All Designs	1.59	0.46	96.91	0.05	0.00	0.04	3.43
Novel Designs	1.66	0.42	96.58	0.06	0.00	0.04	3.43
Analogous Designs	1.57	0.47	97.00	0.04	0.00	0.04	3.43

Table H.3 Empirically observed acceptance probabilities for swaps between adjacent replicas during the FoldDesign simulations for the 354 test proteins.

Replica Number	Fraction of Accepted Swaps
1	0.771
2	0.767
3	0.759
4	0.742
5	0.728
6	0.716
7	0.695
8	0.686
9	0.680
10	0.687
11	0.690
12	0.697
13	0.708
14	0.714
15	0.718
16	0.723

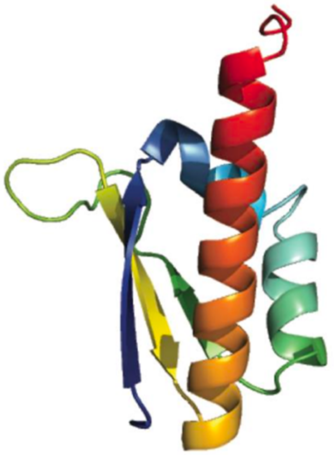
17	0.733
18	0.735
19	0.739
20	0.749
21	0.754
22	0.758
23	0.762
24	0.769
25	0.770
26	0.776
27	0.778
28	0.780
29	0.778
30	0.776
31	0.777
32	0.773
33	0.771
34	0.762
35	0.755
36	0.737
37	0.720
38	0.689
39	0.643

Table H.4 Feature values μ_{kl}/δ_{kl} for each hydrogen bonding restraint type, Tk, in Eq. J.4. The features are presented as averages/standard deviations.

Restraint Type	**Secondary Structure**	$\boldsymbol{f_1}$: $\boldsymbol{D(O_i, H_j)}$ **(Å)**	$\boldsymbol{f_2}$: $\boldsymbol{A(C_i, O_i, H_j)}$ **(degrees)**	$\boldsymbol{f_3}$: $\boldsymbol{A(C_i, O_i, H_j)}$ **(degrees)**	$\boldsymbol{f_4}$: $\boldsymbol{T(C_i, O_i, H_j, N_j)}$ **(degrees)**
T_1	Helix, $j = i + 4$	2.00/0.53	147/10.58	159/11.25	160/25.36
T_2	Helix, $j = i + 3$	2.85/0.32	89/7.70	111/8.98	-160/7.93
T_3	Parallel Strand	2.00/0.30	155/11.77	164/11.29	180/68.96
T_4	Antiparallel Strand	2.00/0.26	151/12.38	163/11.02	-168/69.17

APPENDIX I

Supplementary Figures for Chapter V



FoldDesign Energy: -145.5 k_BT

Figure I.1 Structure and FoldDesign energy for the native 1ec6A fold.

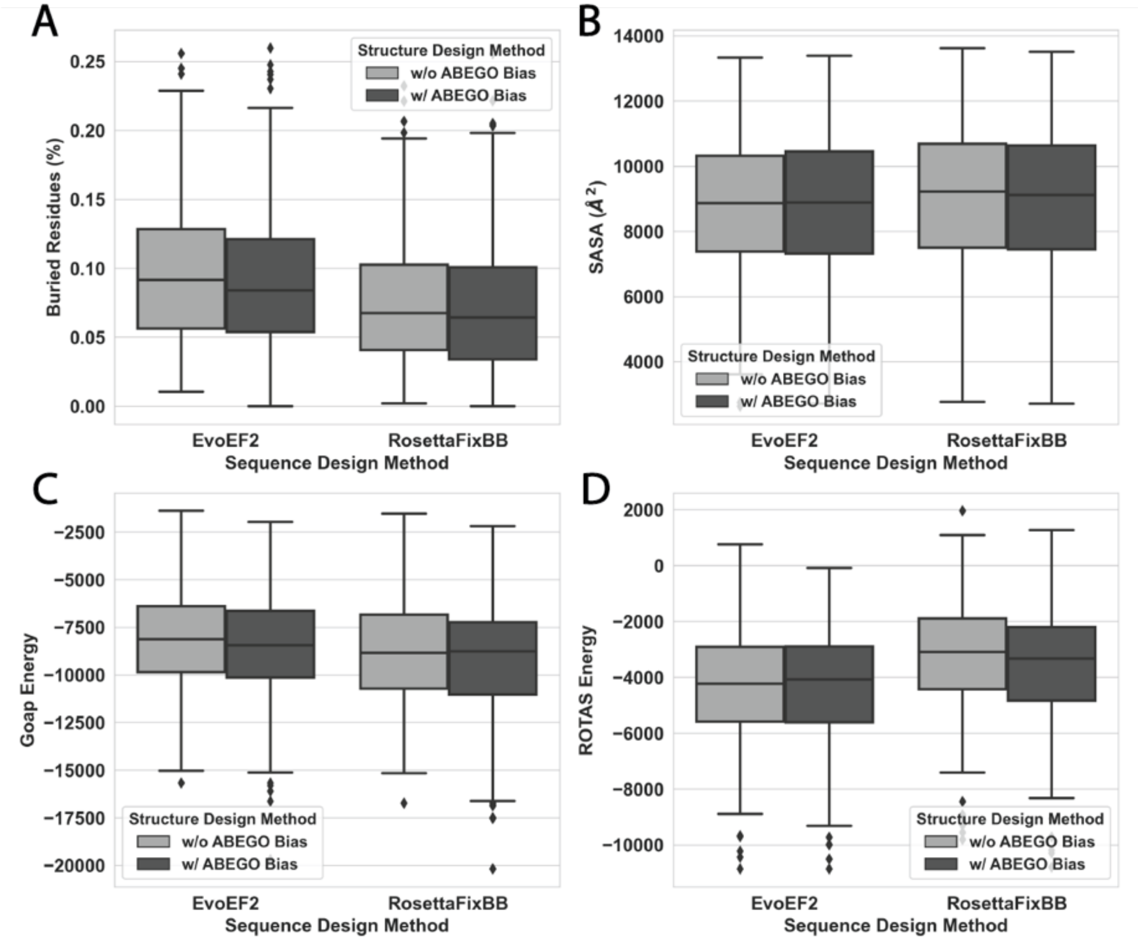


Figure I.2 Comparison of the physical characteristics and energies for the designed folds by Rosetta with and without ABEGO bias on the 354 test proteins, where the sequence for each scaffold was designed by EvoEF2 and RosettaFixBB. A) Proportion of buried residues is plotted for each design, where a buried residue was defined as having a relevant solvent accessible surface area <5%. B) Solvent accessible surface area (SASA) for each design. C-D) Energies for each design calculated by GOAP and ROTAS.

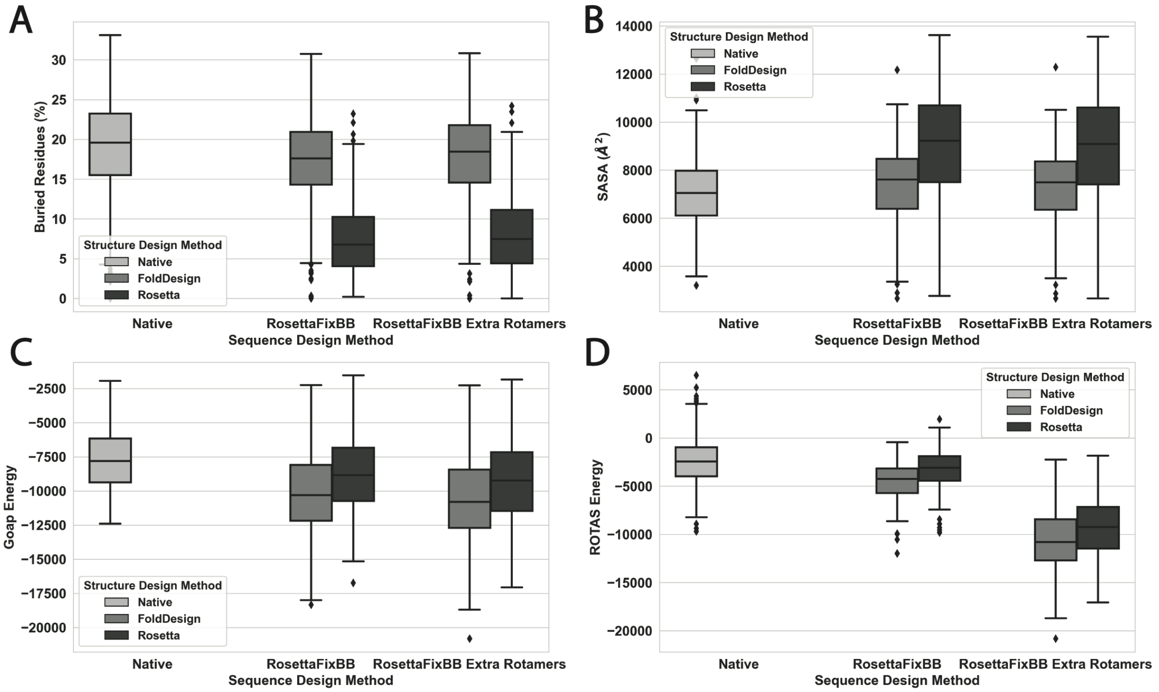


Figure I.3 Comparison of the physical characteristics and energies for the designed folds by FoldDesign and Rosetta on the 354 test proteins, where the sequence for each scaffold was designed by RosettaFixBB with (RosettaFixBB Extra Rotamers) or without (RosettaFixBB) sub-rotamer sampling for the χ_1 and χ_2 angles. The native designation represents the proteins from which the secondary structures of the designed folds were derived. A) Proportion of buried residues is plotted for each protein, where a buried residue was defined as having a relevant solvent accessible surface area <5%. B) Solvent accessible surface area (SASA) for each protein. C-D) Energies for each protein calculated by GOAP and ROTAS.

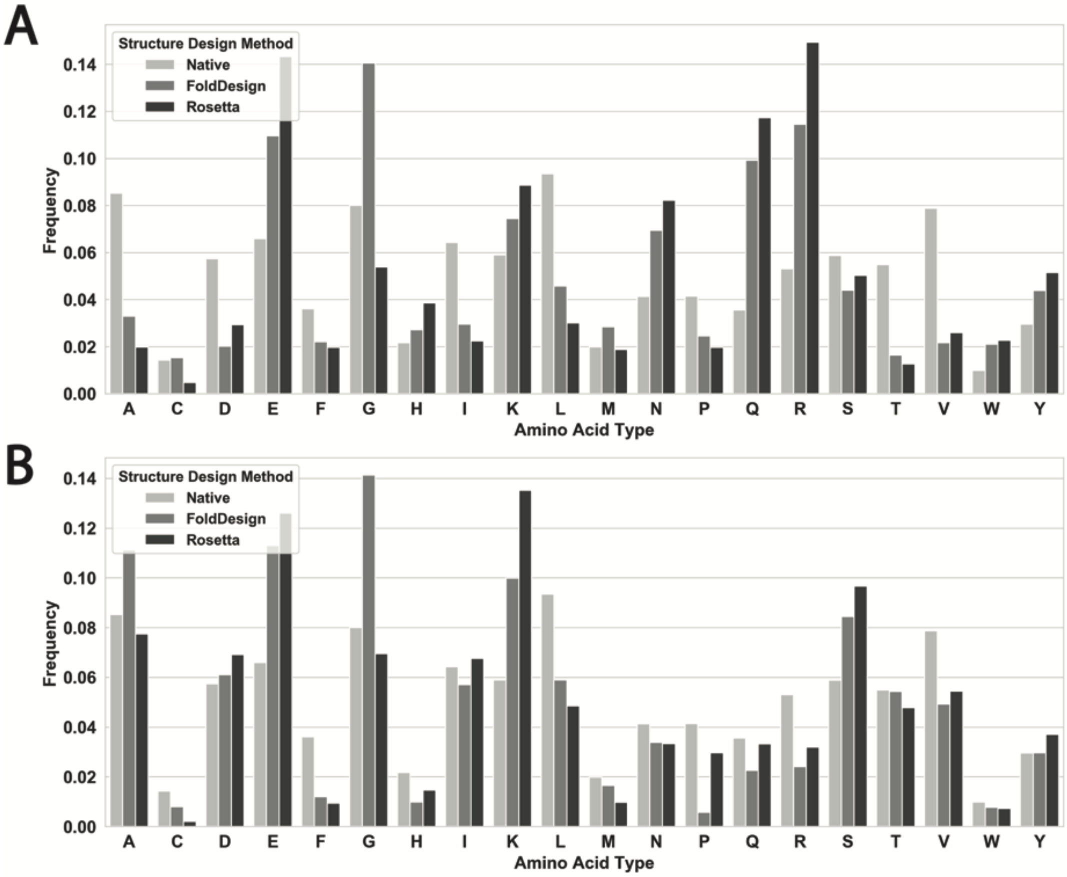


Figure I.4 Comparison of the amino acid distributions for the native proteins as well as the FoldDesign and Rosetta scaffolds whose sequences were designed by EvoEF2 (A) and RosettaFixBB (B), respectively. The native designation represents the 354 proteins from which the secondary structures of the designed folds were derived.

