END-TO-END DIFFERENTIABLE LEARNING OF PROTEIN STRUCTURE

Mohammed AlQuraishi[1,2]

[1] Laboratory of Systems Pharmacology, Harvard Medical School, Boston, MA 02115

[2] Department of Systems Biology, Harvard Medical School, Boston, MA 02115

Address correspondence to: alquraishi@hms.harvard.edu

ABSTRACT

Accurate prediction of protein structure is one of the central challenges of biochemistry. Despite significant progress made by co-evolution methods to predict protein structure from signatures of residue-residue coupling found in the evolutionary record, a direct and explicit mapping between protein sequence and structure remains elusive, with no substantial recent progress. Meanwhile, rapid developments in deep learning, which have found remarkable success in computer vision, natural language processing, and quantum chemistry raise the question of whether a deep learning based approach to protein structure could yield similar advancements. A key ingredient of the success of deep learning is the reformulation of complex, human-designed, multi-stage pipelines with differentiable models that can be jointly optimized end-to-end. We report the development of such a model, which reformulates the entire structure prediction pipeline using differentiable primitives. Achieving this required combining four technical ideas: (1) the adoption of a recurrent neural architecture to encode the internal representation of protein sequence, (2) the parameterization of (local) protein structure by torsional angles, which provides a way to reason over protein conformations without violating the covalent chemistry of protein chains, (3) the coupling of local protein structure to its global representation via recurrent geometric units, and (4) the use of a differentiable loss function to capture deviations between predicted and experimental structures. To our knowledge this is the first end-to-end differentiable model for learning of protein structure. We test the effectiveness of this approach using two challenging tasks: the prediction of novel protein folds without the use of co-evolutionary information, and the prediction of known protein folds without the use of structural templates. On the first task the model achieves state-of-the-art performance, even when compared to methods that rely on co-evolutionary data. On the second task the model is competitive with methods that use experimental protein structures as templates, achieving 3-7Å accuracy despite being template-free. Beyond protein structure prediction, end-to-end differentiable models of proteins represent a new paradigm for learning and modeling protein structure, with potential applications in docking, molecular dynamics, and protein design.

INTRODUCTION

Proteins are linear polymers comprised of asymmetrically repeating chemical units—the twenty naturally occurring amino acids—that fold into well-defined three dimensional structures based on the identity and ordering of their constituent units[1,2]. Because proteins carry out the bulk of molecular activity in the cell, the elucidation of the structures of all proteins is a foundational and longstanding problem in biochemistry. Experimental methods, namely x-ray crystallography, nuclear magnetic resonance, and cryo-electron

microscopy exist for determining the structures of proteins, but they are laborious and costly. This fact has spurred the development of computational methods to predict the structure of proteins from their amino acid sequence[3,4]. Such methods must contend with the staggeringly large space of possible mappings between protein sequence and structure, and the challenging physics of polymeric folding.

These methods fall into two broad categories. The first type attempts to build an explicit mapping between protein sequence and structure by defining a computational process that, acting upon the amino acid sequence of a protein, yields a three-dimensional structure. This category includes molecular dynamics[5] (MD), which use physics-based principles to simulate the dynamical trajectory of the folding process from an unstructured chain to the final, energetically stable, tertiary conformation, and fragment assembly[3] methods which use statistically-derived energy functions, in combination with a sampling process, to arrive at favorable three-dimensional conformations. While in principle these approaches can work for any protein, in practice MD is effective at *ab initio* folding for only very small proteins, and fragment assembly methods achieve high accuracy (3-5Å) only when operating in a template-guided mode, in which an experimental structure of a homologous protein is used to inform the prediction of the new protein.

The second category of methods do not build an explicit map between protein sequence and structure. Instead, they circumvent the problem by searching for signatures of co-evolving residues in large multiple sequence alignments of proteins evolutionarily related to the protein of interest. The co-evolution of two residues is often an indicator of their physical contact in the protein structure, and this information can be used to construct a tentative residue-residue contact map that constrains and guides structure prediction methods from the first category[6,7]. When a large and diverse set of homologous sequences exist for a protein—typically in the thousands to tens of thousands, although the statistical efficiency of the methodology is improving—co-evolution based methods can predict protein structures fairly accurately even when no homologous experimental structures exist. Due to their ability to generalize to previously unseen parts of protein structure space, co-evolution methods represent a genuine breakthrough in our ability to predict protein structure[8]. However, because these methods never derive an explicit sequence-structure map, they do not capture any information about the intrinsic relationship between sequence and structure. Practically, this means that co-evolution methods have little to say about proteins for which no sequence homologs exists, as may arise for newly sequenced bacterial taxa, or as is often necessary in the case of *de novo* protein design. Even in cases of reasonably well characterized proteins, co-evolution methods fundamentally operate on the level of protein families as opposed to individual proteins. This limits their ability to distinguish structural features between closely related proteins or to predict the structural consequences of minor sequence changes such as those

