Deep learning-based prediction of protein structure using learned representations of multiple sequence alignments

Shaun M Kandathil, Joe G Greener, Andy M Lau and David T Jones

Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK
Address for correspondence: d.t.jones@ucl.ac.uk

Abstract

The use of amino acid covariation and other sequence-based features as inputs to deep learning-based predictors of contacts and distances in proteins is now commonplace. The prediction process usually begins by constructing a multiple sequence alignment (MSA) containing homologues of the target protein. The most successful approaches combine large feature sets derived from MSAs, meaning that considerable computational effort is spent deriving these input features. We present a method that greatly reduces the amount of preprocessing required for a target MSA, making the predictor faster to run and easier to install and use. Our approach constructs a directly learned representation of all the sequences in an MSA, starting from a one-hot encoding of the sequences. The learned representation is then used as the input to a ResNet, the latter being the now-standard deep architecture for contact and distance prediction. When supplemented with a fast approximation of a precision matrix, the learned representation can be used to produce distance predictions of comparable or greater accuracy as compared to our original DMPfold method. Constructing representations of complete MSAs also opens up ways of deriving other informative properties, such as predictions of likely eventual model accuracy derived solely by looking at the MSA, as well as a complete end-to-end method for directly predicting α-carbon coordinates, again directly from the MSA alone. Our methods will be made available on GitHub under a permissive license, as part of an upcoming new version of DMPfold.

1. Introduction

Analysis of amino acid residue covariation in deep multiple sequence alignments (MSAs) has revealed that covarying residues are frequently found to be close together in the tertiary structure. This principle has been successfully exploited for predicting inter-residue contacts by a variety of methods. Notable among these are the family of methods based on direct-coupling analysis (DCA) [1–3], and more recently, a number of methods based on deep learning. The latter have produced increasingly precise predictions of inter-residue contacts in recent years [4–6], and have also been adapted to output probabilistic predictions of the *distance* between the residues in contact, either as a probability distribution [7–10], or as real values [11–13]. These predictions are usually made by models that operate on precomputed features derived from an MSA, such as covariance and/or precision matrices, contact predictions from DCA-based methods, together with other features such as predicted secondary structure labels and sequence profiles. This approach, though effective, requires that these features be

precomputed, which can in some cases be a time-consuming process and can sometimes take longer than the rest of the prediction pipeline combined. Additionally, it does not allow the neural network (NN) model to utilise all the information that might be available in the sequence alignment, as a model trained on derived features is limited to using the information available in those handpicked features.

A key difficulty in using MSAs directly as input is the fact that an MSA for a target sequence can have arbitrarily many sequences in it. Only one published method, rawMSA [14], attempts to use an encoding of the MSA itself as the input to a deep neural network. In this method, the difficulty of embedding MSAs of arbitrary depth was addressed by training convolutional networks using predetermined maximum MSA depths. For deeper target MSAs, only a subset of sequences of the MSA are used, thus potentially discarding valuable additional information in the MSA. Here, we demonstrate that it is possible to use all the sequence information in the MSA when constructing a learned representation, and that such a representation can be used for effective prediction of structural features.

In this paper, we present: (a) a method for directly processing an MSA into a learned representation and its use in predicting structural features in proteins; (b) a method which predicts eventual model quality (in terms of TM-score [15]) using this learned representation; and (c) a method for end-to-end prediction of Cα coordinates from the MSA embedding. We show that despite the relatively simple composition of the input feature set, we are able to obtain predictions of comparable and sometimes better accuracy than DMPfold [9].

2. Methods

2.1. Datasets for training and evaluation

2.1.1. MSA embedding and structural feature prediction network

Training was conducted on a set of 31159 domains from the V4.2 CATH s35 representative set of domains, supplemented with 6742 full length chains from the original DMPfold1 training set. A set of 300 chains were held out from training to use as a validation set to monitor convergence. MSAs for each training example in the CATH s35 set were created using 3 iterations of HHblits v3.0b3 [16] using the UniClust30 (October 2017) database, with an E-value threshold of 0.001.

We tested the effectiveness of our method on a set of 39 domains from the CASP13 experiment, categorized as either FM or FM/TBM by the CASP13 assessors. MSAs were built using an approach similar to that taken by our group in CASP13 [5], using the UniClust30 2020_03 database, plus protein sequences from the UniRef100 [17], EBI MGnify [18], IMG/M [19], MetaEuk [20], and NCBI Transcriptome Shotgun Assembly databases. The latter five databases were searched when the initial HHblits/UniRef30 search returned fewer than 2000 hits; otherwise the HHblits MSA was used. The extended sequence databases were searched using 2 rounds of an iterative procedure comprising: hmmsearch [21] against the databases using a query MSA and a permissive E-value threshold; clustering the hits and aligning them

using kClust [22] and MAFFT [23]; preparing a custom database for HHblits using the aligned clusters; and finally, an HHblits search against this database.

2.1.2. TM-score prediction network

Beginning from all the domains in CASP8-12 with available coordinates, domains with greater than 30% sequence identity covering at least 60% of the protein length were clustered together using BlastClust [24]. A representative target was taken from each cluster at random to yield 500 non-redundant training domains overall.

An MSA was generated for each target domain using HHBlits searches against the UniClust30 (October 2017) database with the following parameters: E-value cutoff of 0.1, infinite number of effective sequences, maximum pairwise sequence identity of 90 and the '-diff' parameter set to inf. These parameters were intentionally kept lenient in order to increase the diversity of sequences within each MSA. To enhance the number of datapoints, we subsampled each alignment by generating sub-alignments starting from the target sequence only, and progressively adding 5 sequences at random from the full MSA in a stepwise manner. In most cases, the full alignments were not utilised. In total, 94742 alignments were generated across 500 target domains.

2.2. NN Model architectures, input and output features

2.2.1. MSA embedding and structural feature prediction network

A schematic representation of the MSA embedding procedure is shown in Figure 1. We used a system of gated recurrent unit (GRU) layers to process and embed the input MSA, and output a feature map with 256 x L values, where L is the length of the target sequence. Starting from a 22-dimensional one-hot encoding of each residue in each sequence in the MSA, a stack of 2 GRU layers scans individual columns in the MSA in the vertical direction. The hidden state of these layers is a vector of fixed size (512 in our case), and the hidden state at the end of the vertical scan is used as a fixed-length representation of the information in each column of the MSA. The per-column representations are then passed to another stack of 2 bidirectional GRU layers with 256 hidden units, producing final embeddings for each column of 2 x 256 values. These embeddings are striped vertically and horizontally (similar to the manner in which per-residue features are prepared for use by 2D convolutional layers [4,5]), and combined with a fast approximation of a precision matrix, calculated using the fast_dca algorithm from trRosetta [10]. These features are then used as input to a convolutional Maxout layer, followed by a series of 16 residual blocks composed of one maxout layer [25] and one squeeze-excitation layer [26] each. A schematic of the overall neural network system is shown in Figure 2.