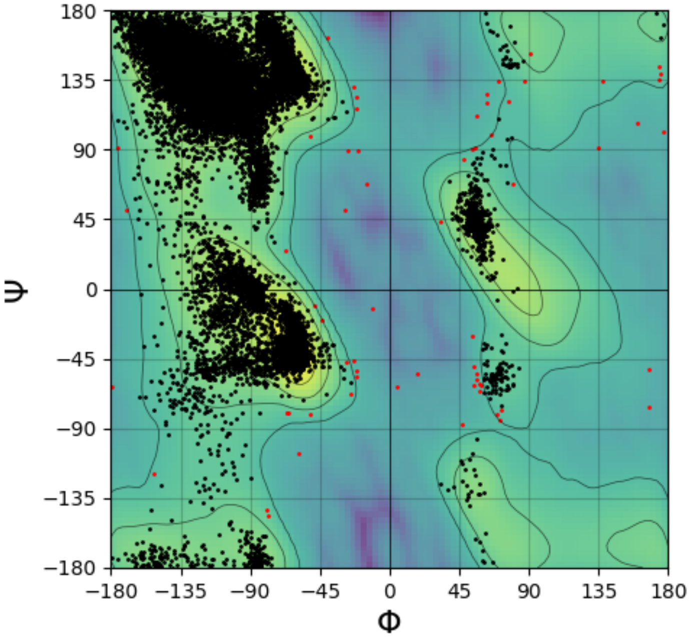


Figure I.5 Ramachandran plot derived from the 354 FoldDesign scaffolds. Favored/allowable torsion angles are plotted using black circles and outliers are plotted using red circles.

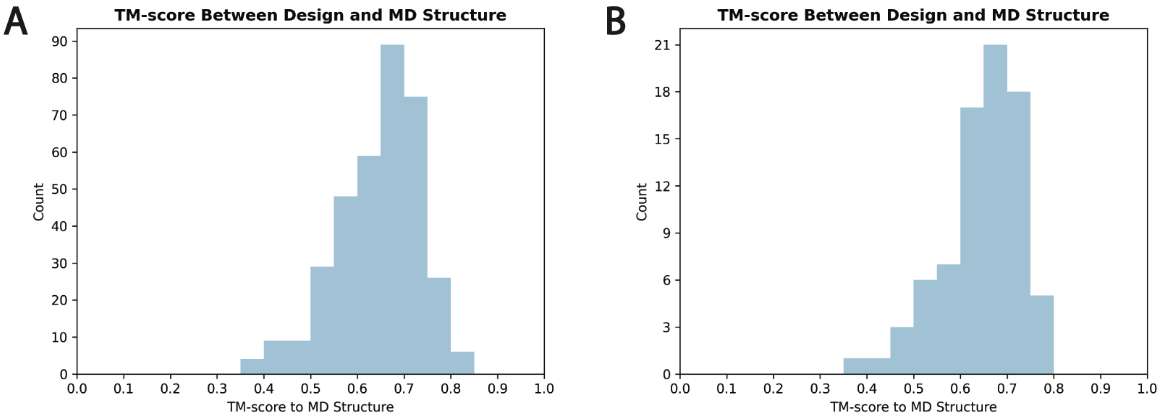


Figure I.6 Assessment of the stability of the novel folds generated by FoldDesign. A) TM-score distribution between the FoldDesign scaffolds and their final MD structures on the 354 test topologies. B) TM-score distribution between the 79 novel FoldDesign structures and their final MD structures.

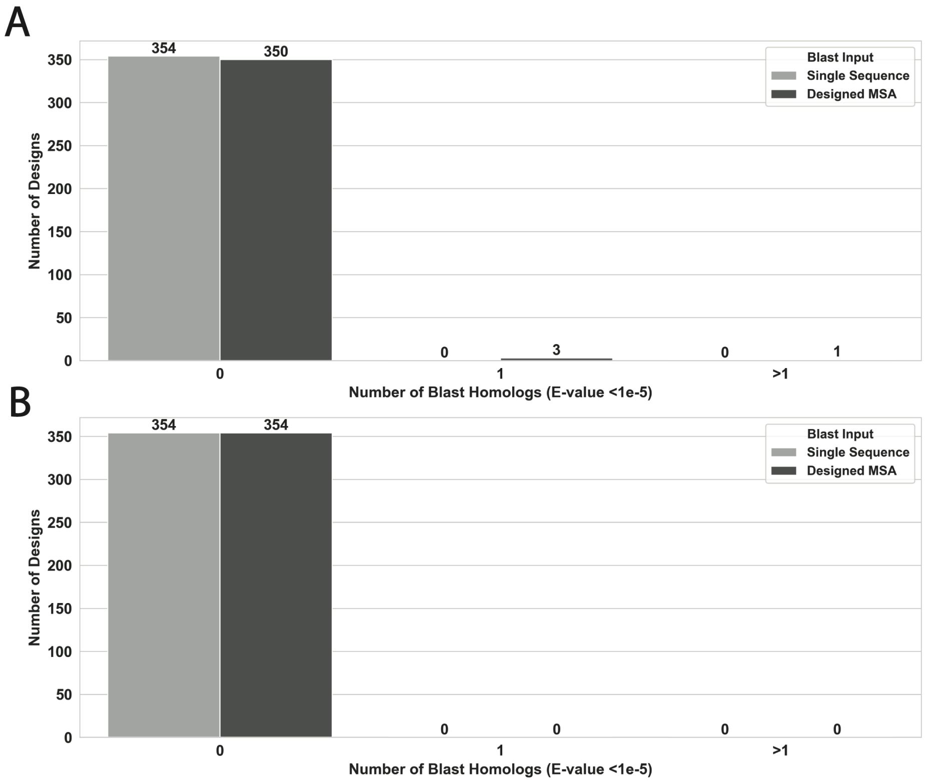


Figure I.7 Sequence homologs detected by searching the FoldDesign designs through the nr database using Blast. The sequences were designed by EvoEF2 (A) or RosettaFixBB (B). Two search strategies were used, either searching the single lowest energy sequence produced by EvoEF2/RosettaFixBB (Single Sequence) or jumpstarting the Blast search from the alignment of all 100 designed sequences (Designed MSA). The x-axis shows the number of Blast hits detected below an E-value threshold of 1e-5, while the y-axis shows the number of FoldDesign designs with the corresponding number of Blast hits.

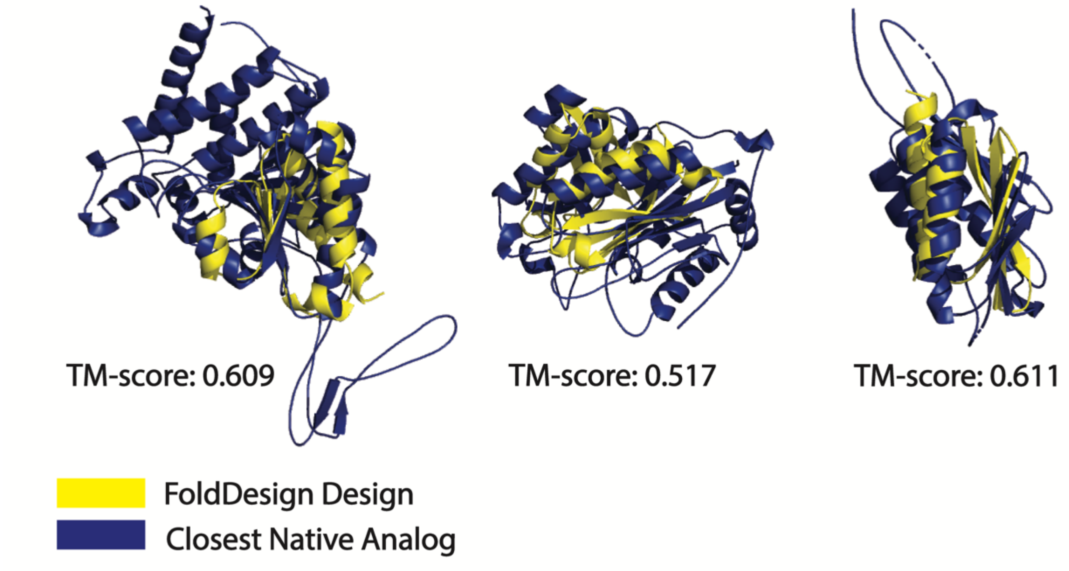


Figure I.8 Structural alignment between the designed proteins shown in Fig. 5.7.B and their closest native analogs in the PDB. The FoldDesign structures are shown in yellow, while the closest native analogs are shown in blue.

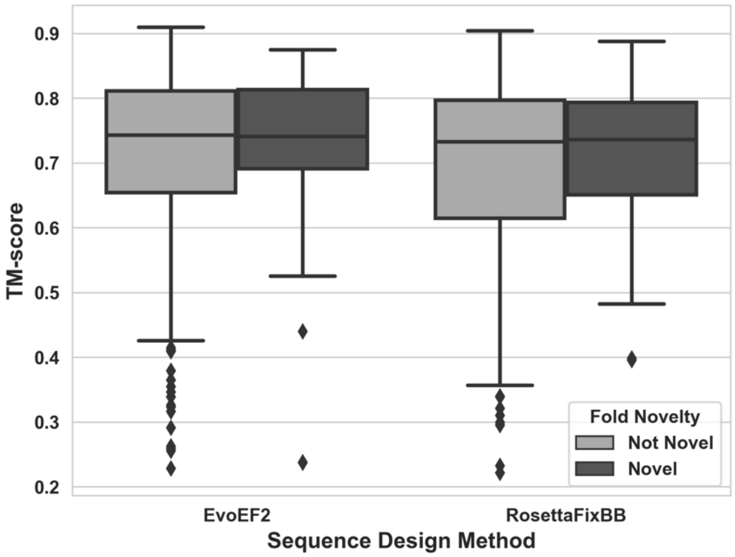


Figure I.9 AlphaFold2 structure prediction results for the 79 FoldDesign scaffolds with novel folds (Novel) and the 275 scaffolds with natural analogs (Not Novel). The y-axis depicts the TM-scores between the AlphaFold2 models and the designed scaffolds, while the x-axis separates the EvoEF2 and RosettaFixBB sequence designs.

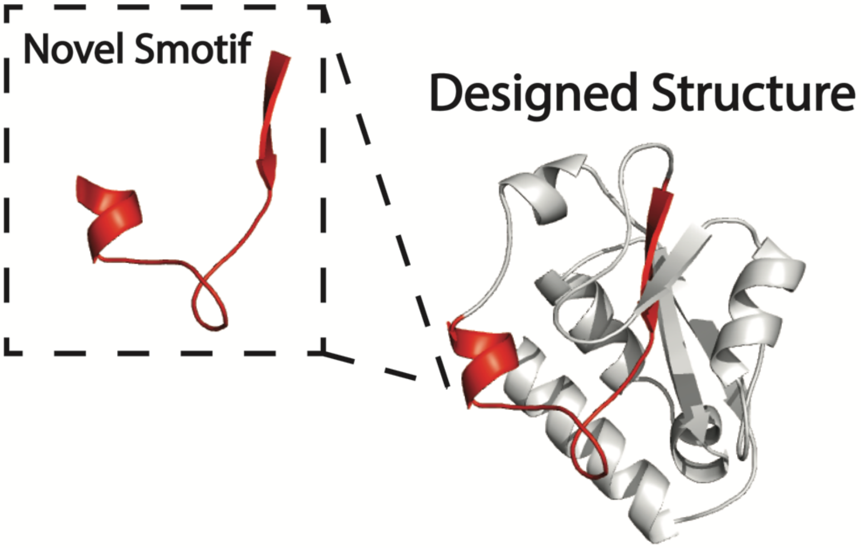


Figure I.10 Novel Smotif geometry. The novel Smotif produced by FoldDesign is shown in the inset and highlighted in red, while the remainder of the structure is shown in gray.

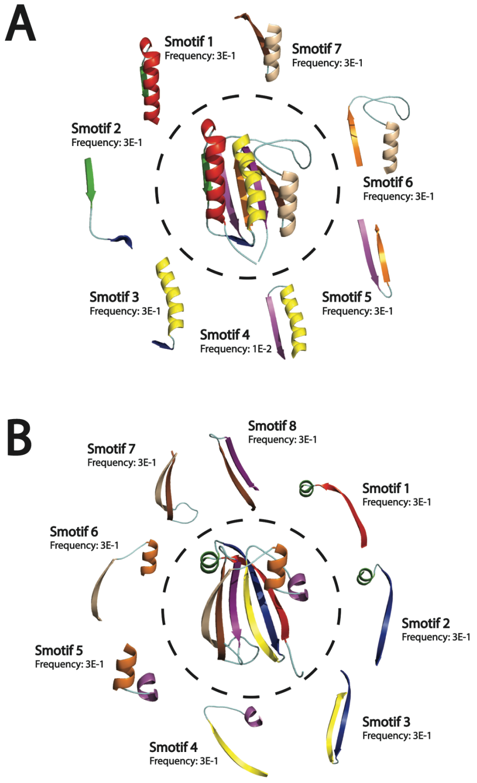


Figure I.11 Smotif geometries found in the native folds. A) Native fold for 1id0A as well as each Smotif in the structure. B) Native fold for 2p19A as well as each Smotif in the structure. The frequencies for each Smotif are the background frequencies calculated from the PDB.