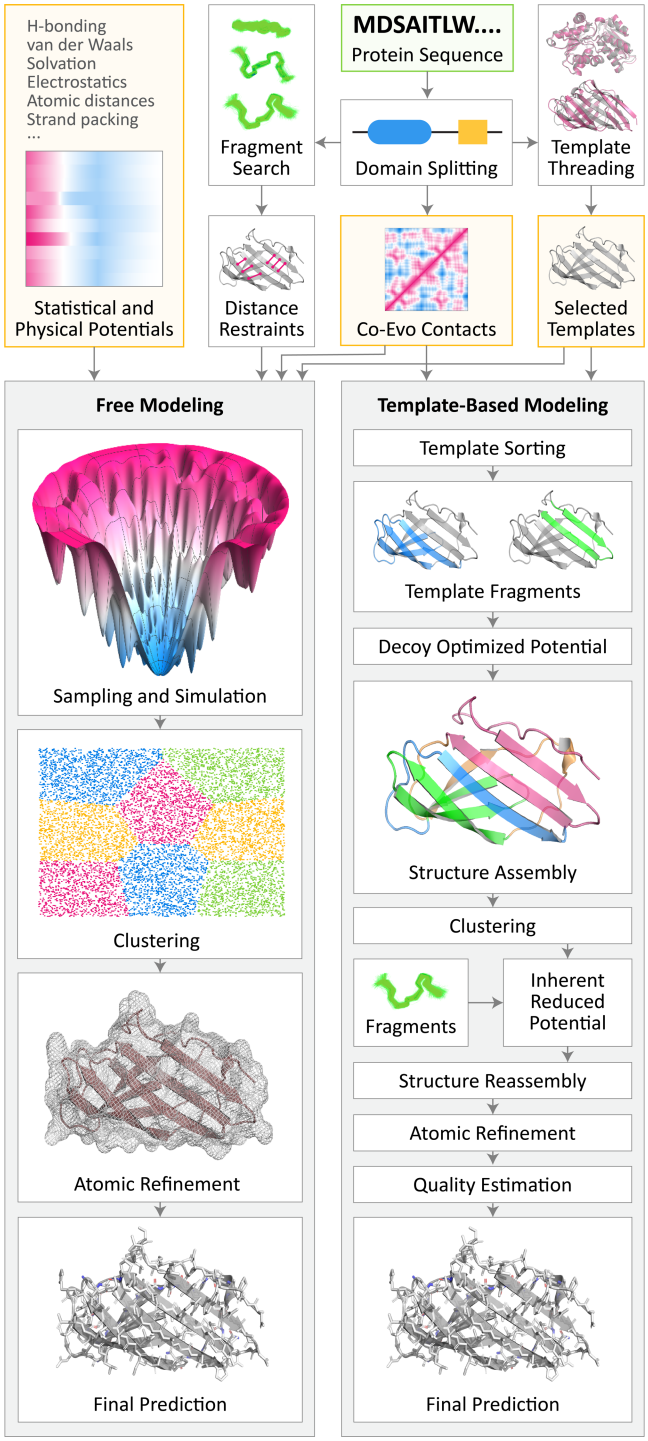

FIGURE 1: Conventional pipelines for protein structure prediction begin with the raw protein sequence (top, green box) and proceed through a series of human-designed modules. Typically, the constituent protein domains are first identified, and a multiple sequence alignment is built to predict co-evolutionary relationships. In *ab initio* or free modeling (left pipeline), used for the prediction of novel folds, a fragment library is searched to identify geometric distance restraints which, along with restraints derived from co-evolutionary information, are used to carry out a sequence of simulation, clustering, and atomic refinement to obtain the final tertiary structure. The simulation step uses a combination of statistical and physical potentials to minimize an energy function through an iterative sampling procedure. In template-based modeling (right pipeline), used for predicting structures with known protein folds, a search process is first carried out to identify explicit templates that can serve as a guide for structure prediction. Once a collection of templates is identified, a complex process of structure assembly is initiated where fragments are combined from multiple templates to assemble a complete structure, which is simultaneously optimized through a sampling procedure to minimize an empirical energy function, until a final prediction is predicted. Boxes highlighted in orange indicate sources of input information beyond the raw protein sequence, including prior knowledge about the physics of protein folding. Figure is based on the widely used I-Tasser and Quark pipelines.

induced by mutations, insertions, or deletions (co-evolution methods have shown promise however in predicting the functional consequences of mutations[9].)

We consider the possibility of solving the difficult problem addressed by the first category of approaches, i.e. an explicit sequence-to-structure map that does not rely on co-evolutionary information, by exploiting recent advances in machine learning, which has undergone a revolution driven by the reemergence of neural networks as building blocks in so-called end-to-end differentiable deep learning models[10]. End-to-end differentiability refers to the capability of jointly optimizing a complex multi-stage pipeline from input to output using a single procedure, which has resulted in unprecedented improvements in accuracy across a wide range of fields, with end-to-end differentiable models

of computer vision, speech recognition, and speech synthesis now approaching and exceeding human performance[10]. End-to-end differentiability is possible when every component of the learning pipeline is made differentiable, so that the basic rules of chain differentiation from calculus can be applied from output to input. The state of current protein structure prediction pipelines closely resembles those of computer vision and speech prior to deep learning: many complex stages, hand-engineered by human experts, each independently optimized (Figure 1). While deep learning has been applied to the problem of protein structure prediction, it has only been used as a component within existing pipelines, specifically for the inference of contacts from the co-evolutionary record[11,12] (second category of methods). These deep learning components still depend on and must interface with traditional protein structure prediction pipelines (first category), which include domain splitting, energy minimization, conformational sampling, geometric constraints, and more[13,14] (Figure 1). This limitation, which prohibits joint optimization of the entirety of the structure prediction pipeline, along with the use of exclusively off-the-shelf neural network components built and optimized for problems very distinct from protein structure, has so far prevented deep learning models from addressing the problem of building an explicit sequence-to-structure map.

The primary technical challenge to developing such a model lies in the necessity of rebuilding the entire structure prediction pipeline using differentiable primitives. The unique requirements of protein structure further necessitate bespoke components designed for the protein folding problem, which is not actively researched in the machine learning community. In this work, we introduce the building blocks necessary to construct an end-to-end differentiable model of protein structure, and test whether this approach can be made competitive with co-evolution and template-guided methods using two challenging tasks: (1) the prediction of new protein folds without using co-evolutionary information, and (2) the prediction of known protein folds without using experimental structures as templates. Surprisingly, we find that on the first problem, the new model can match and exceed the accuracy of co-evolution based methods, despite using only raw sequences and evolutionary profiles for individual residues, i.e. position-specific scoring matrices (PSSMs) that summarize the propensity of a residue to mutate to other amino acids, irrespective of other residues. On the second problem, the new model remains competitive, achieving accuracies of 3-7Å despite eschewing templates. Beyond the immediate application of protein structure prediction, end-to-end differentiable models of proteins represent a new paradigm for learning and modeling protein structure, with the potential to reformulate the representation and simulation of protein structure in fields as diverse as docking, molecular dynamics, and protein design.

RESULTS

Recurrent Geometric Networks

We cast protein structure prediction as a sequence-to-structure problem. The model takes as input the sequence of amino acids and PSSMs of a protein, one residue at a time, and outputs its three-dimensional structure. During training, when model parameters are being fitted, a loss value is also outputted, quantifying the deviation of the predicted structure from the experimental one and serving as a signal for the optimization algorithm to improve the model.

Our model is comprised of three primary stages—computation, geometry, and assessment—which we term a recurrent geometric network (RGN). The first stage is made up of computational units, standard in the field of neural networks (we use what are known as Long Short Term Memory units or LSTMs[15], although many other options are possible[16,17]), which, for each residue position, integrate information about the residue coming from its inputs, e.g. the amino acid present at that residue, along with information about all other residues encoded by the adjacent computational units (Figure 2). We augment the standard LSTM unit with specialized transformations that convert their raw outputs to angles (see supplementary material). By laying these computational units in a recurrent bidirectional topology, each unit receives information about the present residue and residues upstream and downstream all the way to the N- and C-terminus, respectively. In this way, the computation being performed for each residue can incorporate information across the entire protein, and by stacking computational units in multiple layers, the model is able to implicitly generate a multi-scale representation of the protein sequence.

We do not explicitly specify the computations to be carried out in these units, beyond what is stipulated by their functional form, namely that they compute affine transformations followed by sigmoidal nonlinearities (such computations are sufficiently general so as to form a universal Turing machine[18]). Instead, the computations are learned by optimizing the parameters of the RGN to accurately predict protein structures. This requires that the computational units generate outputs that can be interpreted as protein structures, which is the function of the second stage.