The outputs of the ResNet are a collection of structural features for each residue pair, which we collectively term a "multigram": (a) binned distance predictions (distograms) in 34 bins of 0.5 Å width, with the first bin representing distances <4 Å, and the last bin representing distances >20 Å; (b) hydrogen bond predictions as in DMPfold1; and (c) backbone phi and psi torsion angle

predictions, each represented as values in 34 bins covering the range [0,2π]. The different structural features are predicted jointly by the same neural net and we colloquially refer to these combined outputs as a "multigram". For inference, we trained four different versions of the neural net and averaged their predictions. As with DMPfold1 [9], we also trained "iterative" versions of the neural net model, with an extra input feature channel carrying pairwise Cα distances from an existing 3-D structure. This extra feature channel is added in the inputs to the ResNet section of the model.

2.2.2. TM-score prediction network

The input features to the TM-score prediction network are derived from a 'differential distogram'. This distogram is constructed as the difference between two distograms: one generated using a given sub-alignment, and the other generated using only the target sequence. The differential representation describes the change in probability (Δp) at each i,j,k position of the distogram as a result of the sequences that accompany the target sequence in the sub-alignment. We further divide the differential distogram according to residue separation between i,j residues, to consider the predictive power of Δp in the context of short- (2<|i-j|<6), medium- (5<|i-j|<24) and long-range (|i-j|>=24) contacts.

To generate a feature set of fixed length and which is invariant to the protein length, we bin all Δp values within each of the 34 distance bins into a one-dimensional distribution of Δp values. Distance bins 1-33 are each binned into 20 bins (from Δp of -0.1 to +0.1 with width of 0.01). Distance bin 34 has an open ended distance threshold of >20 Å. As such, the Δp distribution is wider, and is instead binned into 40 bins from -0.2 to +0.2 (bin width of 0.01). Overall, distance bins 1-33 and 34 contribute 660 and 40 features, respectively. Finally, the bin counts are flattened into a 1x700 feature array.

Using the above approach, we assessed a total of ten feature sets, consisting of either the binned Δp values derived from short, medium or long-range contacts directly (700 features), or summed or concatenated combinations (700, 1400 or 2100 features). Thus our feature sets can also be grouped according to the number of features used:

700 features:
- Binned Δp - short
- Binned Δp - medium
- Binned Δp - long
- Binned Δp - all

1400 features:
- Binned Δp - short & medium
- Binned Δp - short & long
- Binned Δp - medium & long
- Binned Δp - short & sum(med, long)
- Binned Δp - sum(short, med) & long

2100 features:
- Binned Δp - short & medium & long

The TM-score prediction network is a fully connected neural network consisting of 9 hidden layers, each with 200 neurons. The rectified linear unit activation function (ReLU) is applied to each hidden layer.

Given an alignment, the aim of the TM-score prediction network is to predict the potential TM-score associated with a generated model when using the alignment as an input to a model generation pipeline. Since multiple model conformations can arise from the same alignment, we predict the potential TM-score as a distribution rather than as a single value. To do this, we quantize the TM-score into an ordered set of 100 labels as $\{0.01,\ 0.02,\ ...,\ 1.00\}$. The network produces 100 log softmax outputs corresponding to the TM-score distribution $y \in \mathbb{R}^{100}$ where y_i is the probability that input X has a TM-score of $i/100$. The final TM-score is returned as the expected value of the probability distribution.

2.3. Protein model generation

a) Distance geometry and simulated annealing: DMPfold1 predictions were made using the CNS-based procedure as previously described [9]. The structural features predicted by the DMPfold2 NN model were used as constraints for the distance geometry and simulated annealing procedure in CNS [27], which is identical to that used in DMPfold1. Procedures for converting the distance and torsion angle predictions into upper and lower bounds were optimised in a similar manner as in DMPfold1. After the first round of predictions, one additional round of the iterative distance prediction and model generation step was run to get the final model for both DMPfold1 and 2. 50 candidate models were generated at each step. No all-atom refinement was performed on the models.

b) End-to-end prediction of coordinates from the MSA: To evaluate the ability of the DMPfold2 model to be trained and used in a purely end-to-end fashion, a modified network architecture was used with the normal multigram convolutional output layer replaced by a single channel convolutional layer. The outputs from this layer are converted to a real-value distance matrix D by first averaging across the diagonal to ensure a symmetric matrix and then taking the absolute values. This matrix is projected to 3-D Cα coordinates using multi-dimensional scaling [28]. The following (Gram) matrix is defined:

$$M_{ij} = \frac{D_{1j}^2 + D_{i1}^2 - D_{ij}^2}{2}$$

The eigendecomposition $M = USU^T$ is then calculated, where $X = U\sqrt{S}$ gives the coordinates of the points. Note that if the atoms can be fully embedded in 3-D space there will only be 3 non-zero eigenvalues of M, but this is not necessarily the case for the real-value distance matrix predicted by the network. Coordinates corresponding to the largest 3 eigenvalues were therefore calculated and the rest discarded. Once the 3-D coordinates are obtained, the distance map computed from these coordinates is embeddable in 3-D space. As distances are invariant under mirror symmetry, the

multidimensional scaling can produce a mirror of the correct structure arbitrarily. This can be manually detected by evaluating the overall distribution of torsion angles, and a mirror transformation applied to the coordinates if required. In our end-to-end model, however, we were able to avoid these steps by simply feeding the output coordinates through a final 2-layer bidirectional GRU recurrent network (256 weights per hidden layer), with 3 input channels and a final fully connected layer with 512 inputs and 3 outputs. During training with coordinate RMSD loss, this final network is able to learn to automatically transform the output coordinates for any structures mirrored by the multidimensional scaling process, without manual intervention.