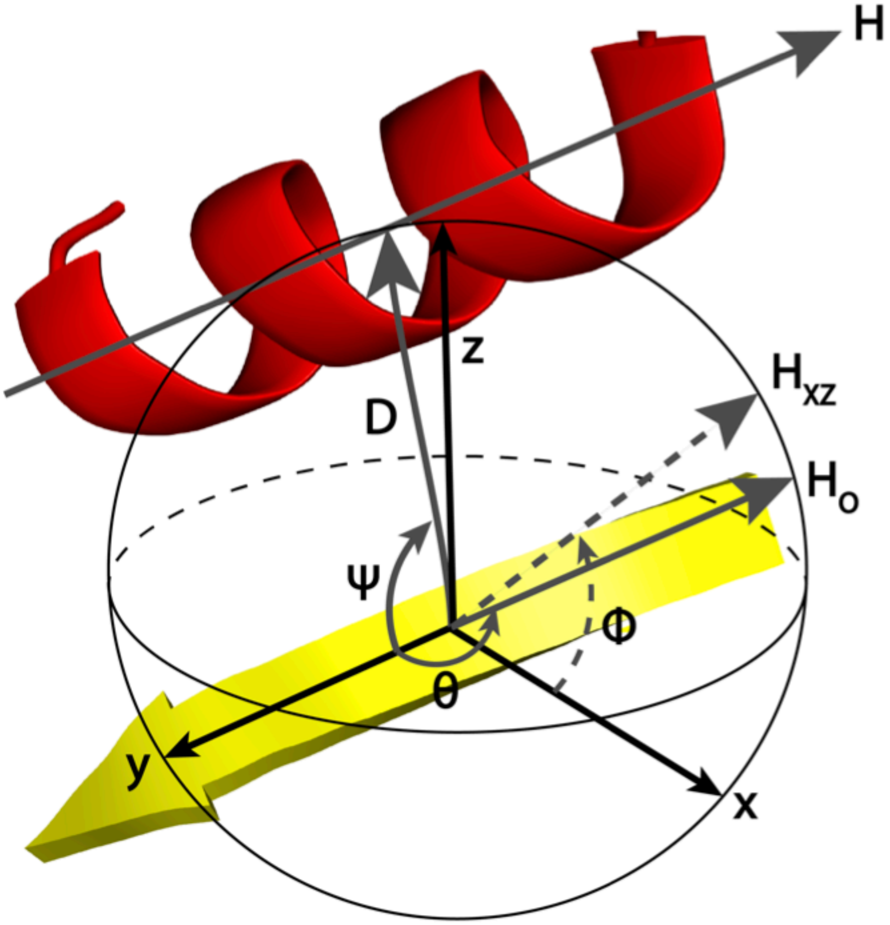


Figure I.12 Illustration of the features used to calculate the energy for packing two secondary structure elements. Note, here a helix and strand are used, but the parameters are the same for two helices or two strands. The y-axis is defined along the direction of the strand, where the origin is set at the center. D is a vector that represents the distance between the center of the strand and the center of the helix, and the x-axis is defined as the cross product between the y-axis vector and the D vector. The z-axis is defined as the cross product of the y-axis and the x-axis. H is the helical axis and H_0 is the helical axis translated to the origin. H_{xz} is the projection of H_0 onto the xz-plane. Lastly, ψ, ϕ, and θ are the angles between the y-axis and the D vector, the x-axis and H_{xz}, and the y-axis and H_0, respectively.

APPENDIX J

Supplementary Texts for Chapter V

Text J.1 Replica-exchange Monte Carlo simulation parameters and movements.

The conformational landscape is explored in FoldDesign using replica-exchange Monte Carlo (REMC) simulations. Within REMC, four parameters need to be carefully considered. First, the highest temperature (T_{max}) should be high enough to enable the simulation to overcome energy barriers, while the lowest temperature (T_{min}) should be low enough to ensure the simulation sufficiently scans the low-energy states. Second, the number of replicas (N_{rep}) should be large enough to ensure sufficient chances for the adjacent replicas to communicate with each other. Third, the number of local movements (N_{sweep}) before the global swaps should be selected to make the local Metropolis search achieve satisfactory equilibrium. After successive rounds of optimization, the final parameters were selected as: $T_{max} = min(20 * (1 + (L - 100) * 0.004), 20 * 2.5)$, $T_{min} = max(1 * (1 + (L - 100) * 0.001), 1 * 0.5)$, $N_{rep} = 40$, and $N_{sweep} = 30 * \sqrt{L}$, where L is the sequence length and a total of 500 REMC simulation cycles are carried out for each design.

Given the maximum and minimum temperature settings, the temperature at each replica *i* is determined using an inverse linear temperature scheme (267, 268). Briefly, the temperature for replica 1 is set to T_{max}, i.e., $T_1 = T_{max}$, and the temperature for the *i*th replica (*i >1*) is determined by the following equation:

$$T_i = \frac{1}{3 * \Delta\beta_{min\_max} + \beta_{i-1} + 12 * \Delta\beta_{\min\ \_max} * \frac{i}{N_{rep} - 2}} \qquad (J.1)$$

Here, β refers to an inverse temperature, where $\Delta\beta_{min\_max} = \frac{(\beta_{min}-\beta_{max})}{N_{rep}-1}$, $\beta_{min} = \frac{1}{T_{min}}$, $\beta_{max} = \frac{1}{T_{max}}$, and $\beta_{i-1} = \frac{1}{T_{i}-1}$. To illustrate the communication between replicas, Table H.3 presents the empirically observed acceptance probabilities for swaps between adjacent replicas during the design simulations for the 354 FoldDesign scaffolds. As can be seen from the table, the fraction of accepted swaps was similar across each of the adjacent replicas, where the average acceptance probability was 0.738, demonstrating a high degree of communication between the replicas.

During the REMC simulations, 11 different conformational movements are used by FoldDesign, as show in Fig. 5.13, to sample the structural space. Movements are accepted or rejected using the Metropolis Criterion (231) based on the associated changes in energy calculated by the energy function described in Text J.2. The major conformational movement is fragment substitution, where the decoy conformation in a selected region of the protein is replaced with the conformation from one of the highest scoring fragments. In order to perform this movement, it is first necessary to identify local fragments from a fragment library that match the input secondary structure topology. The fragment library is composed of 1-20 residue fragments from 29,156 high-resolution PDB structures used by QUARK (16, 92). The fragments were collected from structures deposited on or before 4/3/2014 and shared <30% sequence identity to each other (16, 92). Notably, this library has been extensively validated in the related field of protein structure prediction during even the most recent CASP experiments (20, 164). The information present for each fragment includes the position-wise backbone torsion angles (ϕ, ψ, ω), secondary structure, bond lengths, bond angles, solvent accessibility and Cα coordinates. During the movement, the backbone torsion angles (ϕ, ψ, ω) and backbone bond lengths and angles in the decoy structure are swapped with those present in the selected fragment. Next, cyclical coordinate descent loop closure (91) is used to connect the anchor points and prevent large downstream perturbations. Larger insertions are attempted at the beginning of the simulation, when the protein is largely unfolded, and smaller insertions are attempted as the protein become more compact.

In addition to fragment assembly, FoldDesign uses 10 auxiliary movements. The first of the auxiliary movements involves changing the length of one of the backbone bonds, including the N-Cα, Cα-C, or C-N bonds, by a random value in the range [-0.24 Å, 0.24Å], which is sampled from using a uniform probability distribution. The second movement involves randomly changing one of the backbone angles by a uniform random value in the range [-10°, 10°], including the N_i-$C\alpha_i$-

C_i, $C\alpha_i$-C_i-N_{i+1}, and C_i-N_{i+1}-$C\alpha_{i+1}$ angles, where i corresponds to the residue position. The third auxiliary movement changes one or more of the backbone torsion angles (ϕ, ψ, ω). The ϕ and ψ angles are updated by sampling from the allowed regions in the Ramachandran plots based on the input secondary structure at a given position. The ω angle is changed by a uniform random value selected from the range [-8°, 8°], where the movement is automatically rejected if it would result in the ω angle falling outside of the range of (170°, 190°). The fourth movement is LMProt perturbation (269), which randomly changes the positions of the backbone atoms in a selected region and then attempts to restrict all bond lengths and bond angles to physically allowable values. The fifth movement is segment rotation, which rotates the backbone atoms by a uniform random value in the range of (-90°, 90°) for a 2-12 residue segment along the axis defined by the Cα atoms of the first and last residues of the selected region. The sixth movement is similar to the fragment substitution movement but is based on fragment consensus from the 10 residue long fragments. To perform this movement, the 10 residue long identified fragments are clustered based on the distance matrix defined by their ϕ/ψ angle pairs. Then during the simulations, the ϕ/ψ angle pairs for a 10 residue segment in the decoy structure are swapped for the corresponding angles from the consensus fragments. The seventh movement is a segment shift. It involves shifting the residue numbers in a segment forward or backwards by one residue, which means that the coordinates of each residue are copied from their preceding or subsequent residues in the segment. We then delete the unused coordinates of one residue at the selected terminal region and insert new coordinates for another residue at the other terminal based on physically allowable bond lengths and angles. This movement can easily adjust the β-pairing in two well-aligned β-strands. The eighth auxiliary movement is β-turn formation, which attempts to form a β-turn in regions of the protein whose input secondary structure is defined as coiled. The final two movements are β-strand and α-helix formation. For these two movements, two regions that are defined as β-strands or α-helices are moved closer together based on distance and torsion angle distributions collected from the PDB.

Text J.2 FoldDesign energy function.

The energy function used to guide the FoldDesign simulations is a combination of 10 energy terms:

$$E_{DeepFold} = E_{HB} + E_{ss\_satisfaction} + E_{rama} + E_{hhpack} + E_{sspack} + E_{\mathrm{hspack}} + E_{ev} + E_{generic\_dist} + E_{frag\_dist\_profile} + E_{frag\_solv} + E_{rg} + E_{contact\_num} \qquad (J.2)$$

where E_{HB}, $E_{ss\_satisfaction}$, E_{rama}, E_{hhpack}, E_{sspack}, E_{hspack}, E_{ev}, $E_{generic\_dist}$, $E_{frag\_dist\_profile}$, E_{frag_solv}, E_{rg}, and $E_{contact\_num}$ are terms for backbone hydrogen bonding, secondary structure satisfaction, Ramachandran torsion angles, helix-helix packing, strand-strand packing, helix-strand packing, excluded volume, generic backbone atom distances, fragment-derived distance restraints, fragment-derived solvent accessibility, radius of gyration, and expected contact number, respectively. The equations for each energy term are detailed below.

E_{HB} is calculated as follows:

$$E_{HB} = \sum_{i,j,T_k} E_{hb_{feat}}(i,j,T_k) \qquad (J.3)$$

where *i* and *j* are the residue indices and *T_k* is the kth type of hydrogen bonding restraint. In FoldDesign, there are 4 types of hydrogen bonding restraints: hydrogen bonds between residues *i* and *i+4* in regions defined as helical by the input secondary structure (T_1), virtual hydrogen bonds between residues *i* and *i+3* in regions defined as helical by the input secondary structure (T_2), and hydrogen bonds between residues *i* and *j* in parallel β-strands (T_3) or antiparallel β-strands (T_4) for regions defined as strands by the input secondary structure. The energy for each type of hydrogen bonding restraint is calculated using the following equation:

$$E_{hb_{feat}}(i,j,T_k) = \sum\nolimits_{l=1}^{n_k} \frac{(f_l(i,j) - \mu_{kl})^2}{2\delta_{kl}^2}, \quad n_k = \begin{cases} 4 & k = 1,2 \\ 3 & k = 3,4 \end{cases} \qquad (J.4)$$

where $f_l(i,j)$ is the value of the l^{th} feature from the decoy structure, n_k is the number of features considered for the k^{th} type of hydrogen bond restraint, μ_{kl} is the average value of the l^{th} feature for the k^{th} type of hydrogen bond restraint calculated from the PDB library, and δ_{kl} is the standard deviation of the l^{th} feature for the k^{th} type of hydrogen bond restraint. For hydrogen bonding, we consider four features: the distance, $D(O_i,H_j)$, between backbone atom O_i from residue *i* and the backbone hydrogen, H_j, from residue j, the angle, $A(C_i,O_i,H_j)$, between backbone atoms C_i and O_i from residue i and the backbone hydrogen, H_j, from residue j, the angle, $A(C_i,O_i,H_j)$, between backbone atom O_i from residue i and the backbone hydrogen, H_j, and nitrogen, N_j, from residue j, and the torsion angle, $T(C_i,O_i,Hj,N_j)$, between atoms C_i and O_i from residue i and the backbone hydrogen, H_j, and nitrogen, N_j, from residue j. Note for hydrogen bonding in strand regions, T_3 and T_4 restraints, $T(C_i,O_i,Hj,N_j)$ is not considered as there is a large standard deviation for this feature in strand regions. The values of μ_{kl} and δ_{kl} are shown in Table H.4.