In general, the geometry of a protein backbone can be represented by three torsional angles ϕ, ψ, and ω that define the angles between successive planes spanned by the N, C^{α}, and C' protein backbone atoms[19]. While bond lengths and angles vary as well, their variation is sufficiently limited that they can be assumed fixed. Similar claims hold for side chains as well, although we restrict our attention to backbone structure. For each residue position, the first stage of the model outputs three numbers that correspond to the torsional angles for that residue. These angles are then fed into the second stage, which is comprised of geometric units that successively

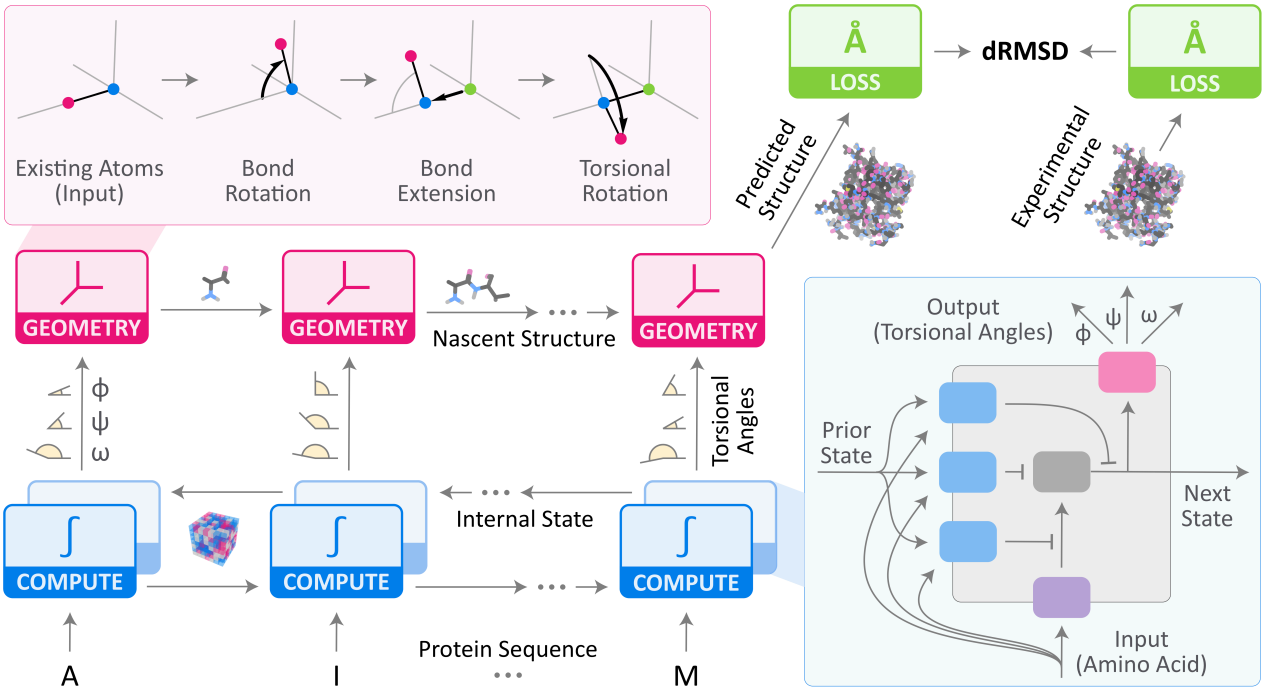

FIGURE 2: Overview of Recurrent Geometric Networks. The raw input sequence of a protein along with its PSSM is fed as input, one residue at a time, to the computational units of an RGN (bottom-left). These units integrate information about the amino acid residue at the current position with the internal states of other computational units operating on other residues. Based on this information, three torsional angles are outputted to the geometric units in the next layer, which sequentially translate these angles into the three-dimensional coordinates of the predicted protein. The predicted structure is then compared to the experimental structure, with the resulting dRMSD value quantifying the deviation between prediction and experiment, which is used as a signal to optimize the parameters of the RGN to make better predictions. **Top-Left Inset:** Internally, a geometric unit receives a partially completed protein backbone chain and a new set of torsional angles. Using this information, the geometric unit extends the nascent protein chain by an additional residue using a series of translations and rotations. **Bottom-Right Inset:** Internally, a computational unit takes information about the current residue as well as the state computed by other computational units flanking the residue, to compute a new state (purple unit). Gating units (blue) control whether the newly computed state replaces the existing state and whether the new state is transmitted as output. This architecture is based on the widely used LSTM unit, with an additional unit (pink) that converts the raw LSTM outputs into torsional angles.

translate this angular information into the three-dimensional backbone of the protein (Figure 2). Each geometric unit takes as input the three torsional angles for the present residue and the partially completed backbone resulting from the geometric unit upstream of it, and outputs a new backbone extended by one residue, which is then fed into the adjacent geometric unit downstream. The output of the final geometric unit is the completed three-dimensional structure of the protein. The geometric units employ translations and rotations computed using cross products, all of which are differentiable, to perform their function.

The output of the second stage is the final three-dimensional structure, which is sufficient if the model is being used purely for predictive purposes. For training however, a third stage is necessary to compute the deviation between the predicted structure and its experimental counterpart. While many metrics exist for

assessing the accuracy of a protein structure, for our purposes the metric must be differentiable. We use the distance-based root mean square deviation (dRMSD) which first computes the pairwise distances between all atoms in the predicted structure and all atoms in the experimental one (separately), and then computes the root mean square of the distance between these sets of distances. In addition to maintaining differentiability, the dRMSD metric is also multi-scale, capturing local and global aspects of protein structure. To train the model, the parameters of the computational units of the RGN are optimized so as to minimize the dRMSD between predicted and experimental structures using the standard techniques of backpropagation[20].

Accurate assessment of model error

Error assessment in machine learning-based models must be carefully performed to ensure that the test data used to assess model performance is sufficiently distinct from the training data used to fit the model, as overfitting can artificially inflate the accuracy of the model. This is particularly challenging in protein structure prediction due to the non-random nature of protein sequence space which makes it difficult to quarantine test data that is evolutionarily unrelated from the training data, and thus free of "information leakage" that could compromise the validity of the assessment[21]. Partly as a response to this well-documented problem, the semi-annual Critical Assessment of Protein Structure Prediction (CASP)[22] has been organized to assess computational methods in a wholly blind fashion, by testing predictors using sequences of solved structures that have not yet been publicly released. CASP organizers divide prediction targets into a free modeling (FM) category meant to test the prediction of novel structural topologies, and a template-based (TBM) category for testing the prediction of targets with known structural homologs in the Protein Data Bank[23] (PDB). This categorization provides an objective third-party delineation of the difficulty of protein targets, and the date marking available data preceding a CASP competition provides a natural demarcation for what sequences and structures can be used to constitute a training set. Based on these principles, we created a dataset based on the CASP11 competition in which the training data includes all sequences and structures available prior to the commencement of CASP11, and the test data is comprised of the structures used during the competition embargo period. The training data is further split into a large subset that can be used to optimize model parameters, and a small subset to assess model performance while training and optimize model hyperparameters (e.g. number of layers.) We use this data set for all our analyses.