The current version of the DMPfold2 end-to-end model predicts Cα coordinates only, however the missing main-chain and Cβ atom positions can very quickly be reconstructed using either PULCHRA [29] or the 'catomain' routine from the DRAGON method [30]. No further optimisation or refinement of the structure was attempted in any of the results shown.

2.4. Training procedures

2.4.1. MSA embedding and structural feature prediction network

The DMPfold2 network was implemented using PyTorch [31] and trained using the Adam optimizer [32] and focal loss [33], with parameter α set to 0.5. We found that focal loss is better able than cross-entropy loss to deal with the imbalance in the ground truth labels (distance distributions at different sequence separation ranges) and can also focus learning on the harder-to-predict long-range distances by automatically reducing the weight on easy-to-predict features of the multigram as training proceeds. An initial learning rate of 0.0003 was used, a dropout probability of 0.1 was used for the recurrent network layers, and dropout 0.2 used in the convolutional layers. For data augmentation, at each epoch each alignment in the training set is subsampled as follows:

a) Random rows from the alignment are selected, up to a maximum of 1000, though always including the target sequence in the first row.
b) Columns with gaps in just one randomly selected alignment row are then deleted. This simulates an evolutionary deletion process in the target sequence, producing biologically relevant random crops of the target sequences i.e. crops that are consistent with previously observed deletions in the alignment. These deleted columns are also removed from the target tensors so that the target distograms, for example, have the same number of rows and columns as the length-reduced target sequence.

Four separate networks were trained from different random weight initializations, so that outputs from this ensemble of networks could be averaged during inference.

For training the end-to-end model of DMPfold2, the same training procedure was followed, though with RMSD coordinate error as the loss function, and with a fine-tuning procedure for weights in common with the standard multigram network i.e. those weights were initialized with

the values taken from one of the original fully-trained DMPfold2 networks. In this case, the RMSD between the output Cα coordinates and the experimental structure Cα coordinates is calculated using a fast quaternion-based method [34], and this was used as the loss function for backpropagation rather than focal loss. In order to avoid numerical instability during training, the final square root in the RMSD calculation is omitted. No model ensembling was used in this case as it would be highly suboptimal to produce a simple average of predicted real coordinates, though we may investigate other ensembling strategies in the future.

2.4.2. TM-score prediction network

The TM-score prediction network was trained using the Adam optimizer with an initial learning rate of 0.001. Losses are calculated using Kullback-Leibler (KL) divergence and are back-propagated with respect to a known TM-score distribution that is derived from a sample of 50 models generated using DMPfold2 with the CNS-based modelling procedure (Section 2.3a) from each alignment.

To assess the performance of the ten network variants, we trained each using 450 domains from CASP8-12 (85033 alignments) and used 35 domains from CASP13 FM targets for testing (2586 alignments). 50 domains from CASP8-12 (9709 alignments) were selected at random and used for validation to prevent overfitting. The same set of validation targets were maintained when training each network variant. Each network was trained for up to 200 epochs and early stopping with a patience of 50 steps was employed to monitor the validation loss to avoid overfitting.

2.5. Performance evaluation procedures

2.5.1. Distances, contacts and tertiary structure predictions

To evaluate distance predictions (distograms), we used a simple measure of the quality of the distograms. For a given residue pair, we calculated the maximum likelihood distance from the distograms by removing the last bin (corresponding to a distance of >20 Å), re-normalising the remainder of the bin probabilities to sum to 1, and summing the bin centre distances multiplied by the re-weighted probabilities. The absolute difference between this maximum likelihood distance and the native distance was calculated, and the mean taken over all residue pairs separated by at least 5 residues in sequence and closer than 20 Å in the native structure.

Distance predictions (distograms) were also converted to binary contact predictions by summing up the probabilities in the distogram up to the bin ending at 8 Å. Contact predictions were assessed by ranking the contacts in descending order of scores, and evaluating the precision of the top-L/x contacts, where L is the length of the target sequence, and x is one of 1, 2, 5, or 10.

2.5.2. TM-score predictions

TM-score prediction accuracy was assessed as the mean absolute error (MAE) with respect to

the true TM-score. The quoted MAE of each network variant was calculated as the average MAE of each target set of data points. We additionally calculated the average potential loss in TM-score for each network variant. This metric is calculated by taking the absolute difference between the actual TM-scores of the predicted best MSA and the true best MSA. One TM-score loss value is calculated per target, and the quoted value is the average over all targets.

3. Results and discussion

3.1. Accuracy of distance predictions

We evaluated the distance predictions from the network by first converting them into binary contacts using the standard 8 Å threshold. Table 1 shows the long-range precision of these contact predictions, comparing the outputs from the DMPfold1 NNs and those from the NNs using MSA embedding, which we will denote as DMPfold2 for convenience. We compared both versions of the DMPfold distance prediction NNs against rawMSA, the only other published method that uses a directly learned representation of the input multiple sequence alignment. The comparison shows that the new method employing a learned representation of the MSA is comparable to the DMPfold1 NNs in terms of contact precision on the 39 CASP13 domains, and on some measures is slightly better. Both methods are substantially more precise than rawMSA. These results are encouraging given that we have replaced a large set of the input features used in DMPfold1 with a learned representation of the MSA. Although the best performance is obtained when supplementing the learned representation with a precision matrix, these observations suggest that accurate prediction of structural features with minimal pre-processing of the MSA is possible.

Although rawMSA does not predict residue-residue distances, we can compare distance predictions for DMPfold1 and DMPfold2. The MAE between the maximum likelihood predicted distance and the native distance was calculated for all native residue pairs closer than 20 Å. This value is 2.01 Å for DMPfold2 and 2.23 Å for DMPfold1, indicating an improvement in performance.

3.2. Tertiary structure model accuracy

As an additional evaluation of the effectiveness of the MSA embedding, we built 3-D models of 39 FM and FM/TBM targets from the CASP13 experiment, using the DMPfold2 NNs and the same CNS-based model building procedure used in DMPfold1, as well as the end-to-end model generation procedure.

Comparison against models generated using the DMPfold1 NNs (Figure 3 and Table 2) shows that the DMPfold2 NNs produce predictions that, on average, enable more accurate structure modelling for these domains. DMPfold2 is able to fold 30 domains to a TM-score of 0.5 or greater, as compared to 26 for DMPfold1. The mean TM-score is 0.557 for DMPfold2 and 0.531 for DMPfold1. Additional benefits may be realised by using different model building procedures,

and we are currently experimenting with a variety of model building protocols with a view to improve model quality.