$E_{ss\_satisfaction}$ is calculated as follows:

$$E_{ss\_satisfaction} = -\sum_{i=1}^{i=L} \begin{cases} -2 \quad if\ ss_i = helix\ and\ input_{ss_i} = strand\ or\ ss_i = strand\ and\ input_{ss_i} = helix \\ 1 \quad if\ ss_i = coil\ and\ input_{ss_i} = coil \\ 2 \quad if\ ss_i = helix\ and\ input_{ss_i} = helix\ or\ ss_i = strand\ and\ input_{ss_i} = strand \\ -1 \quad else \end{cases} \qquad (J.5)$$

where ss_i is the secondary structure of the decoy at position i and $input\_ss_i$ is the input secondary structure at the corresponding position. If the input secondary structure is defined as helical and the secondary structure of the decoy structure is a strand or if the input secondary structure is defined as a strand and the secondary structure of the decoy structure is helical, then a penalty of -2 is assigned to penalize opposite secondary structure assignments more heavily. Similarly, if the helical or strand regions are correct in the decoy structure, then a stronger bonus is assigned. Mismatches in coiled regions are penalized less heavily, and correctly generated coiled regions are also rewarded to a lesser degree as they are more flexible and lack regular hydrogen bonding patterns.

E_{rama} is calculated as follows:

$$E_{rama} = -\sum_{i=2}^{i=L-1} \log\left(P(\phi_i, \psi_i) \mid input_{ss_i}\right) \qquad (J.6)$$

where ϕ_i and ψ_i are the backbone torsion angles at position i and $input\_ss_i$ is the input secondary structure at position i. The probabilities for each backbone torsion angle pair were determined from the I-TASSER (17) PDB library based on the secondary structure at a given position.

E_{hhpack}, E_{sspack}, and E_{hspack} are calculated as follows:

$$\begin{cases} E_{hhpack} = -\sum_{i,j} \log\left(P_{hh}\left(\psi_{ij}, \theta_{ij}, \Phi_{ij}\right) \mid seq\_sep\right) - \sum_{i,j} \log\left(P_{hh}\left(D_{ij}, \theta_{ij}\right) \mid seq\_sep\right) \\ E_{sspack} = -\sum_{i,j} \log\left(P_{ss}\left(\psi_{ij}, \theta_{ij}, \Phi_{ij}\right) \mid seq\_sep\right) - \sum_{i,j} \log\left(P_{ss}\left(D_{ij}, \theta_{ij}\right) \mid seq\_sep\right) \\ E_{hspack} = -\sum_{i,j} \log\left(P_{hs}\left(\psi_{ij}, \theta_{ij}, \Phi_{ij}\right) \mid seq\_sep\right) - \sum_{i,j} \log\left(P_{hs}\left(D_{ij}, \theta_{ij}\right) \mid seq\_sep\right) \end{cases} \quad (J.7)$$

where ψ_{ij}, θ_{ij}, Φ_{ij} are the angles between two secondary structure elements (either two helices, E_{hhpack}, two strands E_{sspack}, or a helix and a strand, E_{hspack}) defined in Fig. I.12, D_{ij} is the distance between the centers of the two secondary structure elements, and seq_sep is the number of residues between two secondary structure elements along the sequence. The potential is split into three different groups depending on the sequence separation, including short, medium, and long-range interactions. Here, short, medium, and long-range refers to residue pairs (*i,j*) that fall in the following ranges, respectively: $6 \leq |i-j| < 12$, $12 \leq |i-j| < 24$, and $|i-j| \geq 24$. The secondary structure specific probabilities distributions for the features were derived from PDB structures in the I-TASSER library and were fit using kernel density estimation to smooth the potentials.

For the estimation of $P(\psi, \theta, \Phi)$, the periodic von Mises probability distribution was used as the kernel function (k_{angle}); specifically $k_{angle}(x, \kappa) = \frac{1}{2\pi I_0(k)} \exp\left(\kappa * cos(x)\right)$, where *x is an angle value,* κ is a tunable concentration parameter*, and* I_0 is the modified Bessel function of the first kind of order zero. Thus, the probability distribution, $P(\psi, \theta, \Phi)$, for each of the three interaction types and sequence separation categories was estimated by $P(\psi, \theta, \Phi|\kappa) = \frac{1}{N}\sum_{i=1}^{N} k_{angle}(\psi - \psi_i, \kappa) k_{angle}(\theta - \theta_i, \kappa) k_{angle}(\Phi - \Phi_i, \kappa)$. Here, Φ was computed over the range [0°, 360°), while θ and Φ were computed over the range [0°, 180°], where a bin size of 1° was used for each angle. Additionally, i denotes the index of the datapoint derived from the PDB dataset for observed values of ψ, θ, and Φ, where the summation was carried out over the *N* datapoints in the dataset for each interaction type and sequence separation category. Lastly, the concentration parameter, κ, may be tuned, where the larger the value of κ is, the narrower the kernels will be. To optimize this parameter, the dataset was randomly divided into 10 equal subsets and the value of κ was varied from 0° to 180° by an increment of 1°, where the value that resulted in the maximum mean log-likelihood for the observed angles across the 10 subsets was used for each interaction type and sequence separation.

For the estimation of $P(D,\theta)$, the same periodic von Mises function was used as the kernel for θ. However, for the distance, *D*, a non-periodic gaussian distribution was used as the kernel function (k_{dist}), specifically $k_{dist}(D,h) = \frac{1}{\sqrt{2\pi h}}\exp\left(\frac{-D^2}{2h^2}\right)$, where *D* is a distance and h is the bandwidth parameter. Thus, the probability distribution, $P(D,\theta)$, for each of the three interaction types and sequence separation categories was estimated by $P(\psi,\theta,\Phi|\kappa,h) = \frac{1}{N}\sum_{i=1}^{N} k_{dist}(D - D_i,h)k_{angle}(\theta-\theta_i,\kappa)$. Here, θ was computed over the range [0°, 180°] with a bin size of 1°, while D was computed over the range [0, 20Å] with a bin size of 0.1 Å. As before, i denotes the index of the datapoint derived from the PDB dataset for observed values of D and θ, where the summation was carried out over the *N* datapoints in the dataset for each interaction type and sequence separation category. Again, κ and h are tunable parameters, where κ was varied from 0° to 180° by an increment of 1°, while h was varied from 0.1 Å to 20 Å using an increment of 0.1 Å. As before, the optimal values of these parameters were determined by randomly splitting the dataset into 10 subsets and selecting the values that resulted in the highest mean log-likelihood across all 10 datasets for the observed values.

E_{ev} is calculated as follows:

$$E_{ev} = \sum_{i=1}^{i=L}\sum_{j=i+1}^{j=L}\sum_{ii}\sum_{jj}\begin{cases}(vdw(i,ii)+vdw(j,jj))^2 - r_{ii,jj}^2 & if\ r_{ii,jj} < vdw(i,ii)+vdw(j,jj)\\ 0\ else\end{cases} \quad (J.8)$$

where clashes are calculated between each atom ii from residue i and atom jj from residue j and $r_{ii,jj}$ is the distance between the two atoms. For the side-chain center atoms, the center of mass of valine is used to assess steric clashes. All atoms presented in Fig. 5.12 are considered except for hydrogen.

$E_{generic\_dist}$ is calculated as follows:

$$E_{generic\_dist} = \sum\nolimits_{i=1}^{i=L} \sum\nolimits_{j=i+1}^{j=L} \sum_{ii} \sum_{jj} -RT * \log\left(\frac{N_{obs}\left(ii, jj, r_{ii,jj}\right)}{r_{ii,jj}^{\alpha} N_{obs}(ii, jj, r_{cut})}\right) \qquad (J.9)$$

where L is the protein length, *i* and *j* are the two residue indices and *ii/jj* are the atoms N, Cα, C, O and Cβ. $N_{obs}(ii, jj, r_{ii,jj})$ is the observed number of pairs between atoms *ii* and *jj* with distance $r_{ii,jj}$ determined from the I-TASSER PDB library. A cutoff, r_{cut}, of 15Å is used and the distances for the observed atom pairs is divided into 0.5Å bins from 0Å to 15Å. The potential is similar to DFIRE, where $\alpha = 1.61$ and $N_{obs}(ii, jj, r_{cut})$ is used to calculate the background probability.

$E_{frag\_dist\_profile}$ is calculated as follows:

$$E_{frag\_dist\_profile} = -\sum_{(i,j)\subseteq S_{dp}} \log\left(N_{ij}\left(d_{ij}\right)\right) \qquad (J.10)$$

where d_{ij} is the distance between the Cα atoms of residues *i* and *j* in the decoy structure and N_{ij} is the distance profile for residues *i* and *j* extracted from the 10 residue long fragments where *d* falls in the range [0Å, 9Å] with a bin width of 0.5 Å. S_{dp} is the set of residues that have fragment-derived distance profiles. To derive the distance profiles, we first analyze each of the 10 residue fragments that originate from the same PDB structure and are aligned to different residues, *i* and *j*. Then we calculate the distance between the Cα atoms for the two positions from the fragments based on their corresponding positions in their PDB structure. If the distance between the two residues in the PDB structure is <9Å, then these positions may be encouraged to from contacts in the designed structure. This procedure is repeated for each query residue pair (*i, j*) to construct a histogram of distances. If the histogram for a given pair of residues has a peak <9Å, then the histogram is saved to calculate the distance profile energy and the residue pair is added to the set S_{dp}.

E_{frag_solv} is calculated as follows:

$$E_{frag\_solv} = \sum_{i=1}^{i=L} \left| s_i - s_i^E \right| \qquad (J.11)$$

where L is the protein length, s_i is the solvent accessibility of residue *i* in the decoy structure, and s_i^E is the expected solvent accessibility derived from the 20 residue fragments. The following formula is used to calculate s_i:

$$s_i = 1 - 0.007 \sum_{d(G_i, G_j) < 9\text{Å}} \frac{A_{aa(j)}}{d^2\left(G_i, G_j\right)} \qquad (J.12)$$

Here, $A_{aa(j)}$ is the maximum solvent accessible surface area for the given residue aa at position *j*. Since polyvaline sequences are used in FoldDesign, the maximum solvent accessible surface area for Valine is used. G_i and G_j are the geometric centers of residues *i* and *j*, $d\left(G_i, G_j\right)$ is the distance between the two geometric centers, and $d^2(G_i, G_j)$ is the squared distance. A cutoff of 9Å is used as residues that are further apart contribute little to the solvent accessibility. As mentioned above, s_i^E is the expected solvent accessibility calculated from the overlapping 20 residue fragments. For each fragment, the solvent accessibility of the residue in its native PDB structure is recorded, and the estimated solvent accessibility is calculated by averaging the solvent accessibility of each fragment residue aligned to position *i*.

E_{rg} is calculated as follows:

$$E_{rg} = \begin{cases} 0 & r_{min} \leq r \leq r_{max} \\ (r_{min} - r)^2 & r < r_{min} \\ (r - r_{max})^2 & r > r_{max} \end{cases} \qquad (J.13)$$

where r is the radius of gyration for the decoy structure calculated from the Cα positions produced during the FoldDesign simulations and r_{min}/r_{max} are the estimated minimum and maximum radii of gyration calculated from the PDB based on the protein length and secondary structure composition. More specifically, the minimum and maximum radii of gyration are estimated following previous work in protein structure prediction by QUARK (16), where $r_{min} =$

$2.316L^{0.358} - 0.5$ and $r_{max} = max\{r_{min} + 8.0, 0.5\sqrt{3/5}N_{maxh}\}$. Here, N_{maxh} is the length of the longest helix in the structure and L is the protein length. Using these values, 95% of the experimental structures in the PDB have a radius of gyration within $[r_{min}, r_{max}]$ (16).

$E_{contact\_num}$ is calculated as follows:

$$\begin{aligned} E_{contact\_num} = & \left|num_{short\_cont} - expected\_num_{short\_cont}\right| \\ & + \left|num_{med\_cont} - expected\_num_{med\_cont}\right| \\ & + \left|num_{long\_cont} - expected\_num_{long\_cont}\right| \qquad (J.14) \end{aligned}$$

where num_{short_cont}, num_{med_cont}, and num_{long_cont} are the number of short, medium, and long-range contacts in the decoy structure. Here, short, medium, and long-range contacts refer to residue pairs (*i,j*) that fall in the following ranges, respectively: $6 \leq |i - j| < 12$, $12 \leq |i - j| < 24$, and $|i - j| \geq 24$. $expected\_num_{short\_cont}$, $expected\_num_{med\_cont}$, and $expected\_num_{long\_cont}$ are the expected short, medium, and long-range contacts calculated from PDB structures in the I-TASSER library based on protein length.

Text J.3 Rosetta protocol used to generate designed folds.