RGNs predict new topologies with state-of-the-art accuracy without using co-evolutionary data

We sought to assess RGNs on a difficult task that has yet to be achieved consistently by any computational model: the prediction of new protein topologies without the aid of co-evolutionary information. Virtually all recent progress in protein structure prediction has come from the use of co-evolutionary data, making

successful prediction of protein structure without such information a strong validator of the model's independent capabilities and its potential to complement existing methods. We carry out our assessments on the FM structures used in CASP11, comparing methods using dRMSD and TM scores[24]. The dRMSD has the advantage of not requiring the predicted and experimental structures to be globally aligned, and is consequently able to detect regions of high local concordance even if the global structure is poorly aligned. On the other hand, it has the disadvantage of being sensitive to protein length, resulting in higher dRMSDs for larger proteins. The TM score has the advantage of being length-normalized, but the disadvantage of requiring a global alignment between structures. TM scores range from 0 to 1, with higher scores corresponding to better accuracy. A TM score of < 0.17 corresponds to a randomly chosen unrelated protein, and TM scores > 0.5 generally correspond to the same protein fold[25]. Table 1 compares the RGN model to the top five fully automated predictors in CASP11, known as "servers" in CASP terminology ("humans" are combinations of automated servers and manual processing by experts to improve structures—we do not compare against this group as all our processing is automated). Figure 3A shows the distribution of prediction accuracies over all servers during a CASP competition, and highlights where the RGN model lies. Figure 3B breaks down the accuracy per protein structure, comparing the RGN model with the best servers at CASP11. On dRMSD the RGN model outperforms all other methods, while it is tied with the best CASP11 server on TM score. The dRMSD is directly minimized by the RGN model and may thus give it an unfair advantage. Conversely, the TM score is one of the official metrics at CASP and is optimized for by CASP servers, potentially giving them an unfair advantage (TM scores were never used during training or validation of RGN models—they were only computed once, to calculate the numbers shown in Table 1.) Note that starting with the CASP11 experiment, co-evolution-based methods became available[26,27], which results in a substantial handicap against RGNs.

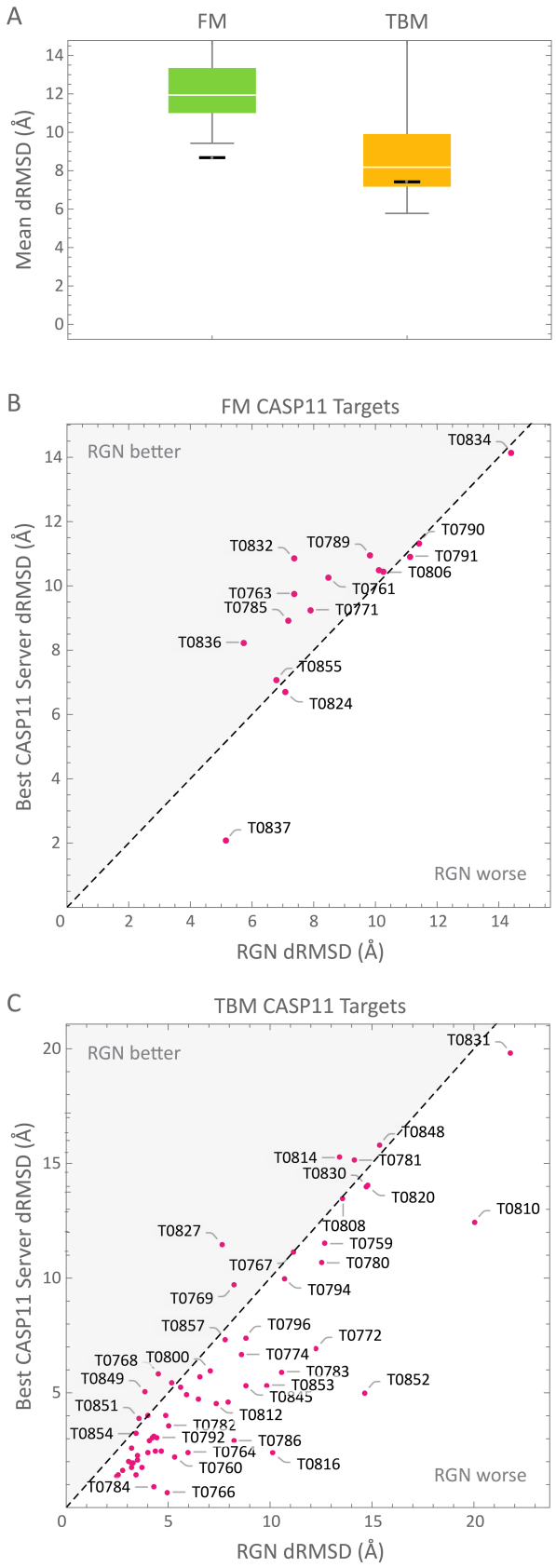

FIGURE 3: Results Overview. (A) The distribution of mean dRMSD (lower is better) achieved by servers at CASP11 is shown for the FM (green) and TBM / TBM-hard (orange) categories. All servers that predicted >95% of structures with >90% coverage were included (this included the best performing one). Performance of the RGN model on the same set of structures is shown with a thick black bar. On FM structures, the RGN model outperforms the best server, while it performs at around the top 25% quantile on TBM structures (wide white line corresponds to the median.) In all instances the RGN model did not have access to co-evolutionary data or make use of structural templates, unlike the top CASP11 servers. **(B)** Scatterplot comparing individual FM predictions made by the best FM server at CASP11 and RGN predictions. In all cases except one RGN predictions score roughly the same or better than the best server. **(C)** Scatterplot comparing individual TBM predictions made by the best TBM server at CASP11 and RGN predictions. In the majority of cases the RGN model does not perform as well as the best server, but the difference is generally around 1Å.

	FM category	
	dRMSD	**TM score**
RGN	8.7Å	0.28
CASP11 (1st)	9.4Å	0.28
CASP11 (2nd)	9.7Å	0.27
CASP11 (3rd)	10.9Å	0.25
CASP11 (4th)	11.7Å	0.23
CASP11 (5th)	13.5Å	0.22

TABLE 1: The average dRMSD (lower is better) and TM scores (higher is better) achieved by the RGN model and the top five servers at CASP11 in the FM category.

	TBM category	
	dRMSD	**TM score**
RGN	7.4Å	0.47
CASP11 (1st)	5.8Å	0.66
CASP11 (2nd)	6.0Å	0.66
CASP11 (3rd)	6.3Å	0.65
CASP11 (4th)	6.5Å	0.64
CASP11 (5th)	6.8Å	0.64

TABLE 2: The average dRMSD (lower is better) and TM scores (higher is better) achieved by the RGN model and the top five servers at CASP11 in the TBM or TBM hard categories.

RGNs predict known topologies with competitive accuracy without using templates

While prediction of novel topologies pushes the boundary of protein science, many biological applications of protein structure prediction revolve around proteins whose structure can be accurately predicted (~3-5Å) using template-based methods that use a structural homolog from the PDB as a guide. We sought to challenge RGNs to predict the structures of such proteins without using templates. If the model learns generalizable features of protein structure, it should be able to perform competitively when predicting TBM structures despite eschewing templates, as it would be operating in a densely sampled region of protein structure space (by definition, TBM proteins are ones with structural homologs in the PDB.) Nonetheless, the problem is extremely challenging as the use of templates provides a substantial advantage to template-based methods[28]. Table 2 compares the RGN model to the top five servers on CASP11, and Figure 3A shows the distribution of prediction accuracies over all predictors. Figure 3C breaks down the accuracy per protein structure, comparing RGNs with the best server at CASP11. Training data is the same as in the new topologies assessment, while assessment was carried out using the TBM (and TBM-hard) structures of each CASP competition. In the majority of cases the RGN model does not perform as well as the best CASP11 server, although the difference is generally around 1Å. Given that RGNs are not using experimental structures to guide predictions, while CASP11 servers are, this suggests that RGNs are learning a general model of protein structure, and their improved performance in the TBM category relative to the FM category may reflect the additional sampling of data in the TBM regions of protein space.