Figure 4 compares the TM-scores of models generated by DMPfold2, using either the standard CNS-based structure reconstruction approach, or coordinates predicted by the end-to-end version of the neural networks. We found that using either PULCHRA or catomain to build the missing main-chain and Cβ positions gave very similar results, with catomain producing a slightly higher mean TM-score. It can be seen that although the models generated by the end-to-end version are not as precise as the CNS models on average (mean TM-score of 0.484 compared to 0.557 for the CNS version), they are still often of acceptable quality, especially when considering the relatively straightforward method used to reconstruct atom positions. Interestingly, there was one target (T0986s2-D1) for which the end-to-end prediction produced the correct fold where CNS could not. The end-to-end prediction for this domain had a TM-score of 0.52 compared to 0.23 for the CNS version.

3.3. Evaluation of model quality estimates predicted directly from MSAs

We developed a network that predicted the eventual model TM-score directly from an MSA. Ten variants of this network were trained, using features that described short, medium and long-range interactions, or combinations thereof (Table 3 and Figure 5). These networks each take an MSA as input and outputs a predicted TM-score distribution. The performance of each network is assessed as the average MAE between the predicted and actual TM-score. The latter is evaluated as the mean TM-score of 50 models that were produced with DMPfold2 when given the MSA as an input. We assessed the performance of each network by testing on the CASP13 target domains.

Comparison of the ten network variants reveals that the lowest average MAE of 0.062 is achieved using the *sum(short, med) & long* features. This feature set fundamentally consist of short/medium-range contacts (2>|i-j|>24) and long-range contacts (|i-j|>=24). While the lowest MAE is achieved using the *sum(short, med) & long* features, similarly low MAE values were also obtained from other feature sets. The MAE values of the ten networks cluster into two groups - those that achieve above or below an MAE of 0.09. This clustering is paralleled by the inclusion of long-range features which result in a lower MAE being achieved. However, while it appears that short and medium-range contact-based features are less informative on their own, their inclusion into the network synergistically complements the predictive power of the long-range features.

A key benefit of being able to predict the eventual TM-score is that MSAs can be ranked according to their predicted modelling power, prior to the time-consuming step of model generation. To assess the ranking power of the networks, we calculated the potential loss in TM-score for each set of target alignments for each network variant (Table 3). This metric answers the following question: what is the potential loss in the actual TM-score if alignments are ranked according to the TM-score predictions? The lowest average potential TM-score loss was 0.0338 using the *short & sum(med, long)* features. However, similarly low TM-score loss

can be observed across the other networks tested, even for networks which do not achieve a low MAE. Taken together, this indicates that meaningful MSA ranking does not necessarily rely on accurate TM-score predictions.

4. Conclusions and future directions

Recent successes in applying deep learning to protein structure prediction have mostly relied on the use of large, precomputed feature sets as inputs to the deep learning model. Here, we show that it is possible to directly process a multiple sequence alignment into a learned representation that is more effective for predicting structural features in proteins. Using this approach, we are developing the next generation of our DMPfold method for deep learning-based structure prediction. The distance predictions from the DMPfold2 NNs are of comparable or higher precision than those obtained from DMPfold1 NNs, and the structural models built from the DMPfold2 restraint sets are also more accurate on average than those from DMPfold1. It is expected that allowing the network to access all the information in a raw MSA, rather than just pairwise frequency information, for example, enables it to extract richer information that can be used for more accurate prediction of structural features. Although the best performance is obtained when using the learned MSA representation alongside an on-the-fly computed precision matrix, the work in this study opens up the possibility of using just the MSA representation itself as the sole input to the ResNet for predicting protein structure, though that will probably require using methods with better ability to deal with very long range dependencies (see below). It also enables new lines of work that were prohibitive with large feature sets (such as those used in DMPfold1), due to the time and storage requirements of using those features. As an example, we developed a predictor of model TM-score that uses different versions of predicted distograms as input, each of which was generated using different versions of an input MSA. Training a predictor in this fashion would have been extremely time-consuming with a more traditional large feature set. The ability to predict TM-score from an MSA, together with the ability to easily work with different versions of an MSA, opens up the possibility of *optimizing* the set of sequences in an MSA so as to produce the most accurate structure predictions. We trialled the use of such a procedure in the CASP14 experiment.

The ability of DMPfold2 to be easily trained and used in a strictly end-to-end manner also opens up new possibilities. The idea of end-to-end *de novo* prediction [35,36] has certainly been tantalizing, in that a 3-D model can be produced in a fraction of a second compared to the hours or days needed previously. So far, however, published end-to-end methods have not been able to produce models comparable to the state-of-the-art in *de novo* prediction, mainly because they have not effectively exploited covariation data as inputs. Indeed, the very idea of having to spend time pre-calculating covariation input sequence features would obviously devalue the whole concept of end-to-end modelling. By combining direct MSA embedding with the idea of end-to-end coordinate generation by a learned multidimensional scaling process in the neural network, DMPfold2 is able to produce Cα coordinates comparable in accuracy to the full CNS modelling approach in just a few hundred milliseconds per target. This will allow quick validation of *de novo* predictions before full modelling is carried out, or could be combined with a refinement method to almost completely replace the whole 3-D modelling pipeline. Another

usage example would be to visualize changes to the final 3-D model as the input sequences are changed, virtually in real time.

The use of two gated recurrent networks to embed the entire MSA is clearly effective, but was something of a design compromise in that RNNs are relatively fast to train, but have known limitations in terms of the limits of modelling long range dependencies in sequences. In theory, gated RNNs are able to avoid the problems of vanishing gradients when modelling long sequences, but in practice, dependencies beyond a window of a few hundred time steps are poorly modelled. For this reason we used two GRU networks, one to embed in the vertical (sequence number) and one in the horizontal (residue number) direction so that the number of time steps that each GRU would need to model would be limited by either the lengths of typical protein domains or the depths of typical MSAs. For MSAs with longer sequences or with many more homologous sequences, even gated RNNs will start to become ineffective. We are currently investigating alternative means of embedding MSAs, such as the use of models based on new efficient transformer architectures [37], which are far more memory-efficient than the original Transformer [38] due to avoiding the calculation of large self-attention matrices over the length of the sequences. Standard transformer models have already been used to embed unaligned protein sequences [39–41], but the most efficient transformer models released in the last year are now capable of handling sequence lengths even in the millions, and so this suggests that a single deep transformer model with a compressed self-attention mechanism could, in principle, embed a whole MSA in one go by treating it as a single sequence. Current experiments along these lines are promising.