The following command was used to generate backbones by Rosetta:

```
<rosetta_bin>/main/source/bin/rosetta_scripts.static.linuxgccrelease -database <rosetta_bin>/main/database/ -s ./input.pdb -parser:protocol ./backbone_generation.xml -nstruct 250
```

The contents of the backbone_generation.xml files are detailed below, which were adapted from a representative recent publication (270).

```
<ROSETTASCRIPTS>
    <SCOREFXNS>
      <ScoreFunction name="SFXN1" weights="fldsgn_cen_omega02.wts" />
    </SCOREFXNS>
    <FILTERS>
        <ScoreType name="cen_total" scorefxn="SFXN1" score_type="total_score" threshold="1000000" />
        <ScoreType name="vdw" scorefxn="SFXN1" score_type="vdw" threshold="1000000" />
        <ScoreType name="rg" scorefxn="SFXN1" score_type="rg" threshold="1000000" />
        <ScoreType name="cen_rama" scorefxn="SFXN1" score_type="rama" threshold="1000000" />
        <ScoreType name="sspair" scorefxn="SFXN1" score_type="ss_pair" threshold="1000000" />
        <ScoreType name="rsigma" scorefxn="SFXN1" score_type="rsigma" threshold="1000000" />
    </FILTERS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <MOVERS>
        <Dssp name="dssp"/>
        <SwitchResidueTypeSetMover name="fullatom" set="fa_standard"/>
        <SwitchResidueTypeSetMover name="cent" set="centroid"/>
        <MakePolyX name="polyval" aa="VAl" keep_pro="1" />
        <BluePrintBDR name="bdr1" scorefxn="SFXN1" use_abego_bias="1" blueprint="blueprint.xml"/>
        <MinMover name="min1" scorefxn="SFXN1" chi="1" bb="1" type="dfpmin_armijo_nonmonotone_atol" 
tolerance="0.0001"/>
        <ParsedProtocol name="cenmin1" >
          <Add mover_name="cent" />
          <Add mover_name="min1" />
          <Add mover_name="fullatom" />
        </ParsedProtocol>
        <ParsedProtocol name="bdr1ss" >
          <Add mover_name="bdr1" />
          <Add mover_name="cenmin1" />
          <Add mover_name="dssp" />
        </ParsedProtocol>
    </MOVERS>
    <PROTOCOLS>
        <Add mover_name="bdr1ss" />
        <Add mover_name="fullatom" />
        <Add filter_name="cen_total" />
        <Add filter_name="vdw" />
        <Add filter_name="rg" />
        <Add filter_name="cen_rama" />
        <Add filter_name="sspair" />
        <Add filter_name="rsigma" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

The contents of the weights file (fldsgn_cen_omega02.wts) were as follows, which were also adapted from the previous study (270):

```
vdw 1.0
rg 1.0
rama 0.1
hs_pair 1.0
ss_pair 1.0
rsigma 1.0
omega 0.5
hbond_lr_bb 1.0
hbond_sr_bb 1.0

STRAND_STRAND_WEIGHTS 1 11
```

Here, for each input topology, 250 designs were generated using Rosetta, where the final designs were selected from the lowest energy structures as assessed by the Rosetta centroid energy function. In terms of the total number of conformational movements, the average number of movements attempted by Rosetta per design was 8,291,689.9, not including the L-BFGS-based minimization, which was slightly higher than the 6,000,000 movements attempted by FoldDesign for each design. This protocol follows the standard, widely used fragment assembly-based design procedure by Rosetta, where topologies are defined by the BluePrintBDR mover and built using stepwise Monte Carlo fragment assembly simulations guided by the Rosetta centroid energy function (271). Following this, the designs were minimized using L-BFGS optimization of the internal coordinates and filtered using a combination of score thresholds. Since the purpose of the benchmark tests was to perform fully automated *de novo* protein design, no user-provided restraints were utilized other than the 3-state secondary structure sequences. An example of the Rosetta blueprint files without and with ABEGO bias are provided in Texts J.8 and J.9 (see below), respectively.

Text J.4 Analysis of the results with ABEGO bias and sub-rotamer sampling.

In Chapter 5, Rosetta was run without ABEGO bias, which divides the Ramachandran plot into 4 regions (A,B,E,G) and restricts the fragment selection to the region defined by the specified bias for each residue (249). This bias allows for more control over the fragment selection process and fold definition; however, given that the benchmark dataset was composed of just the 3-state SS sequences from the native proteins, the proper ABEGO definition for each position is ambiguous as the same SS type can be sampled from multiple regions of the Ramachandran plot, e.g., right-handed (ABEGO region A) vs. left-handed alpha helices (ABEGO region G). Nevertheless, given that this bias is often used, we reran Rosetta and restricted helical regions to the A region of the Ramachandran plot and strands to the B region of the Ramachandran plot (249). We then calculated the percent of buried residues/SASA and the GOAP/ROTAS energies for the Rosetta designs that utilized ABEGO bias, where the results are summarized in Fig. I.2. This analysis showed that there was not a significant difference in the percent of buried residues/SASA or the GOAP/ROTAS energies between the designs that utilized ABEGO bias and those that did not (with *p*-values >0.05).

Additionally, similar to EvoEF2, RosettaFixBB was run without sub-rotamer sampling (see Text J.6 for the RosettaFixBB protocol). To examine if enabling additional rotameric sampling during the sequence design impacted the results, we reran RosettaFixBB with χ_1 and χ_2 sub-rotamer sampling enabled for the FoldDesign and Rosetta scaffolds (see Text J.7 for the RosettaFixBB protocol with sub-rotamers enabled), where the results are depicted in Fig. I.3. Overall, only the ROTAS energy improved significantly (*p*-values <0.05) with the addition of sub-rotamer sampling, which may be expected as ROTAS places special emphasis on the rotameric conformations adopted by the side-chains (248). Nevertheless, the FoldDesign scaffolds still had significantly lower ROTAS energies (-10684.5) than the Rosetta scaffolds (-9446.4) with a *p*-value of 7.7E-08. Thus, enabling sub-rotamer sampling and including ABEGO bias did not alter the conclusions drawn in the text, where it would be expected that any improvements in the sequence design protocol would benefit both FoldDesign and Rosetta.

Text J.5 Analysis of the amino acid compositions of the designed scaffolds.

Given that Valine is used as the generic center of mass in FoldDesign and Rosetta (see 5.3 Methods), one important issue is to examine whether the designed scaffolds exhibited any systematic bias against particular amino acids, such as smaller non-polar residues like Glycine and Alanine as well as bulkier aromatic amino acids or Proline. In Fig. I.4, we plot the frequency of each of the 20 amino acids in the EvoEF2/RosettaFixBB designed sequences for the FoldDesign and Rosetta scaffolds compared to the frequency from the corresponding native protein sequences. As expected, the specific amino acid preferences varied depending on the sequence design method that was used; however, it can be observed that there was no bias towards Valine for FoldDesign or Rosetta, and smaller non-polar amino acids such as Glycine and Alanine were well represented in the designed sequences, as well as bulkier amino acids like Tryptophan, Tyrosine, and Proline, with some variation for Proline and Alanine depending on the sequence design method. Quantitatively, the Kullback-Leibler (KL) divergence between the native amino acid distribution and the distributions for the EvoEF2/RosettaFixBB sequence designs for the FoldDesign scaffolds was 0.236/0.122, which was slightly lower than the KL divergence for the Rosetta scaffolds (0.352/0.123). In addition, since FoldDesign does not include any chirality restraints on the backbone torsion angles during the folding simulations, the designed folds contained structures with both right- and left-handed helices and covered the full diversity of the torsion angle space adopted by natural proteins as highlighted in the Ramachandran plot (Fig. I.5).

Text J.6 RosettaFixBB protocol without sub-rotamer sampling.

The following command was used to generate sequence designs by RosettaFixBB without sub-rotamer sampling:

```
<rosetta_bin>/main/source/bin/fixbb.static.linuxgccrelease -database <rosetta_bin>/main/database/ -s ./design.pdb -nstruct 100
```

Text J.7 RosettaFixBB protocol with sub-rotamer sampling.

The following command was used to generate sequence designs by RosettaFixBB with χ_1 and χ_2 sub-rotamer sampling:

```
<rosetta_bin>/main/source/bin/fixbb.static.linuxgccrelease -database <rosetta_bin>/main/database/ -s ./design.pdb -nstruct 100 -
ex1 -ex2
```

Text J.8 Example Rosetta blueprint file without ABEGO bias.

The following illustrates the contents of the Rosetta blueprint file without ABEGO bias for the secondary structure topology derived from 2jx8A.

```
1 V L R
2 V L R
0 V H R
0 V H R
0 V H R
0 V H R
```

0VHR

0VHR

0VLR

0VLR

0VLR

0VER

0VER

0VER

0VER

0VLR

0VLR

0VLR

0VLR

0VER

0VER

0VER

0VER

0VER

0VER

0VLR

0VLR

0VLR

0VLR

0VER

0VER

0VER

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

0VLR

Text J.9 Example Rosetta blueprint file with ABEGO bias.

The following illustrates the contents of the Rosetta blueprint file with ABEGO bias for the secondary structure topology derived from 2jx8A.

```
1 V L  R
2 V L  R
0 V HA R
0 V HA R
0 V HA R
0 V HA R
0 V HA R
0 V HA R
0 V L  R
0 V L  R
0 V L  R
0 V EB R
0 V EB R
0 V EB R
0 V EB R
0 V L  R
0 V L  R
0 V L  R
0 V L  R
0 V EB R
0 V EB R
0 V EB R
0 V EB R
0 V EB R
0 V EB R
0 V L  R
0 V L  R
0 V L  R
0 V L  R
0 V EB R
0 V EB R
0 V EB R
0 V L  R
0 V L  R
0 V L  R
0 V L  R
```