A representative sampling spanning the full quality spectrum of FM and TBM predictions is shown in Figure 4. We observe that while global topology is often, but not always, correctly predicted, secondary structure is often poorly predicted, sometimes extremely so (e.g. T0827). This likely reflects the fact that RGNs do not encode any biophysical priors on protein structure, including any knowledge relating to secondary structure elements such as α-helices or β-sheets. RGNs must learn everything from scratch, and do not utilize any form of post-hoc energy minimization. Future incorporation of such information may further improve accuracy.

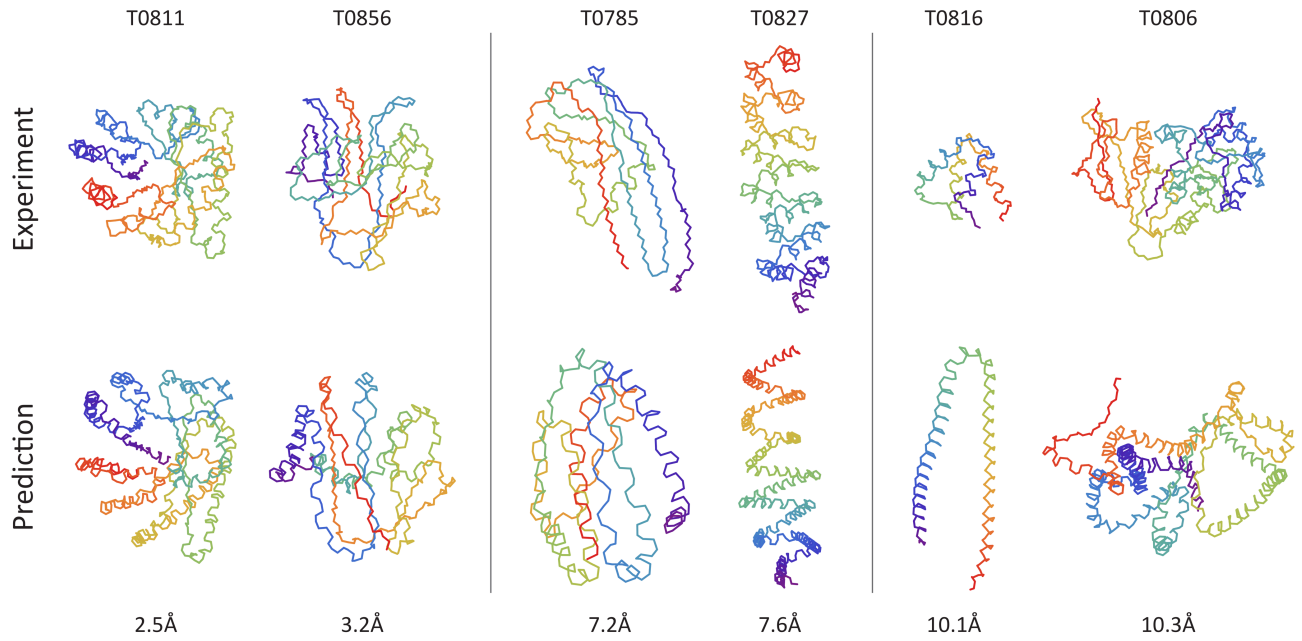

FIGURE 4: Representative Structures of RGN Predictions. Traces of backbone atoms of well (left), fairly (middle), and poorly (right) predicted structures representative of RGN performance are shown (bottom) along with their experimental counterparts from the PDB (top). The CASP11 identifier is displayed above each structure, and the dRMSD below. A color spectrum spans the length of the protein chain to aid in visualization.

RGN prediction accuracy is uniform along protein chain

Template-based methods for structure prediction use a specific structure (or set of structures) from the PDB as the basis for predicting a new protein sequence, based on detected homology between the template and the new sequence. The search for and detection of such structural templates is a difficult process, and modern prediction methods excel at this complex task. In practice, such predictions result in accurate regions where the structure of the new protein happens to fully coincide with the structure of the template, and inaccurate regions in areas where the template deviates from the new structure. For practical biological applications, the regions where the template and the new protein deviate are often the ones of highest interest, yet because a large part of the protein overlaps with the template, the error in these regions is masked by the large overlapping stretches, inflating overall accuracy[29–31]. We sought to test whether RGNs suffer from a similar limitation. For each protein domain in the TBM category (excluding TBM-hard entries which do not have good templates), we first compared the dRMSD of its best predicted structure (across all CASP11 servers) to the dRMSD of the best template found by the CASP11 organizers (all dRMSDs are computed against the experimental structures.) Such templates are found using a direct structure-to-structure comparison, and are not necessarily representative of the templates used by CASP11 servers, as they naturally do not have access to the target structures. Nonetheless, we found these two sets of dRMSDs to be correlated, with an R^2 value of 0.47. In contrast, when

comparing the dRMSDs of RGN predictions to the templates, we found the resulting R^2 value to be only 0.13. To directly assess the question of local structural fitness along the protein chain, we then split TBM proteins into short 15-residue fragments, and compared fragments from the best templates against the best CASP11 server predictions and RGN predictions (Figure 5). We used cRMSD against the experimental structural fragments as the metric, which requires that the fragments be structurally aligned and computes the direct RMSD between their atoms, because these fragments are short enough that their structural alignments are meaningful. To select the best server prediction and template for each domain, we used the global dRMSD as before. We again found CASP11 server predictions to be correlated with template quality with an R^2 value of 0.48, while RGN predictions showed virtually no correlation with an R^2 value of 0.07. Taken together these results suggest that template-based predictions are strongly dependent on the existence of high-quality templates, locally and globally, while RGN predictions are not.