Overall, we have demonstrated that the idea of embedding whole MSAs into a linear representation using standard language modelling approaches can produce excellent results, both in terms of residue contact and distance prediction accuracy, and the final generation of 3-D structures directly from sequence. By being able to directly link individual amino acids in an MSA to the outputs of the network, many structural bioinformatics applications are made easier e.g. modelling variant effects or protein design. At the very least, these direct MSA embedding methods make *de novo* protein structure prediction methods far more efficient and easier to use.

Figures

Figure 1. Details of the MSA embedding procedure. The MSA is represented by a one-hot encoding of 22 residue types (including gaps and unknown residues). First, the residues in a single column of the MSA are treated as timesteps and fed as input to a stack of two Gated Recurrent Unit (GRU) network layers. The final hidden state of the second GRU, obtained after processing the whole column of the MSA, is used as an embedding of the information in that column. The process is then repeated for the remaining columns in the MSA, producing a separate embedding for each MSA column. Finally, these per-column embeddings are used as inputs to a stack of 2 bidirectional GRU layers that produces an embedding of all sequences and columns in the MSA. The dimensions of the input tensor and embeddings are shown in parentheses.

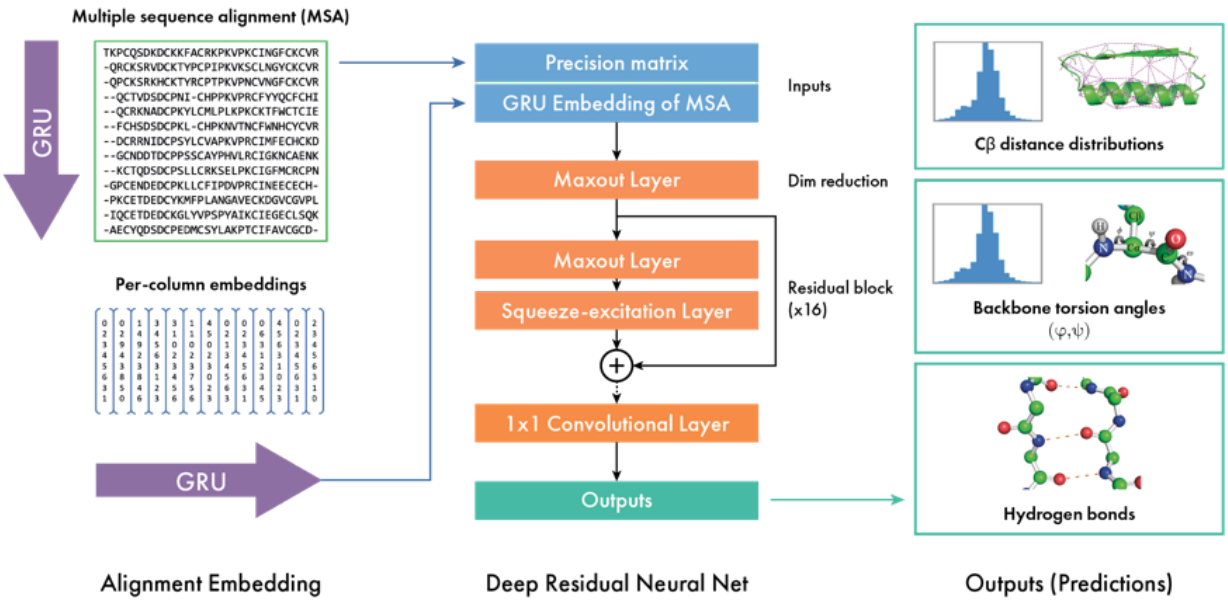

Figure 2. **Schematic representation of the DMPfold2 neural net.** The input multiple sequence alignment (MSA) is processed into a learned embedding, as shown in detail in Figure 1. This embedding is combined with a precision matrix calculated from the MSA, and fed to a convolutional Maxout layer to reduce its dimensionality, before being fed to a series of 16 residual neural net (ResNet) blocks. Each block is composed of a convolutional Maxout layer and a Squeeze-excitation layer. The outputs from the network are collected as the outputs of a 2D convolutional layer with 1x1 filter, and are a combination of different structural features, represented in the right-hand column. All outputs from the network are predicted jointly.