```
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
0 V L   R
```

Text J.10 Relative frequency of Smotifs for the test protein structures.

In Fig. 5.10, we first split the Smotifs into 4 bins based on the normalized background frequency of the Smotifs that appear in the PDB structures, i.e., [0, 1E-3], (1E-3, 1E-2], (1E-2, 1E-1], and (1E-1, 1], where the normalized background frequency of a Smotif is equal to the number of times that the Smotif appeared in the 51,094 non-redundant full-chain structures in the I-TASSER template library divided by the total number of Smotifs in the structural library.

For a given protein i in the test set of the 79 novel folds or the 354 native structures, the relative frequency of Smotifs for one of the 4 bins, j, is calculated by

$$Relative\ Frequency\ (i,j) = \frac{Num\_Smotif_{i,j}}{\sum_{j=1}^{j=4} Num\_Smotif_{i,j}} \qquad (J.15)$$

where $Num\_Smotif_{i,j}$ is the number of Smotifs from the i-th protein that fall into the j-th bin.

BIBLIOGRAPHY

1. C. B. Anfinsen, Principles That Govern Folding of Protein Chains. *Science* **181**, 223-230 (1973).

2. J. P. Glusker, X-ray crystallography of proteins. *Methods Biochem Anal* **37**, 1-72 (1994).

3. Cavanaugh J., Fairbrother W. J., Palmer A.G., S. N., *Protein NMR spectroscopy: principles and practice* (New York: Academic Press, 1996).

4. Y. Cheng, Single-Particle Cryo-EM at Crystallographic Resolution. *Cell* **161**, 450-457 (2015).

5. H. M. Berman *et al.*, The Protein Data Bank. *Nucleic Acids Res* **28**, 235-242 (2000).

6. A. Bairoch *et al.*, The Universal Protein Resource (UniProt). *Nucleic Acids Res* **36**, D190-D195 (2008).

7. R. N. Consortium, RNAcentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic Acids Res* **49**, D212-D220 (2021).

8. M. Levitt, A. Warshel, Computer-simulation of protein folding. *Nature* **253**, 694-698 (1975).

9. P. N. Lewis, F. A. Momany, H. A. Scheraga, Folding of polypeptide chains in proteins - proposed mechanism for folding. *P Natl Acad Sci USA* **68**, 2293-& (1971).

10. J. A. Mccammon, B. R. Gelin, M. Karplus, Dynamics of folded proteins. *Nature* **267**, 585-590 (1977).

11. J. U. Bowie, R. Luthy, D. Eisenberg, A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **253**, 164-170 (1991).

12. J. Skolnick, A. Kolinski, Simulations of the folding of a globular protein. *Science* **250**, 1121-1125 (1990).

13. A. Sali, T. L. Blundell, Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779-815 (1993).

14. K. T. Simons, C. Kooperberg, E. Huang, D. Baker, Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.* **268**, 209-225 (1997).

15. A. Roy, A. Kucukural, Y. Zhang, I-TASSER: a unified platform for automated protein structure and function prediction. *Nat Protoc* **5**, 725-738 (2010).

16. D. Xu, Y. Zhang, Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* **80**, 1715-1735 (2012).

17. J. Yang *et al.*, The I-TASSER Suite: protein structure and function prediction. *Nature Methods* **12**, 7-8 (2015).

18. S. Ovchinnikov *et al.*, Protein structure determination using metagenome sequence data. *Science* **355**, 294-298 (2017).

19. S. Wang, S. Q. Sun, Z. Li, R. Y. Zhang, J. B. Xu, Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *Plos Comput Biol* **13** (2017).

20. W. Zheng *et al.*, Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins* 10.1002/prot.25792 (2019).

21. A. W. Senior *et al.*, Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706-710 (2020).

22. J. Yang *et al.*, Improved protein structure prediction using predicted interresidue orientations. *Proc Natl Acad Sci U S A* **117**, 1496-1503 (2020).

23. D. Fischer, D. Eisenberg, Assigning folds to the proteins encoded by the genome of Mycoplasma genitalium. *Proc Natl Acad Sci U S A* **94**, 11929-11934 (1997).

24. R. Sanchez, A. Sali, Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins* **Suppl. 1**, 50-58 (1997).

25. Y. Zhang, J. Skolnick, Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl. Acad. Sci. USA* **101**, 7594-7599 (2004).

26. L. Malmstrom *et al.*, Superfamily assignments for the yeast proteome through integration of structure prediction with the gene ontology. *PLoS biology* **5**, e76 (2007).

27. S. Mukherjee, A. Szilagyi, A. Roy, Y. Zhang, "Genome-wide protein structure prediction" in Multiscale approaches to protein modeling: structure prediction, dynamics, thermodynamics and macromolecular assemblies*,* A. Kolniski, Ed. (Springer-London, 2010), pp. 810-842.

28. D. Xu, Y. Zhang, Ab Initio structure prediction for Escherichia coli: towards genome-wide protein structure modeling and fold assignment. *Sci Rep* **3**, 1895 (2013).

29. C. Zhang *et al.*, Functions of Essential Genes and a Scale-Free Protein Interaction Network Revealed by Structure-Based Function and Interaction Prediction for a Minimal Genome. *J Proteome Res* **20**, 1178-1189 (2021).

30. D. E. Kim, D. Chivian, D. Baker, Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res* **32**, W526-531 (2004).

31. L. A. Kelley, M. J. Sternberg, Protein structure prediction on the Web: a case study using the Phyre server. *Nat Protoc* **4**, 363-371 (2009).

32. T. Schwede, J. Kopp, N. Guex, M. C. Peitsch, SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Res* **31**, 3381-3385 (2003).

33. J. Soding, A. Biegert, A. N. Lupas, The HHpred interactive server for protein homology detection and structure prediction. *Nucleic. Acids Res.* **33**, W244-248 (2005).

34. Z. Wang, J. Eickholt, J. Cheng, MULTICOM: a multi-level combination approach to protein structure prediction and its assessments in CASP8. *Bioinformatics* **26**, 882-888 (2010).

35. M. Källberg *et al.*, Template-based protein structure modeling using the RaptorX web server. *Nat. Protocols* **7**, 1511-1522 (2012).

36. J. Xu, Distance-based protein folding powered by deep learning. *Proc Natl Acad Sci U S A* **116**, 16856-16865 (2019).

37. N. Vaidehi *et al.*, Prediction of structure and function of G protein-coupled receptors. *P Natl Acad Sci USA* **99**, 12622-12627 (2002).

38. Y. Zhang *et al.*, Three-dimensional structural view of the central metabolic network of Thermotoga maritima. *Science* **325**, 1544-1549 (2009).

39. Y. Loewenstein *et al.*, Protein function annotation by homology-based inference. *Genome Biol* **10**, 207 (2009).

40. P. Radivojac *et al.*, A large-scale evaluation of computational protein function prediction. *Nat Methods* **10**, 221-227 (2013).

41. C. Zhang *et al.*, Protein Structure and Sequence Reanalysis of 2019-nCoV Genome Refutes Snakes as Its Intermediate Host and the Unique Similarity between Its Spike Protein Insertions and HIV-1. *J Proteome Res* **19**, 1351-1360 (2020).

42. E. Capriotti, P. Fariselli, R. Casadio, I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res* **33**, W306-310 (2005).

43. N. Tokuriki, D. S. Tawfik, Stability effects of mutations and protein evolvability. *Curr Opin Struct Biol* **19**, 596-604 (2009).

44. L. Quan, Q. Lv, Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics* **32**, 2936-2946 (2016).

45. E. Porta-Pardo, T. Hrabe, A. Godzik, Cancer3D: understanding cancer mutations through protein structures. *Nucleic Acids Res* **43**, D968-973 (2015).

46. D. E. Pires, D. B. Ascher, T. L. Blundell, mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* **30**, 335-342 (2014).

47. E. Porta-Pardo, A. Godzik, Mutation drivers of immunological responses to cancer. *Cancer Immunol Res* **4**, 789-798 (2016).

48. L. Sundaram *et al.*, Predicting the clinical impact of human mutation with deep neural networks. *Nat Genet* **50**, 1161-1170 (2018).

49. J. Woodard, C. Zhang, Y. Zhang, ADDRESS: A Database of Disease-associated Human Variants Incorporating Protein Structure and Folding Stabilities. *J Mol Biol* 10.1016/j.jmb.2021.166840, 166840 (2021).

50. A. Evers, G. Klebe, Successful virtual screening for a submicromolar antagonist of the neurokinin-1 receptor based on a ligand-supported homology model. *Journal of medicinal chemistry* **47**, 5381-5392 (2004).

51. G. Klebe, Virtual ligand screening: strategies, perspectives and limitations. *Drug discovery today* **11**, 580-594 (2006).

52. H. Zhou, J. Skolnick, FINDSITE(X): a structure-based, small molecule virtual screening approach with application to all identified human gpcrs. *Mol Pharm* **9**, 1775-1784 (2012).

53. A. Roy, Y. Zhang, Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. *Structure* **20**, 987-997 (2012).

54. Y. Y. Tseng, J. Dundas, J. Liang, Predicting protein function and binding profile via matching of local evolutionary and geometric surface patterns. *J Mol Biol* **387**, 451-464 (2009).

55. S. Vajda, F. Guarnieri, Characterization of protein-ligand interaction sites using experimental and computational methods. *Curr Opin Drug Disc* **9**, 354-362 (2006).

56. S. Choudhary, Y. S. Malik, S. Tomar, Identification of SARS-CoV-2 Cell Entry Inhibitors by Drug Repurposing Using in silico Structure-Based Virtual Screening Approach. *Front Immunol* **11**, 1664 (2020).

57. W. K. B. Chan, Y. Zhang, Virtual Screening of Human Class-A GPCRs Using Ligand Profiles Built on Multiple Ligand-Receptor Interactions. *J Mol Biol* **432**, 4872-4890 (2020).

58. I. D. Kuntz, Structure-based strategies for drug design and discovery. *Science* **257**, 1078-1082 (1992).

59. J. Drews, Drug discovery: a historical perspective. *Science* **287**, 1960-1964 (2000).

60. A. Evers, T. Klabunde, Structure-based drug discovery using GPCR homology modeling: Successful virtual screening for antagonists of the Alpha1A adrenergic receptor. *Journal of medicinal chemistry* **48**, 1088-1097 (2005).

61. S. Ekins, J. Mestres, B. Testa, In silico pharmacology for drug discovery: applications to targets and beyond. *British journal of pharmacology* **152**, 21-37 (2007).

62. Y. Shan *et al.*, How Does a Drug Molecule Find Its Target Binding Site? *Journal of the American Chemical Society* **133**, 9181-9183 (2011).

63. X. Han *et al.*, Discovery of ARD-69 as a Highly Potent Proteolysis Targeting Chimera (PROTAC) Degrader of Androgen Receptor (AR) for the Treatment of Prostate Cancer. *Journal of medicinal chemistry* 10.1021/acs.jmedchem.8b01631 (2019).

64. C. R. Darwin, *The Origin of Species* (John Murry, Landon, 1859).

65. N. K. Fox, S. E. Brenner, J. M. Chandonia, SCOPe: Structural Classification of Proteins--extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* **42**, D304-309 (2014).

66. Y. Zhang, I. A. Hubner, A. K. Arakaki, E. Shakhnovich, J. Skolnick, On the origin and highly likely completeness of single-domain protein structures. *Proc Natl Acad Sci U S A* **103**, 2605-2610 (2006).

67. Y. Zhang, J. Skolnick, The protein structure prediction problem could be solved using the current PDB library. *P Natl Acad Sci USA* **102**, 1029-1034 (2005).

68. L. Cao *et al.*, De novo design of picomolar SARS-CoV-2 miniprotein inhibitors. *Science* **370**, 426-431 (2020).

69. D.-A. Silva *et al.*, De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186-191 (2019).

70. A. Chevalier *et al.*, Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74-79 (2017).

71. J. Dou *et al.*, De novo design of a fluorescence-activating β-barrel. *Nature* **561**, 485-491 (2018).

72. C. E. Tinberg *et al.*, Computational design of ligand-binding proteins with high affinity and selectivity. *Nature* **501**, 212-216 (2013).

73. M. J. Lajoie *et al.*, Designed protein logic to target cells with precise combinations of surface antigens. *Science* **369**, 1637-1643 (2020).

74. S. R. Eddy, R. Durbin, RNA sequence analysis using covariance models. *Nucleic Acids Res* **22**, 2079-2088 (1994).

75. S. T. Wu, Y. Zhang, MUSTER: Improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* **72**, 547-556 (2008).

76. J. Soding, Protein homology detection by HMM-HMM comparison. *Bioinformatics* **21**, 951-960 (2005).

77. S. T. Wu, Y. Zhang, LOMETS: A local meta-threading-server for protein structure prediction. *Nucleic Acids Res* **35**, 3375-3382 (2007).

78. K. Ginalski, A. Elofsson, D. Fischer, L. Rychlewski, 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics* **19**, 1015-1018 (2003).

79. H. Park, F. DiMaio, D. Baker, The Origin of Consistent Protein Structure Refinement from Structural Averaging. *Structure* **23**, 1123-1128 (2015).

80. Y. Zhang, Progress and challenges in protein structure prediction. *Curr Opin Struc Biol* **18**, 342-348 (2008).

81. Y. Li, Zheng, W., Zhang, C., Bell, E., Huang, X., Pearce, R., Zhou, X., Zhang, Y. (2020) Protein 3D Structure Prediction by D-I-TASSER in CASP14. in *14th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction*.

82. J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, Critical assessment of methods of protein structure prediction (CASP)Round XII. *Proteins* **86**, 7-15 (2018).

83. Y. Zhang, A. Kolinski, J. Skolnick, TOUCHSTONE II: A new approach to ab initio protein structure prediction. *Biophys J* **85**, 1145-1164 (2003).

84. Y. F. Song *et al.*, High-Resolution Comparative Modeling with RosettaCM. *Structure* **21**, 1735-1742 (2013).

85. J. A. Cruz *et al.*, RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* **18**, 610-625 (2012).

86. Z. Miao *et al.*, RNA-Puzzles Round IV: 3D structure predictions of four ribozymes and two aptamers. *RNA* **26**, 982-995 (2020).