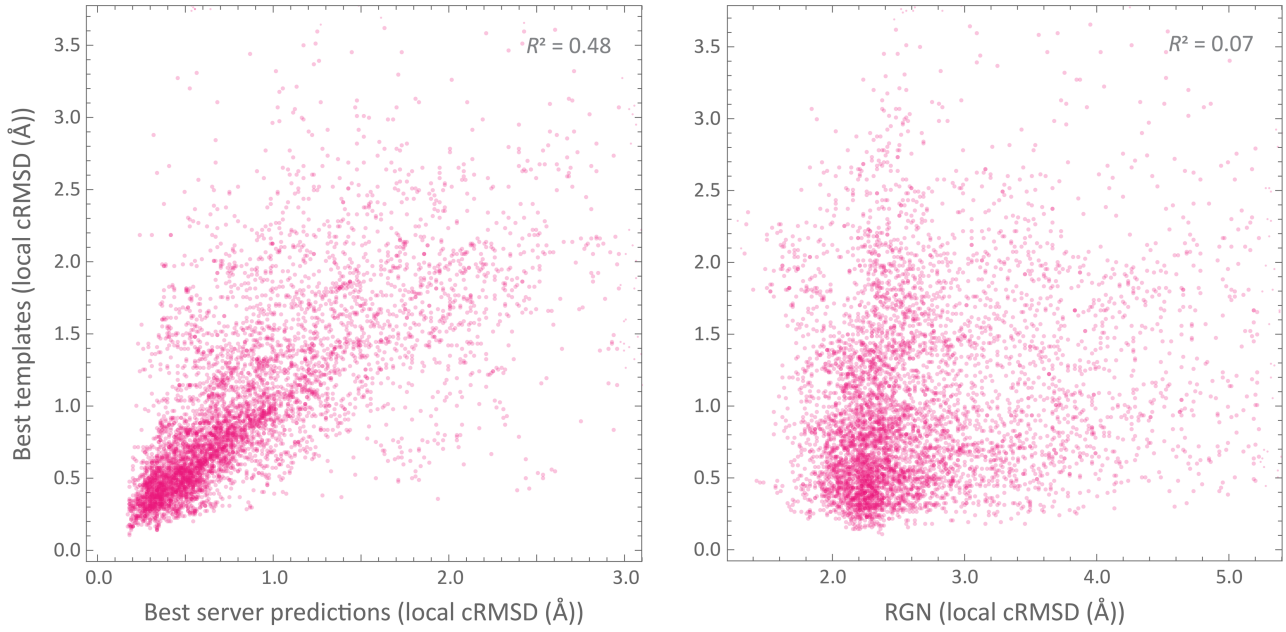

FIGURE 5: Scatterplot comparing the cRMSDs of small 15-residue fragments from TBM domains between the best templates found by CASP11 organizers and the best CASP11 server predictions (left) and RGN predictions (right). Only templates and predictions that cover > 85% of the full protein sequence are considered, and the selection of best templates and predictions is based on global dRMSD with respect to the experimental structures.

DISCUSSION

RGNs simplify prediction pipeline and increase its speed by several orders of magnitude

Traditional protein structure prediction pipelines are extremely complex (Figure 1). They begin by processing the input sequence to detect structural domains that can be independently modelled, and then run a series of algorithms to predict sequence characteristics such as propensity for secondary structure formation, solvent

accessibility, and disordered regions. In co-evolutionary methods, a multiple sequence alignment is used to predict a map of intra-protein residue contacts, and in template-based methods, the PDB is searched for structural templates that can act as the basis for prediction. All these sources of information are then converted into geometric constraints to guide the energy minimization and conformation sampling process, where a large library of protein fragments, guided by statistical analysis, are randomly swapped in and out of putative structures to minimize an expertly-derived energy model of protein folding. Depending on the complexity of the pipeline, this process consumes hours to days, and the codebase can span millions of lines of code as in the case of the leading Rosetta framework[32].

In contrast, RGNs are much simpler. The inputs are the raw protein sequence and its associated PSSM. The entire pipeline consists of a single end-to-end differentiable model comprised of computational and geometric units during prediction, and a loss unit during training. Instead of sampling millions of protein conformations, RGNs make predictions with a single pass, effectively folding the energy minimization and sampling process into the structural "reasoning" performed by the computational units. The model used in this paper is comprised of only a few thousand lines of code, which in addition to greatly simplifying the prediction pipeline, also results in dramatically increased speeds (Table 3). RGNs make very different trade-offs from conventional prediction pipelines. Because they are learned from scratch, training time can take weeks to months. However, once trained, RGNs make predictions in milliseconds, enabling entirely new uses for structure prediction such as docking and virtual screening. For instance, a ligand-aware version of RGNs could potentially output one or more protein conformations in response to distinct ligand poses, taking into account the flexibility of the protein chain and the location and orientation of the ligand. The speed with which protein conformations can be sampled makes this a realistic possibility for virtual screening, unlike traditional pipelines whose use would be prohibitive. Recent advances in machine learning have further enabled generative models of structured objects such as images[33] and DNA[34] using generative adversarial networks[35] and variational autoencoders[36,37]. Incorporating RGNs into such models could enable sampling of viable protein conformations, speeding up MD simulations.

Model	Prediction Speed	Training Time
Rosetta[27], I-Tasser[13], Quark[14]	hours to days	N/A
Raptor X[11], DeepContact[12] + CONFOLD[38]	hours to day	hours
Recurrent Geometric Networks	milliseconds	weeks to months

TABLE 3: Approximate speeds for prediction and training of various structure prediction approaches are shown. The top row corresponds to the most complex and established set of methods, which rely heavily on simulation and sampling, and typically have only a minimal learning component. The second row corresponds to co-evolution-based contact prediction methods, which rely on a learning procedure, plus the CONFOLD method to convert the predicted contact maps into tertiary structures.

RGNs learn a multi-scale representation of protein sequence

A persistent limitation of methods that build explicit mappings between sequence and structure, including MD and fragment assembly methods, is their reliance on predetermined energy models that do not permit substantial learning from data. They also rely on single scale representation, typically operating on the atomic or residue level—in some cases on secondary structure—and are thus unable to form a multi-scale representation of protein sequence. Such a representation could capture the sequence-structure motifs that have arisen during the course of evolution and that span a handful of residues all the way to entire domains[39,40]. Unlike this category of methods, co-evolution methods have leveraged data learning and multi-scale neural network architectures to build hierarchical representations of protein co-evolutionary couplings, and this has resulted in substantially improved performance[11,12]. RGNs bridge this gap by simultaneously building an explicit sequence-to-structure map and by being learnable and multi-scale. Through their recurrent architecture, RGNs are able to model long protein sequence fragments and discover higher-order relationships between these fragments. As additional structural and sequence data become available, and as new recurrent architectures emerge that are able to capture even longer range interactions than LSTMs, RGNs can automatically learn to improve their performance, while implicitly capturing sequence-structure relationships that may be uncovered using neural network probing techniques[41–45].

RGNs operate on three parameterizations of protein structure

The RGN multi-stage architecture results in three distinct parameterizations of protein structure upon which the model can operate. The first is torsional, capturing angular relationships between adjacent residues and can be thought of as local. The advantage of this parameterization is that it virtually guarantees that resulting proteins are locally correct, particularly as bond lengths and angles are held fixed (and thus are always biophysical), and torsional angles are the immediate outputs of the computational units. The geometric units then build a second parameterization of protein structure in terms of the absolute Cartesian coordinates of protein atoms. Such a parameterization is useful for immediately revealing features that rely on the coordination of multiple atoms in absolute space, such as the catalytic triad of an enzyme's active site. Even if these atoms are widely distributed along the protein chain, once they are brought together in three-dimensional space, their coordination would be evident in the Cartesian parameterization. Although RGNs do not currently take advantage of this, it is possible to use suitable neural network architectures, such as 3D convolutional networks, to operate on this parameterization and directly move atoms in absolute space. While an unconstrained version of this approach is unlikely to yield physically meaningful structures, the direct coupling of the first and second stages may yield superior results if this approach is used to refine the placement of atoms. Finally, the third parameterization is constructed in the dRMSD loss stage, which computes pairwise distances between all atoms within the

structure. This distance-based parameterization, effectively reducing a structure to a matrix of distances, is simultaneously local and global. It is useful as the error signal for optimization, as we have used it, but may also be extended to incorporate prior knowledge that can be suitably expressed in terms of atomic distances. This includes prior physical knowledge, such as electrostatic effects (e.g. Coulombic potential), as well as prior statistical knowledge, such as evolutionary couplings. RGNs thus provide multiple points of entry both for incorporating additional information as well as for refining and operating on protein structure.