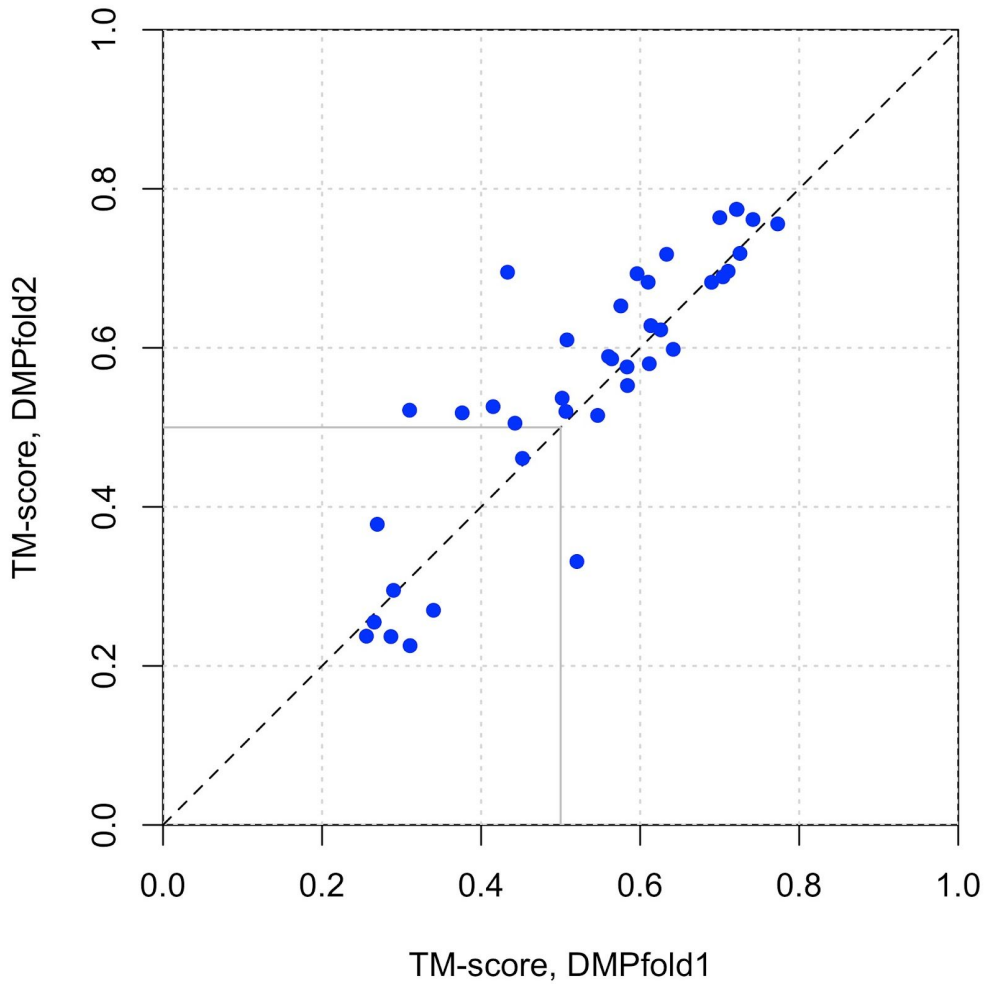

Figure 3. TM-scores obtained on the CASP13 domains, comparing DMPfold1 and DMPfold2. Detailed model accuracy data for each target can be found in Table 2. A dashed line of unit slope is drawn, as well as segments demarcating the TM-score >= 0.5 regions of the plot. Overall, DMPfold2 achieves comparable or greater model accuracy than DMPfold1, using a similar model generation strategy.

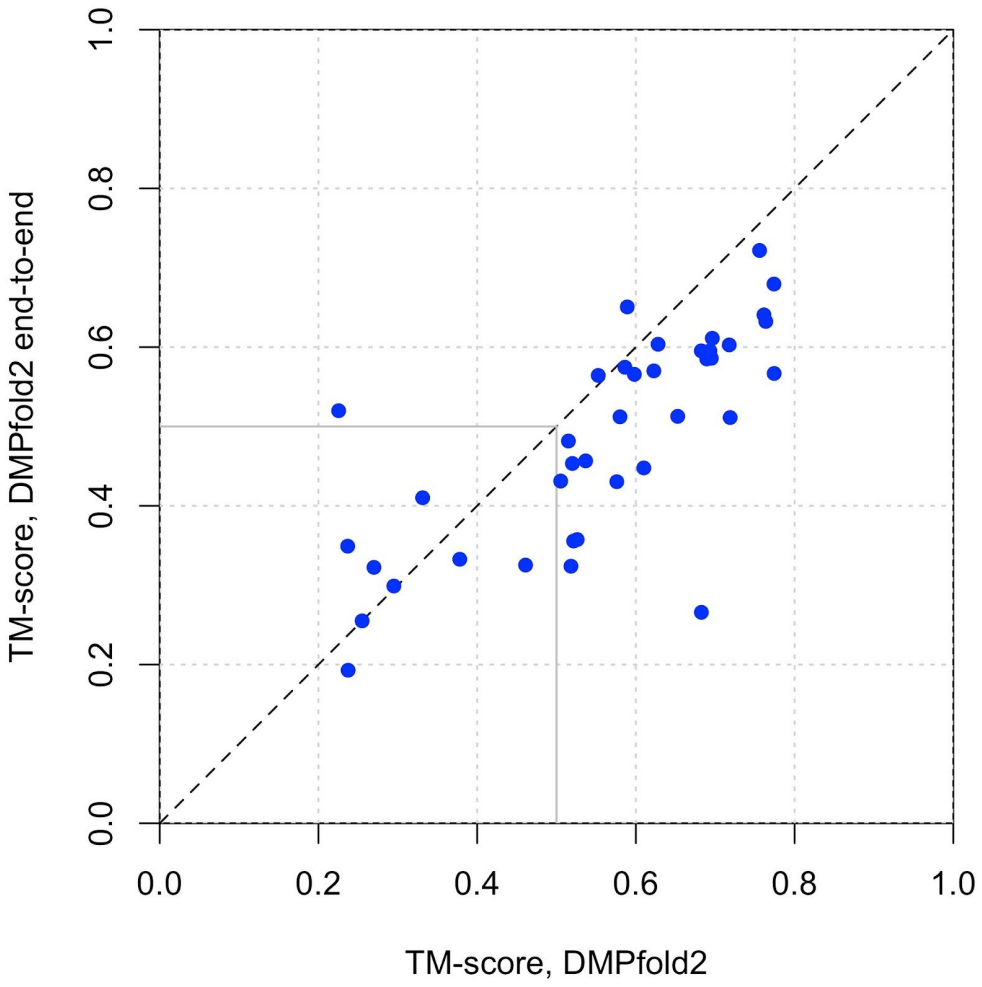

Figure 4. TM-scores obtained on the CASP13 domains, comparing DMPfold2 using the CNS-based distance geometry approach and DMPfold2 trained to predict coordinates in an end-to-end fashion. Data are represented as in Figure 3. For the end-to-end model, data are shown using the 'catomain' procedure for rebuilding all main-chain and Cβ positions. Similar results are obtained when using PULCHRA to rebuild these atom positions. One target, T0986s2-D1, could be folded correctly (TM-score>0.5) by the end-to-end procedure but not by CNS.