87. S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, S. R. Eddy, Rfam: an RNA family database. *Nucleic Acids Res* **31**, 439-441 (2003).

88. M. Rother, K. Rother, T. Puton, J. M. Bujnicki, ModeRNA: a tool for comparative modeling of RNA 3D structure. *Nucleic Acids Res* **39**, 4007-4022 (2011).

89. S. C. Flores, Y. Wan, R. Russell, R. B. Altman, "PREDICTING RNA STRUCTURE BY MULTIPLE TEMPLATE HOMOLOGY MODELING" in Biocomputing 2010. (WORLD SCIENTIFIC, 2009), doi:10.1142/9789814295291_0024

10.1142/9789814295291_0024, pp. 216-227.

90. J. U. Bowie, D. Eisenberg, An Evolutionary Approach to Folding Small Alpha-Helical Proteins That Uses Sequence Information and an Empirical Guiding Fitness Function. *P Natl Acad Sci USA* **91**, 4436-4440 (1994).

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

92. D. Xu, Y. Zhang, Toward optimal fragment generations for ab initio protein structure assembly. *Proteins* **81**, 229-239 (2013).

93. R. Das, D. Baker, Automated de novo prediction of native-like RNA tertiary structures. *Proc Natl Acad Sci U S A* **104**, 14664-14669 (2007).

94. A. M. Watkins, R. Rangan, R. Das, FARFAR2: Improved De Novo Rosetta Prediction of Complex Global RNA Folds. *Structure* **28**, 963-976.e966 (2020).

95. Y. Zhao *et al.*, Automated and fast building of three-dimensional RNA structures. *Sci Rep* **2**, 734 (2012).

96. M. Popenda *et al.*, Automated 3D structure composition for large RNAs. *Nucleic Acids Res* **40**, e112 (2012).

97. X. Xu, C. Zhao, S. J. Chen, VfoldLA: A web server for loop assembly-based prediction of putative 3D RNA structures. *J Struct Biol* **207**, 235-240 (2019).

98. W. Zheng *et al.*, Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins* **87**, 1149-1164 (2019).

99. C. X. Zhang, S. M. Mortuza, B. J. He, Y. T. Wang, Y. Zhang, Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. *Proteins* **86**, 136-151 (2018).

100. S. Ovchinnikov, H. Park, D. E. Kim, F. DiMaio, D. Baker, Protein structure prediction using Rosetta in CASP12. *Proteins* **86**, 113-121 (2018).

101. Z. Miao *et al.*, RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA* **23**, 655-672 (2017).

102. Z. Miao *et al.*, RNA-Puzzles Round II: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA* **21**, 1066-1084 (2015).

103. U. Gobel, C. Sander, R. Schneider, A. Valencia, Correlated mutations and residue contacts in proteins. *Proteins* **18**, 309-317 (1994).

104. S. R. Eddy, Multiple alignment using hidden Markov models. *Proc Int Conf Intell Syst Mol Biol* **3**, 114-120 (1995).

105. S. W. Lockless, R. Ranganathan, Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. *Science* **286**, 295-299 (1999).

106. M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, T. Hwa, Identification of direct residue contacts in protein-protein interaction by message passing. *P Natl Acad Sci USA* **106**, 67-72 (2009).

107. F. Morcos *et al.*, Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *P Natl Acad Sci USA* **108**, E1293-E1301 (2011).

108. I. Kass, A. Horovitz, Mapping pathways of allosteric communication in GroEL by analysis of correlated mutations. *Proteins* **48**, 611-617 (2002).

109. D. K. Y. Chiu, T. Kolodziejczak, Inferring Consensus Structure from Nucleic-Acid Sequences. *Comput Appl Biosci* **7**, 347-352 (1991).

110. M. Ekeberg, C. Lovkvist, Y. H. Lan, M. Weigt, E. Aurell, Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Phys Rev E* **87** (2013).

111. C. Baldassi *et al.*, Fast and Accurate Multivariate Gaussian Modeling of Protein Families: Predicting Residue Contacts and Protein-Interaction Partners. *Plos One* **9** (2014).

112. D. T. Jones, D. W. A. Buchan, D. Cozzetto, M. Pontil, PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* **28**, 184-190 (2012).

113. Y. Li, J. Hu, C. X. Zhang, D. J. Yu, Y. Zhang, ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks. *Bioinformatics* **35**, 4647-4655 (2019).

114. H. P. Sun, Y. Huang, X. F. Wang, Y. Zhang, H. B. Shen, Improving accuracy of protein contact prediction using balanced network deconvolution. *Proteins* **83**, 485-496 (2015).

115. R. Pearce, Y. Zhang, Toward the solution of the protein structure prediction problem. *J Biol Chem* 10.1016/j.jbc.2021.100870, 100870 (2021).

116. K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, Deep Residual Learning for Image Recognition. *Proc Cvpr Ieee* 10.1109/Cvpr.2016.90, 770-778 (2016).

117. Y. Li, C. Zhang, E. W. Bell, D. J. Yu, Y. Zhang, Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. *Proteins* **87**, 1082-1091 (2019).

118. A. W. Senior *et al.*, Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706-+ (2020).

119. J. Zhang, Y. Zhang, A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *Plos One* **5**, e15386 (2010).

120. J. Ingraham, A. J. Riesselman, C. Sander, D. S. Marks (2019) Learning Protein Structure with a Differentiable Simulator. in *International Conference on Learning Representations* (New Orleans, Louisiana, United States).

121. Y. Li *et al.* (2020) Protein 3D Structure Prediction by Zhang Human Group in CASP14. in *14th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction*.

122. R. Pearce, Y. Li, G. S. Omenn, Y. Zhang, Fast and accurate Ab Initio Protein structure prediction using deep learning potentials. *Plos Comput Biol* **18**, e1010539 (2022).

123. J. Jumper *et al.*, Highly accurate protein structure prediction with AlphaFold. *Nature* 10.1038/s41586-021-03819-2 (2021).

124. A. Vaswani *et al.*, Attention is All you Need. *ArXiv* **abs/1706.03762** (2017).

125. R. Pearce, Y. Zhang, Deep learning techniques have significantly impacted protein structure prediction and protein design. *Curr Opin Struc Biol* **68**, 194-207 (2021).

126. M. Baek *et al.*, Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871-876 (2021).

127. R. Pearce, G. S. Omenn, Y. Zhang, &lt;em&gt;De Novo&lt;/em&gt; RNA Tertiary Structure Prediction at Atomic Resolution Using Geometric Potentials from Deep Learning. *bioRxiv* 10.1101/2022.05.15.491755, 2022.2005.2015.491755 (2022).

128. B. Kuhlman *et al.*, Design of a novel globular protein fold with atomic-level accuracy. *Science* **302**, 1364-1368 (2003).

129. P. S. Huang *et al.*, RosettaRemodel: A Generalized Framework for Flexible Backbone Protein Design. *Plos One* **6** (2011).

130. P. S. Huang *et al.*, High thermodynamic stability of parametrically designed helical bundles. *Science* **346**, 481-485 (2014).

131. R. F. Alford *et al.*, The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *J Chem Theory Comput* **13**, 3031-3048 (2017).

132. X. Huang, R. Pearce, Y. Zhang, EvoEF2: accurate and fast energy function for computational protein design. *Bioinformatics* **36**, 1135-1142 (2020).

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

134. R. Pearce, X. Huang, D. Setiawan, Y. Zhang, EvoDesign: Designing Protein-Protein Binding Interactions Using Evolutionary Interface Profiles in Conjunction with an Optimized Physical Energy Function. *J Mol Biol* **431**, 2467-2476 (2019).

135. K. Y. Wei *et al.*, Computational design of closely related proteins that adopt two well-defined but structurally divergent folds. *P Natl Acad Sci USA* **117**, 7208-7215 (2020).

136. Z. B. Chen *et al.*, De novo design of protein logic gates. *Science* **368**, 78-+ (2020).

137. Z. B. Chen *et al.*, Programmable design of orthogonal protein heterodimers. *Nature* **565**, 106-+ (2019).

138. S. E. Boyken, De novo design of protein homo-oligomers with modular hydrogen-bond network-mediated specificity (vol 352, aag1318, 2016). *Science* **353**, 879-879 (2016).

139. J. Y. Dou *et al.*, De novo design of a fluorescence-activating beta-barrel. *Nature* **561**, 485-+ (2018).

140. A. Chevalier *et al.*, Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74-+ (2017).

141. D. A. Silva *et al.*, De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186-+ (2019).

142. F. Sesterhenn *et al.*, De novo protein design enables the precise induction of RSV-neutralizing antibodies. *Science* **368** (2020).

143. B. E. Correia *et al.*, Proof of principle for epitope-focused vaccine design. *Nature* **507**, 201-206 (2014).

144. F. Sesterhenn *et al.*, Boosting subdominant neutralizing antibody responses with a computationally designed epitope-focused immunogen. *PLoS biology* **17** (2019).

145. N. F. Polizzi, W. F. DeGrado, A defined structural unit enables de novo design of small-molecule-binding proteins. *Science* **369**, 1227-1233 (2020).

146. X. Huang, R. Pearce, Y. Zhang, De novo design of protein peptides to block association of the SARS-CoV-2 spike protein with human ACE2. *Aging (Albany NY)* **12**, 11263-11276 (2020).

147. P.-S. Huang *et al.*, De novo design of a four-fold symmetric TIM-barrel protein with atomic-level accuracy. *Nature Chemical Biology* **12**, 29-34 (2016).

148. D. Baker, What has de novo protein design taught us about protein folding and biophysics? *Protein Sci* **28**, 678-683 (2019).

149. I. Anishchenko *et al.*, De novo protein design by deep network hallucination. *Nature* **600**, 547-552 (2021).

150. J. Wang *et al.*, Scaffolding protein functional sites using deep learning. *Science* **377**, 387-394 (2022).

151. B. Huang *et al.*, A backbone-centred energy function of neural networks for protein design. *Nature* **602**, 523-528 (2022).

152. W. R. Taylor, A 'periodic table' for protein structures. *Nature* **416**, 657-660 (2002).

153. Z. Harteveld *et al.*, A generic framework for hierarchical <em>de novo</em> protein design. *bioRxiv* 10.1101/2022.04.07.487481, 2022.2004.2007.487481 (2022).

154. T. M. Jacobs *et al.*, Design of structurally distinct proteins using strategies inspired by evolution. *Science* **352**, 687-690 (2016).

155. S. L. Guffy, F. D. Teets, M. I. Langlois, B. Kuhlman, Protocols for Requirement-Driven Protein Design in the Rosetta Modeling Program. *J Chem Inf Model* **58**, 895-901 (2018).

156. N. Koga *et al.*, Principles for designing ideal protein structures. *Nature* **491**, 222-227 (2012).

157. X. Pan, T. Kortemme, Recent advances in de novo protein design: Principles, methods, and applications. *J Biol Chem* **296**, 100558 (2021).

158. W. Zhou, T. Smidlehner, R. Jerala, Synthetic biology principles for the design of protein with novel structures and functions. *FEBS Lett* **594**, 2199-2212 (2020).

159. P. S. Huang, S. E. Boyken, D. Baker, The coming of age of de novo protein design. *Nature* **537**, 320-327 (2016).

160. R. Dunbrack (2014) Template-based modeling assessment in CASP11. in *11th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction* (Riviera Maya, Mexico).

161. L. N. Kinch, W. Li, B. Monastyrskyy, A. Kryshtafovych, N. V. Grishin, Evaluation of free modeling targets in CASP11 and ROLL. *Proteins* **84 Suppl 1**, 51-66 (2016).

162. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

163. A. W. Senior *et al.*, Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706-710 (2020).

164. W. Zheng *et al.*, Protein structure prediction using deep learning distance and hydrogen-bonding restraints in CASP14. *Proteins* 10.1002/prot.26193 (2021).

165. W. Li *et al.*, TOUCHSTONEX: protein structure prediction with sparse NMR data. *Proteins* **53**, 290-306 (2003).

166. P. Barth, B. Wallner, D. Baker, Prediction of membrane protein structures with complex topologies using limited constraints. *Proc Natl Acad Sci U S A* **106**, 1409-1414 (2009).

167. S. Wu, Y. Zhang, A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics* **24**, 924-931 (2008).

168. W. Zheng *et al.*, LOMETS2: improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. *Nucleic Acids Res* **47**, W429-W436 (2019).

169. J. Xu, Y. Zhang, How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26**, 889-895 (2010).

170. Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702-710 (2004).

171. C. J. Williams *et al.*, MolProbity: More and better reference data for improved all-atom structure validation. *Protein Sci* **27**, 293-315 (2018).

172. B. Rost, C. Sander, R. Schneider, Redefining the goals of protein secondary structure prediction. *J Mol Biol* **235**, 13-26 (1994).

173. W. Zheng *et al.*, Folding non-homology proteins by coupling deep-learning contact maps with I-TASSER assembly simulations. *Cell Reports Methods* **1**, 100014 (2021).

174. J. G. Greener, S. M. Kandathil, D. T. Jones, Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints. *Nat Commun* **10** (2019).

175. L. Kinch *et al.*, CASP9 assessment of free modeling target predictions. *Proteins* **79 Suppl 10**, 59-73 (2011).

176. C. H. Tai, H. Bai, T. J. Taylor, B. Lee, Assessment of template-free modeling in CASP10 and ROLL. *Proteins* **82 Suppl 2**, 57-83 (2014).

177. A. Rives *et al.*, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv* 10.1101/622803, 622803 (2020).

178. R. Rao *et al.*, MSA Transformer. *bioRxiv* 10.1101/2021.02.12.430858, 2021.2002.2012.430858 (2021).

179. P. Yang, W. Zheng, K. Ning, Y. Zhang, Decoding microbiome and protein family linkage to improve protein structure prediction. *bioRxiv* 10.1101/2021.04.15.440088, 2021.2004.2015.440088 (2021).

180. C. Zhang, W. Zheng, S. M. Mortuza, Y. Li, Y. Zhang, DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics* **36**, 2105-2112 (2020).

181. Y. Li *et al.*, Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks. *Plos Comput Biol* **17**, e1008865 (2021).