Immediate extensions

RGNs permits simple extensions that may improve their performance and broaden their applicability. While we presented a minimal version of the RGN model to focus on its core competency, it is almost entirely complementary to existing approaches and can be easily integrated with them. In addition to incorporating co-evolutionary information, both as priors to the distance-based parameterization and also as raw inputs, RGNs can easily incorporate templates as well, by using existing template finding methods and supplying the selected templates, perhaps with a confidence score, as input to the RGN. An opposing direction is to further limit the inputs to RGNs, by jettisoning the use of PSSMs and requiring it to operate strictly on raw protein sequences. This would broaden the method's applicability and effectiveness in protein design and variant prediction. Finally, while we restricted our attention to protein backbone prediction, RGNs can be extended to predict side-chain conformations, in effect creating a branched curve structure in lieu of the single linear curve that the model currently predicts. The generality of the proposed model, coupled with the central role that protein structure representation plays in all existing biomolecular modeling pipelines, suggests that moving forward RGNs will have a central role to play in the computational modeling of biomolecules.

ACKNOWLEDGMENTS

We are indebted to Peter Sorger for his mentorship and support. We thank Jasper Snoek and Adrian Jinich for their editorial comments and many helpful discussions, and Uraib Aboudi, Ramy Arnaout, Karen Sachs, and Nazim Bouatta for their insightful feedback. We also thank Martin Steinegger and Milot Mirdita for their help with using the HHblits and MMseqs2 packages, Sergey Ovchinnikov for his help with metagenomics sequences, Andriy Kryshtafovych for his help with CASP structures, Sean Eddy for his help with using the JackHMMer package, and Raffaele Potami, Amir Karger, and Kristina Holton for their help with using the HPC resources at Harvard Medical School. This work was supported by NIGMS Grant P50GM107618. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs used for this research.

REFERENCES

1. Dill, K. A. Dominant forces in protein folding. *Biochemistry (Mosc.)* **29,** 7133–7155 (1990).
2. Branden, C. & Tooze, J. *Introduction to Protein Structure*. (Garland Science, 1999).
3. Gajda, M. J., Pawlowski, M. & Bujnicki, J. M. Protein Structure Prediction: From Recognition of Matches with Known Structures to Recombination of Fragments. in *Multiscale Approaches to Protein Modeling* (ed. Kolinski, A.) 231–254 (Springer New York, 2011).
4. Gajda, M. J., Pawlowski, M. & Bujnicki, J. M. *Multiscale Approaches to Protein Modeling*. (Springer New York, 2011).
5. Marx, D. & Hutter, J. *Ab initio molecular dynamics: basic theory and advanced methods*. (Cambridge University Press, 2012).
6. Marks, D. S. *et al.* Protein 3D structure computed from evolutionary sequence variation. *PloS One* **6,** e28766 (2011).
7. Hopf, T. A. *et al.* Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife* **3,** (2014).
8. Juan, D. de, Pazos, F. & Valencia, A. Emerging methods in protein co-evolution. *Nat. Rev. Genet.* **14,** 249–261 (2013).
9. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35,** 128–135 (2017).
10. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).
11. Wang, S., Sun, S., Li, Z., Zhang, R. & Xu, J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *bioRxiv* 073239 (2016). doi:10.1101/073239
12. Liu, Y., Palmedo, P., Ye, Q., Berger, B. & Peng, J. Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks. *Cell Syst.* **0,** (2017).
13. Yang, J. *et al.* The I-TASSER Suite: protein structure and function prediction. *Nat. Methods* **12,** 7–8 (2015).
14. Xu, D. & Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins* **80,** 1715–1735 (2012).
15. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9,** 1735–1780 (1997).
16. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. & Schmidhuber, J. LSTM: A Search Space Odyssey. *ArXiv150304069 Cs* (2015).
17. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv14123555 Cs* (2014).
18. Siegelmann, H. T. & Sontag, E. D. On the Computational Power of Neural Nets. *J. Comput. Syst. Sci.* **50,** 132–150 (1995).
19. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* **7,** 95–99 (1963).
20. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (The MIT Press, 2016).
21. Orlando, G., Raimondi, D. & Vranken, W. F. Observation selection bias in contact prediction and its implications for structural bioinformatics. *Sci. Rep.* **6,** 36679 (2016).
22. Moult, J., Pedersen, J. T., Judson, R. & Fidelis, K. A large-scale experiment to assess protein structure prediction methods. *Proteins Struct. Funct. Bioinforma.* **23,** ii–iv (1995).
23. Bernstein, F. C. *et al.* The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.* **112,** 535–542 (1977).

24. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins Struct. Funct. Bioinforma.* **57,** 702–710 (2004).
25. Xu, J. & Zhang, Y. How significant is a protein structure similarity with TM-score = 0.5? *Bioinforma. Oxf. Engl.* **26,** 889–895 (2010).
26. Kryshtafovych, A., Monastyrskyy, B. & Fidelis, K. CASP11 statistics and the prediction center evaluation system. *Proteins Struct. Funct. Bioinforma.* **84,** 15–19 (2016).
27. Ovchinnikov, S. *et al.* Improved de novo structure prediction in CASP11 by incorporating coevolution information into Rosetta. *Proteins Struct. Funct. Bioinforma.* **84,** 67–75 (2016).
28. Zhou, Y., Duan, Y., Yang, Y., Faraggi, E. & Lei, H. Trends in template/fragment-free protein structure prediction. *Theor. Chem. Acc.* **128,** 3–16 (2010).
29. Perez, A., Morrone, J. A., Brini, E., MacCallum, J. L. & Dill, K. A. Blind protein structure prediction using accelerated free-energy simulations. *Sci. Adv.* **2,** e1601274 (2016).
30. Contreras-Moreira, B., Ezkurdia, I., Tress, M. L. & Valencia, A. Empirical limits for template-based protein structure prediction: the CASP5 example. *FEBS Lett.* **579,** 1203–1207 (2005).
31. Dill, K. A. & MacCallum, J. L. The Protein-Folding Problem, 50 Years On. *Science* **338,** 1042–1046 (2012).
32. Leaver-Fay, A. *et al.* ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* **487,** 545–574 (2011).
33. Karras, T., Aila, T., Laine, S. & Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *ArXiv171010196 Cs Stat* (2017).
34. Killoran, N., Lee, L. J., Delong, A., Duvenaud, D. & Frey, B. J. Generating and designing DNA with deep generative models. *ArXiv171206148 Cs Q-Bio Stat* (2017).
35. Goodfellow, I. J. *et al.* Generative Adversarial Networks. *ArXiv14062661 Cs Stat* (2014).
36. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *ArXiv13126114 Cs Stat* (2013).
37. Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv14014082 Cs Stat* (2014).
38. Adhikari, B., Bhattacharya, D., Cao, R. & Cheng, J. CONFOLD: Residue-residue contact-guided ab initio protein folding. *Proteins* **83,** 1436–1449 (2015).
39. Alva, V., Söding, J. & Lupas, A. N. A vocabulary of ancient peptides at the origin of folded proteins. *eLife* **4,** e09410 (2015).
40. Ponting, C. P. & Russell, R. R. The Natural History of Protein Domains. *Annu. Rev. Biophys. Biomol. Struct.* **31,** 45–71 (2002).
41. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *ArXiv13126034 Cs* (2013).
42. Alain, G. & Bengio, Y. Understanding intermediate layers using linear classifier probes. *ArXiv161001644 Cs Stat* (2016).
43. Koh, P. W. & Liang, P. Understanding Black-box Predictions via Influence Functions. *ArXiv170304730 Cs Stat* (2017).
44. Nguyen, A., Yosinski, J. & Clune, J. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. *ArXiv160203616 Cs* (2016).
45. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. in *PMLR* 3145–3153 (2017).