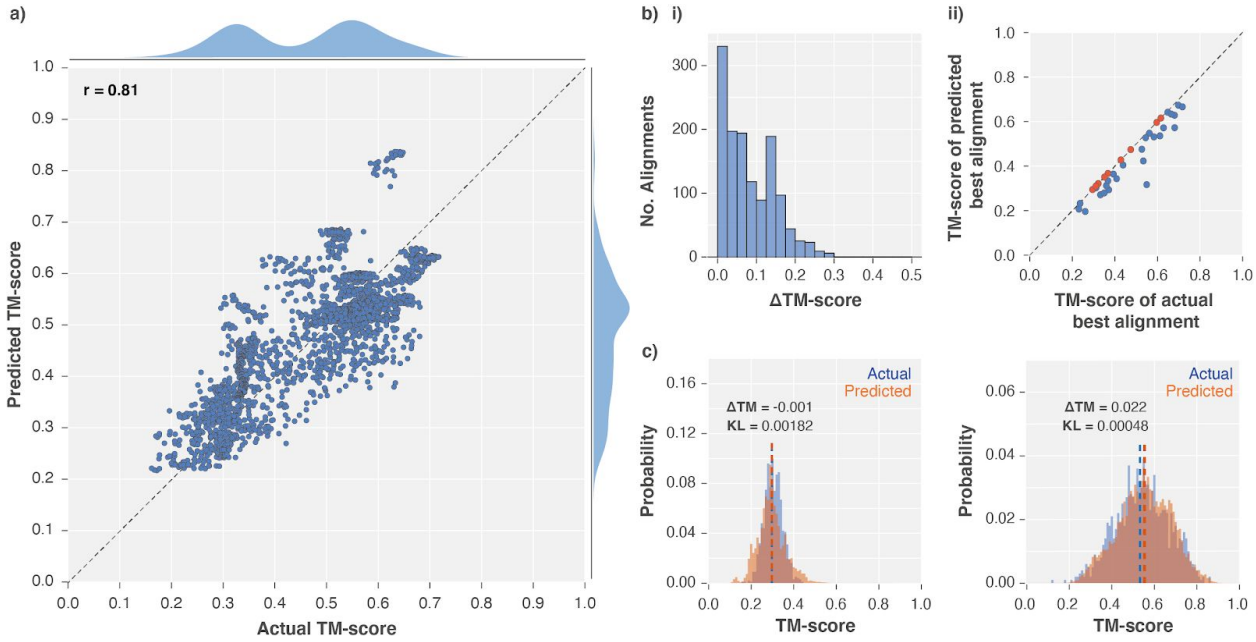

Figure 5. TM-score predictions directly from alignments. (a) Actual vs predicted TM-scores for 2586 alignments across 35 domains from CASP13 FM targets. 'r' represents the correlation coefficient. (b) Accuracy of TM-score predictions. (i) Distribution of absolute TM-score error (ΔTM-score) of predictions. (ii) Comparison of actual TM-scores of predicted best alignment and actual best alignment. Data points in red (9 targets) indicate that the actual best alignment has been correctly predicted by the network when ranked by predicted TM-score. (c) Examples of actual (blue) vs predicted (orange) TM-score distributions. Blue and orange dotted lines represent the mean TM-score of 50 models generated using the alignment, and the expected value of the predicted TM-score distribution respectively. ΔTM and KL labels show the numerical difference in TM-score between the two dotted lines, and the KL-divergence between the two distributions, respectively. The data shown is from the network variant trained using *sum(short, med) & long* features.

Tables

Table 1. Comparison of long-range (|i-j| >= 24) contact precision on the 39 FM and FM/TBM domains from CASP13. Predictions for rawMSA were computed using MSAs built using only HHblits searches against the UniRef30 database, as we found that predictions made using these MSAs were more precise than those obtained using the deeper MSAs built using the metagenomic sequences. DMPfold 1 and 2 used the deeper MSAs.

Method	Mean long-range contact precision in range [0,1]			
	Top-L	Top-L/2	Top-L/5	Top-L/10
DMPfold2	0.4808	0.6309	0.7568	0.7732
DMPfold1	0.4866	0.6318	0.7423	0.7775
rawMSA	0.1774	0.2456	0.3163	0.3760

Table 2. Model accuracy on the 39 CASP13 domains, comparing models built using structural feature predictions from either DMPfold1 or DMPfold2 NNs. The same MSAs were supplied to each method, and only the top-ranking model selected by each modelling run was evaluated for each target.

Target	DMPfold1			DMPfold2		
	GDT_HA	GDT_TS	TM-Score	GDT_HA	GDT_TS	TM-Score
T0949-D1	36.434	54.845	0.626	35.853	54.264	0.623
T0950-D1	24.342	40.863	0.610	33.187	50.000	0.683
T0953s1-D1	25.373	44.030	0.376	41.045	57.836	0.518
T0953s2-D1	23.295	34.659	0.256	23.864	31.250	0.237
T0955-D1	41.463	52.439	0.415	52.439	71.341	0.526
T0957s2-D1	36.774	58.226	0.690	35.161	56.129	0.682
T0958-D1	50.974	70.779	0.704	50.325	70.130	0.689
T0960-D2	38.690	57.143	0.564	38.690	59.226	0.586
T0963-D2	41.463	59.451	0.584	34.451	55.488	0.553
T0968s1-D1	46.186	67.373	0.742	52.966	72.458	0.761
T0968s2-D1	41.957	63.696	0.721	48.043	69.565	0.774
T0969-D1	19.986	37.641	0.633	30.297	50.000	0.718

T0970-D1	40.000	58.529	0.596	48.529	68.235	0.693
T0975-D1	19.217	33.185	0.507	21.530	35.943	0.520
T0978-D1	18.765	35.472	0.584	24.516	38.438	0.576
T0980s1-D1	18.750	30.288	0.310	32.452	50.000	0.522
T0981-D3	28.941	48.399	0.642	29.557	46.305	0.598
T0986s1-D1	47.011	68.750	0.726	48.370	69.565	0.719
T0986s2-D1	12.419	22.419	0.311	11.613	18.226	0.225
T0987-D1	22.568	40.676	0.547	24.730	39.459	0.515
T0987-D2	23.597	38.903	0.520	14.796	23.724	0.331
T0989-D1	24.440	37.500	0.443	29.478	43.657	0.505
T0989-D2	16.071	29.464	0.340	12.946	22.321	0.270
T0990-D1	44.408	65.461	0.611	41.118	61.513	0.580
T0990-D3	11.502	19.601	0.290	12.441	20.305	0.295
T0992-D1	50.467	72.430	0.773	49.299	70.561	0.756
T0997-D1	36.216	58.243	0.722	43.243	64.459	0.774
T0998-D1	10.843	18.825	0.269	20.783	31.325	0.378
T1000-D2	27.785	46.807	0.700	34.035	54.620	0.764
T1001-D1	28.237	46.763	0.560	32.374	54.137	0.589
T1005-D1	21.472	39.034	0.613	24.693	41.718	0.628
T1008-D1	19.481	27.597	0.266	19.805	27.597	0.255
T1010-D1	41.190	58.810	0.711	40.119	57.262	0.696
T1015s1-D1	27.273	42.898	0.452	35.511	45.739	0.461
T1017s2-D1	27.600	43.400	0.502	27.400	46.600	0.537
T1019s1-D1	33.621	53.879	0.433	53.448	73.707	0.695
T1021s3-D1	27.410	45.783	0.576	33.886	53.313	0.653
T1021s3-D2	13.918	26.031	0.287	18.814	26.804	0.237
T1022s1-D1	23.878	39.904	0.508	32.692	50.160	0.610
Mean	29.334	45.902	0.531	33.192	49.574	0.557

Table 3. Comparison of features used for predicting eventual TM-score of models generated from MSAs. Each network was trained using 450 CASP8-12 domains (85033 alignments) and tested on 35 domains from CASP13 FM targets (2586 alignments). The MAE and TM loss are both calculated as the average values across each set of target alignments.