182. H. Kamisetty, S. Ovchinnikov, D. Baker, Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proc Natl Acad Sci U S A* **110**, 15674-15679 (2013).

183. S. Seemayer, M. Gruber, J. Soding, CCMpred--fast and precise prediction of protein residue-residue contacts from correlated mutations. *Bioinformatics* **30**, 3128-3130 (2014).

184. S. T. Wu, Y. Zhang, ANGLOR: A Composite Machine-Learning Algorithm for Protein Backbone Torsion Angle Prediction. *Plos One* **3** (2008).

185. L. Armijo, Minimization of Functions Having Lipschitz Continuous First Partial Derivatives. *Pac J Math* **16**, 1-& (1966).

186. R. W. Yao, Y. Wang, L. L. Chen, Cellular functions of long noncoding RNAs. *Nat Cell Biol* **21**, 542-551 (2019).

187. M. J. Boniecki *et al.*, SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. *Nucleic Acids Res* **44**, e63 (2016).

188. K. Sato, M. Akiyama, Y. Sakakibara, RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat Commun* **12**, 941 (2021).

189. J. Singh, J. Hanson, K. Paliwal, Y. Zhou, RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat Commun* **10**, 5407 (2019).

190. S. Sun, W. Wang, Z. Peng, J. Yang, RNA inter-nucleotide 3D closeness prediction by deep residual neural networks. *Bioinformatics* **37**, 1093-1098 (2021).

191. T. Zhang *et al.*, RNAcmap: a fully automatic pipeline for predicting contact maps of RNAs by evolutionary coupling analysis. *Bioinformatics* **37**, 3494-3500 (2021).

192. R. Das, J. Karanicolas, D. Baker, Atomic accuracy in predicting and designing noncanonical RNA structure. *Nat Methods* **7**, 291-294 (2010).

193. J. L. Townshend Raphael *et al.*, Geometric deep learning of RNA structure. *Science* **373**, 1047-1051 (2021).

194. C. Zhang, Y. Zhang, A. M. Pyle, rMSA: database search and multiple sequence alignment generation to improve RNA struc-ture modeling. *ISMB*, In press (2022).

195. S. Gong, C. Zhang, Y. Zhang, RNA-align: quick and accurate alignment of RNA 3D structures based on size-independent TM-scoreRNA. *Bioinformatics* **35**, 4459-4461 (2019).

196. M. Meyer *et al.*, Speciation of a group I intron into a lariat capping ribozyme. *Proceedings of the National Academy of Sciences* **111**, 7659-7664 (2014).

197. A. Peselis, A. Serganov, Structural insights into ligand binding and gene expression control by an adenosylcobalamin riboswitch. *Nature Structural & Molecular Biology* **19**, 1182-1184 (2012).

198. N. B. Suslov *et al.*, Crystal structure of the Varkud satellite ribozyme. *Nature Chemical Biology* **11**, 840-846 (2015).

199. A. Ren, D. J. Patel, c-di-AMP binds the ydaO riboswitch in two pseudo-symmetry–related pockets. *Nature Chemical Biology* **10**, 780-786 (2014).

200. N. R. Coordinators, Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* **42**, D7-D17 (2014).

201. S. F. Altschul *et al.*, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389-3402 (1997).

202. T. J. Wheeler, S. R. Eddy, nhmmer: DNA homology search with profile HMMs. *Bioinformatics* **29**, 2487-2489 (2013).

203. S. E. Seemann, P. Menzel, R. Backofen, J. Gorodkin, The PETfold and PETcofold web servers for intra- and intermolecular structures of multiple RNA sequences. *Nucleic Acids Res* **39**, W107-W111 (2011).

204. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

205. P. Danaee *et al.*, bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Res* **46**, 5381-5394 (2018).

206. J. Towns *et al.*, XSEDE: Accelerating Scientific Discovery. *Comput Sci Eng* **16**, 62-74 (2014).

207. H. Jeong, S. P. Mason, A. L. Barabasi, Z. N. Oltvai, Lethality and centrality in protein networks. *Nature* **411**, 41-42 (2001).

208. A. Szilagyi, V. Grimm, A. K. Arakaki, J. Skolnick, Prediction of physical protein-protein interactions. *Phys Biol* **2**, S1-S16 (2005).

209. J. Karanicolas, B. Kuhlman, Computational design of affinity and specificity at protein-protein interfaces. *Curr Opin Struct Biol* **19**, 458-463 (2009).

210. G. Grigoryan, A. W. Reinke, A. E. Keating, Design of protein-interaction specificity gives selective bZIP-binding peptides. *Nature* **458**, 859-864 (2009).

211. S. J. Fleishman *et al.*, Computational Design of Proteins Targeting the Conserved Stem Region of Influenza Hemagglutinin. *Science* **332**, 816-821 (2011).

212. N. P. King *et al.*, Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science* **336**, 1171-1174 (2012).

213. D. Shultis, P. Mitra, X. Huang, J. Johnson, Z. Y, Changing the Apoptosis Pathway through Evolutionary Protein Design. *J Mol Biol*, in press (2019).

214. P. Mitra, D. Shultis, Y. Zhang, EvoDesign: De novo protein design based on structural and evolutionary profiles. *Nucleic Acids Res* **41**, W273-280 (2013).

215. P. Mitra *et al.*, An evolution-based approach to de novo protein design and case study on Mycobacterium tuberculosis. *Plos Comput Biol* **9**, e1003298 (2013).

216. P. Xiong, C. Zhang, W. Zheng, Y. Zhang, BindProfX: assessing mutation-induced binding affinity change by protein interface profiles with pseudo-counts. *J Mol Biol* **429**, 426-434 (2017).

217. J. Schymkowitz *et al.*, The FoldX web server: an online force field. *Nucleic Acids Res* **33**, W382-388 (2005).

218. M. Gao, J. Skolnick, iAlign: a method for the structural comparison of protein-protein interfaces. *Bioinformatics* **26**, 2259-2265 (2010).

219. J. R. Brender, Y. Zhang, Predicting the effect of mutations on protein-protein binding interactions through structure-based interface profiles. *Plos Comput Biol* **11**, e1004494 (2015).

220. Y. Zhang, J. Skolnick, TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic. Acids Res.* **33**, 2302-2309 (2005).

221. Y. Zhang, J. Skolnick, SPICKER: a clustering approach to identify near-native protein folds. *Journal of computational chemistry* **25**, 865-871 (2004).

222. J. E. Jones, On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature. *Proc. R. Soc. Lond. A* **106**, 441-462 (1924).

223. J. E. Jones, On the determination of molecular fields.—II. From the equation of state of a gas. *Proc. R. Soc. Lond. A* **106**, 463-477 (1924).

224. B. R. Brooks *et al.*, CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry* **4**, 187-217 (1983).

225. D. Sitkoff, K. A. Sharp, B. Honig, Accurate Calculation of Hydration Free Energies Using Macroscopic Solvent Models. *The Journal of Physical Chemistry* **98**, 1978-1988 (1994).

226. T. Kortemme, A. V. Morozov, D. Baker, An Orientation-dependent Hydrogen Bonding Potential Improves Prediction of Specificity and Structure for Proteins and Protein–Protein Complexes. *Journal of Molecular Biology* **326**, 1239-1259 (2003).

227. T. Lazaridis, M. Karplus, Effective energy function for proteins in solution. *Proteins: Structure, Function, and Bioinformatics* **35**, 133-152 (1999).

228. R. Guerois, J. E. Nielsen, L. Serrano, Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *Journal of Molecular Biology* **320**, 369-387 (2002).

229. J. Jankauskaite, B. Jimenez-Garcia, J. Dapkunas, J. Fernandez-Recio, I. H. Moal, SKEMPI 2.0: An updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* 10.1101/341735 (2018).

230. H. Y. Deng, Y. Jia, Y. Zhang, 3DRobot: automated generation of diverse and well-packed protein structure decoys. *Bioinformatics* **32**, 378-387 (2016).

231. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of State Calculations by Fast Computing Machines. *J Chem Phys* **21**, 1087-1092 (1953).

232. B. G. Pierce *et al.*, ZDOCK server: interactive docking prediction of protein-protein complexes and symmetric multimers. *Bioinformatics* **30**, 1771-1773 (2014).

233. E. Procko *et al.*, A computationally designed inhibitor of an Epstein-Barr viral Bcl-2 protein induces apoptosis in infected cells. *Cell* **157**, 1644-1656 (2014).

234. P. L. Kastritis, J. P. Rodrigues, G. E. Folkers, R. Boelens, A. M. Bonvin, Proteins feel more than they see: fine-tuning of binding affinity by properties of the non-interacting surface. *J Mol Biol* **426**, 2632-2652 (2014).

235. Y. Li, Y. Zhang, REMO: A new protocol to refine full atomic protein models from C-alpha traces by optimizing hydrogen-bonding networks. *Proteins* **76**, 665-676 (2009).

236. R. Yan, D. Xu, J. Yang, S. Walker, Y. Zhang, A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Sci Rep* **3**, 2619 (2013).

237. H. Chen, H. X. Zhou, Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Res* **33**, 3193-3199 (2005).

238. S. Wu, Y. Zhang, ANGLOR: a composite machine-learning algorithm for protein backbone torsion angle prediction. *Plos One* **3**, e3400 (2008).

239. W. Kabsch, C. Sander, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577-2637 (1983).

240. D. Shultis, G. Dodge, Y. Zhang, Crystal structure of designed PX domain from cytokine-independent survival kinase and implications on evolution-based protein engineering. *J Struct Biol* **191**, 197-206 (2015).

241. B. I. Dahiyat, C. A. Sarisky, S. L. Mayo, De novo protein design: towards fully automated sequence selection. *J Mol Biol* **273**, 789-796 (1997).

242. G. A. Lazar, J. R. Desjarlais, T. M. Handel, De novo design of the hydrophobic core of ubiquitin. *Protein Sci* **6**, 1167-1178 (1997).

243. G. Dantas, B. Kuhlman, D. Callender, M. Wong, D. Baker, A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. *J Mol Biol* **332**, 449-460 (2003).

244. E. Verschueren *et al.*, Protein design with fragment databases. *Curr Opin Struct Biol* **21**, 452-459 (2011).

245. H. Deng, Y. Jia, Y. Zhang, 3DRobot: automated generation of diverse and well-packed protein structure decoys. *Bioinformatics* **32**, 378-387 (2016).

246. C. B. Anfinsen, Principles that govern the folding of protein chains. *Science* **181**, 223-230 (1973).

247. H. Zhou, J. Skolnick, GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophys J* **101**, 2043-2052 (2011).

248. J. Park, K. Saitou, ROTAS: a rotamer-dependent, atomic statistical potential for assessment and prediction of protein structures. *BMC Bioinformatics* **15**, 307-307 (2014).

249. Y. R. Lin *et al.*, Control over overall shape and size in de novo designed proteins. *P Natl Acad Sci USA* **112**, E5478-E5485 (2015).

250. M. J. Abraham *et al.*, GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1-2**, 19-25 (2015).

251. M. Baek, D. Baker, Deep learning and protein structure modeling. *Nat Methods* **19**, 13-14 (2022).

252. V. B. Chen *et al.*, MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallogr D Biol Crystallogr* **66**, 12-21 (2010).

253. N. Fernandez-Fuentes, B. Oliva, A. Fiser, A supersecondary structure library and search algorithm for modeling loops in protein structures. *Nucleic Acids Res* **34**, 2085-2097 (2006).

254. N. Fernandez-Fuentes, J. M. Dybas, A. Fiser, Structural characteristics of novel protein folds. *Plos Comput Biol* **6**, e1000750 (2010).

255. S. Wu, Y. Zhang, Recognizing protein substructure similarity using segmental threading. *Structure* **18**, 858-867 (2010).

256. S. Wu, J. Skolnick, Y. Zhang, Ab initio modeling of small proteins by iterative TASSER simulations. *BMC Biology* **5**, 17 (2007).

257. D. Xu, Y. Zhang, Improving the physical realism and structural accuracy of protein models by a two-step atomic-level energy minimization. *Biophys J* **101**, 2525-2534 (2011).

258. R. Pearce, X. Huang, G. S. Omenn, Y. Zhang, De novo protein fold design through sequence-independent fragment assembly simulations. *Proceedings of the National Academy of Sciences* **120**, e2208275120 (2023).

259. R. Evans *et al.*, Protein complex prediction with AlphaFold-Multimer. *bioRxiv* 10.1101/2021.10.04.463034, 2021.2010.2004.463034 (2022).

260. Z. Lin *et al.*, Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv* 10.1101/2022.07.20.500902, 2022.2007.2020.500902 (2022).

261. C. Norn *et al.*, Protein sequence design by conformational landscape optimization. *Proc Natl Acad Sci U S A* **118** (2021).

262. J. L. Watson *et al.*, Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv* 10.1101/2022.12.09.519842, 2022.2012.2009.519842 (2022).

263. S. B. Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**, 443-453 (1970).

264. S. Henikoff, J. G. Henikoff, Amino-Acid Substitution Matrices from Protein Blocks. *P Natl Acad Sci USA* **89**, 10915-10919 (1992).

265. Z. Xiang, B. Honig, Extending the accuracy limits of prediction for side-chain conformations. *J Mol Biol* **311**, 421-430 (2001).

266. D. A. Case *et al.*, The Amber biomolecular simulation programs. *Journal of computational chemistry* **26**, 1668-1688 (2005).

267. M. Ulmke, H. Müller-Krumbhaar, Linear scaling of computer time with the inverse temperature for the grand canonical quantum Monte Carlo method. *Zeitschrift für Physik B Condensed Matter* **89**, 239-241 (1992).

268. I. Rozada, M. Aramon, J. Machta, H. G. Katzgraber, Effects of setting temperatures in the parallel tempering Monte Carlo algorithm. *Phys Rev E* **100**, 043311 (2019).

269. R. A. da Silva, L. Degrève, A. Caliri, LMProt: an efficient algorithm for Monte Carlo sampling of protein conformational space. *Biophys J* **87**, 1567-1577 (2004).

270. A. A. Vorobieva *et al.*, De novo design of transmembrane beta barrels. *Science* **371** (2021).

271. L. An, G. R. Lee, De Novo Protein Design Using the Blueprint Builder in Rosetta. *Current Protocols in Protein Science* **102**, e116 (2020).