SUPPLEMENTARY MATERIAL

Model

We featurize a protein of length L as a sequence of vectors $(x_1, \cdots, x_L)$ where $x_t \in \mathbb{R}^d$ for all t. The dimensionality d is 41, where 20 dimensions are used as a one-hot indicator of the amino acid residue at a given position, another 20 dimensions are used for the PSSM of that position, and 1 dimension is used to encode the information content of the position. The PSSM values are sigmoid transformed to lie between 0 and 1. The sequence of input vectors are fed to an LSTM, whose basic formulation is described by the following set of equations.

$$
\begin{aligned}
i_t &= \sigma(W_i[x_t, h_{t-1}] + b_i) \\
f_t &= \sigma\big(W_f[x_t, h_{t-1}] + b_f\big) \\
o_t &= \sigma(W_o[x_t, h_{t-1}] + b_o) \\
\tilde{c}_t &= tanh(W_c[x_t, h_{t-1}] + b_c) \\
c_t &= i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
$$

W_i, W_f, W_o, W_c are weight matrices, b_i, b_f, b_o, b_c are bias vectors, h_t and c_t are the hidden and memory cell state for residue t, respectively, and $\odot$ is element-wise multiplication. We use two LSTMs, running independently in opposite directions (1 to L and L to 1), to output two hidden states $h_t^{(f)}$ and $h_t^{(b)}$ for each residue position t corresponding to the forward and backward directions. Depending on the RGN architecture, these two hidden states are either the final outputs states or they are fed as inputs into one or more LSTM layers.

The outputs from the last LSTM layer form a sequence of a concatenated hidden state vectors $\left(\left[h_1^{(f)}, h_1^{(b)}\right], \cdots, \left[h_L^{(f)}, h_L^{(b)}\right]\right)$. Each concatenated vector is then fed into an angularization layer described by the following set of equations:

$$
\begin{aligned}
p_t &= softmax\left(W_\varphi \left[h_t^{(f)}, h_t^{(b)}\right] + b_\varphi\right) \\
\varphi_t &= arg\big(p_t \, exp(i\Phi)\big)
\end{aligned}
$$

W_φ is a weight matrix, b_φ is a bias vector, Φ is a learned alphabet matrix, and arg is the complex-valued argument function. Exponentiation of the complex-valued matrix $i\Phi$ is performed element-wise. The Φ matrix

defines an alphabet of size m whose letters correspond to triplets of torsional angles defined over the 3-torus. The angularization layer interprets the LSTM hidden state outputs as weights over the alphabet, using them to compute a weighted average of the letters of the alphabet (independently for each torsional angle) to generate the final set of torsional angles $\varphi_t \in S^1 \times S^1 \times S^1$ for residue t (we are overloading the standard notation for protein backbone torsional angles, with φ_t corresponding to the (ψ, φ, ω) triplet). Note that φ_t may be alternatively computed using the following equation, where the trigonometric operations are performed element-wise:

$$\varphi_t = atan2\big(p_t\ sin(\Phi), p_t\ cos(\Phi)\big)$$

The resulting sequence of torsional angles $(\varphi_1, \cdots, \varphi_L)$ is then fed sequentially, along with the coordinates of the last three atoms of the nascent protein chain $(c_1, \cdots, c_{3t})$, into recurrent geometric units that convert this sequence into 3D Cartesian coordinates, with three coordinates resulting from each residue, corresponding to the N, C^{α}, and C' backbone atoms. Multiple mathematically-equivalent formulations exist for this transformation; we adopt one based on the Natural Extension Reference Frame ([REF]), described by the following set of equations:

$$\tilde{c}_k = r_{k\ mod\ 3} \begin{bmatrix} cos(\theta_{k \bmod 3}) \\ cos\big(\varphi_{\lfloor k/3 \rfloor, k \bmod 3}\big)\ sin(\theta_{k \bmod 3}) \\ sin\big(\varphi_{\lfloor k/3 \rfloor, k \bmod 3}\big)\ sin(\theta_{k \bmod 3}) \end{bmatrix}$$

$$m_k = c_{k-1} - c_{k-2}$$

$$n_k = m_{k-1} \times \widehat{m_k}$$

$$M_k = [\widehat{m_k}, \widehat{n_k} \times \widehat{m_k}, \widehat{n_k}]$$

$$c_k = M_k \tilde{c}_k + c_{k-1}$$

Where r_k is the length of the bond connecting atoms $k-1$ and k, θ_k is the bond angle formed by atoms $k-2, k-1$, and k, $\varphi_{\lfloor k/3 \rfloor, k \bmod 3}$ is the predicted torsional angle formed by atoms $k-2$ and $k-1$, c_k is the position of the newly predicted atom k, $\widehat{m}$ is the unit-normalized version of m, and $\times$ is the cross product. Note that k indexes atoms 1 through $3L$, since there are three backbone atoms per residue. For each residue t we compute c_{3t-2}, c_{3t-1}, and c_{3t} using the three predicted torsional angles of residue t, specifically $\varphi_{t,j} = \varphi_{\lfloor \frac{3t}{3} \rfloor, (3t+j) \bmod 3}$ for $j = \{0,1,2\}$. The bond lengths and angles are fixed, with three bond lengths (r_0, r_1, r_2) corresponding to N-C^{α}, C^{α}-C', and C'-N, and three bond angles $(\theta_0, \theta_1, \theta_2)$ corresponding to N-C^{α}-C', C^{α}-C'-N, and C'-N-C^{α}. As there are only three unique values we have $r_k = r_{k \bmod 3}$ and $\theta_k = \theta_{k \bmod 3}$. In practice we employ a modified version of the above equations which enable much higher computational efficiency, described in [REF].

The resulting sequence $(c_1, \cdots, c_{3L})$ fully describes the protein backbone chain structure and is the model's final predicted output. For training purposes a loss is necessary to optimize model parameters. We use the $dRMSD$ metric as it is differentiable and captures both local and global aspects of protein structure. It is defined by the following set of equations:

$$\tilde{d}_{j,k} = \left\| c_j - c_k \right\|_2$$

$$d_{j,k} = \tilde{d}_{j,k}^{(\mathrm{exp})} - \tilde{d}_{j,k}^{(\mathrm{pred})}$$

$$dRMSD = \frac{\|D\|_2}{L(L-1)}$$

Where $\{d_{j,k}\}$ are the elements of matrix D, and $\tilde{d}_{j,k}^{(\mathrm{exp})}$ and $\tilde{d}_{j,k}^{(\mathrm{pred})}$ are computed using the coordinates of the experimental and predicted structures, respectively. In effect, the $dRMSD$ computes the ℓ_2-norm of the distances over distances, by first computing the pairwise distances between all atoms in both the predicted and experimental structures individually, and then computing the distances between those distances.