Features	**Inputs**	**Avg. MAE**	**Avg. TM loss**
short	700	0.1085	0.0544
med	700	0.0903	0.0439
long	700	0.0691	0.0399
sum(short, med, long)	700	0.0633	0.0410
short & med	1400	0.0939	0.0416
short & long	1400	0.0627	0.0385
med & long	1400	0.0642	0.0402
short & sum(med, long)	1400	0.0670	0.0338
sum(short, med) & long	1400	0.0620	0.0379
short, med, long	2100	0.0657	0.0399

References

1. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proc Natl Acad Sci U S A. 2011;108: E1293–301. doi:10.1073/pnas.1111471108

2. Balakrishnan S, Kamisetty H, Carbonell JG, Lee S-I, Langmead CJ. Learning generative models for protein fold families. Proteins. 2011;79: 1061–1078. doi:10.1002/prot.22934

3. Ekeberg M, Lövkvist C, Lan Y, Weigt M, Aurell E. Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Phys Rev E Stat Nonlin Soft Matter Phys. 2013;87: 012707. doi:10.1103/PhysRevE.87.012707

4. Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. PLoS Comput Biol. 2017;13: e1005324. doi:10.1371/journal.pcbi.1005324

5. Kandathil SM, Greener JG, Jones DT. Prediction of interresidue contacts with DeepMetaPSICOV in CASP13. Proteins. 2019;87: 1092–1099. doi:10.1002/prot.25779

6. Li Y, Zhang C, Bell EW, Yu D, Zhang Y. Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. Proteins. 2019; 560029. doi:10.1002/prot.25798

7. Xu J. Distance-based protein folding powered by deep learning. Proc Natl Acad Sci U S A. 2019;116: 16856–16865. doi:10.1073/pnas.1821309116

8. Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, et al. Improved protein structure prediction using potentials from deep learning. Nature. 2020;577: 706–710. doi:10.1038/s41586-019-1923-7

9. Greener JG, Kandathil SM, Jones DT. Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints. Nat Commun. 2019;10: 3977. doi:10.1038/s41467-019-11994-0

10. Yang J, Anishchenko I, Park H, Peng Z, Ovchinnikov S, Baker D. Improved protein structure prediction using predicted interresidue orientations. Proc Natl Acad Sci U S A. 2020;117: 1496–1503. doi:10.1073/pnas.1914677117

11. Kukic P, Mirabello C, Tradigo G, Walsh I, Veltri P, Pollastri G. Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. BMC Bioinformatics. 2014;15: 6. doi:10.1186/1471-2105-15-6

12. Wu T, Guo Z, Hou J, Cheng J. DeepDist: real-value inter-residue distance prediction with deep residual convolutional network. doi:10.1101/2020.03.17.995910

13. Adhikari B. A fully open-source framework for deep learning protein real-valued distances. Sci Rep. 2020;10: 13374. doi:10.1038/s41598-020-70181-0

14. Mirabello C, Wallner B. rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments. PLoS One. 2019;14: e0220182. doi:10.1371/journal.pone.0220182

15. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure

template quality. Proteins. 2004;57: 702–710. doi:10.1002/prot.20264

16. Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods. 2011;9: 173–175. doi:10.1038/nmeth.1818

17. Suzek BE, Wang Y, Huang H, McGarvey PB, Wu CH, UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics. 2015;31: 926–932. doi:10.1093/bioinformatics/btu739

18. Mitchell AL, Almeida A, Beracochea M, Boland M, Burgin J, Cochrane G, et al. MGnify: the microbiome analysis resource in 2020. Nucleic Acids Res. 2020;48: D570–D578. doi:10.1093/nar/gkz1035

19. Chen I-MA, Chu K, Palaniappan K, Pillay M, Ratner A, Huang J, et al. IMG/M v.5.0: an integrated data management and comparative analysis system for microbial genomes and microbiomes. Nucleic Acids Res. 2019;47: D666–D677. doi:10.1093/nar/gky901

20. Levy Karin E, Mirdita M, Söding J. MetaEuk—sensitive, high-throughput gene discovery, and annotation for large-scale eukaryotic metagenomics. Microbiome. 2020;8: 48. doi:10.1186/s40168-020-00808-x

21. Eddy SR. Accelerated Profile HMM Searches. PLoS Comput Biol. 2011;7: e1002195. doi:10.1371/journal.pcbi.1002195

22. Hauser M, Mayer CE, Söding J. kClust: fast and sensitive clustering of large protein sequence databases. BMC Bioinformatics. 2013;14: 248. doi:10.1186/1471-2105-14-248

23. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol. 2013;30: 772–780. doi:10.1093/molbev/mst010

24. Dondoshansky I, Wolf Y. BlastClust documentation (NCBI software development toolkit). NCBI, Bethesda, MD. 2000;14. Available: https://ftp.ncbi.nih.gov/blast/documents/blastclust.html

25. Goodfellow I, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout Networks. In: Dasgupta S, McAllester D, editors. Proceedings of the 30th International Conference on Machine Learning. Atlanta, Georgia, USA: PMLR; 2013. pp. 1319–1327. Available: http://proceedings.mlr.press/v28/goodfellow13.html

26. Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. doi:10.1109/cvpr.2018.00745

27. Brunger AT. Version 1.2 of the Crystallography and NMR system. Nat Protoc. 2007;2: 2728.

28. Young G, Householder AS. Discussion of a set of points in terms of their mutual distances. Psychometrika. 1938;3: 19–22. doi:10.1007/bf02287916

29. Rotkiewicz P, Skolnick J. Fast procedure for reconstruction of full-atom protein models from reduced representations. J Comput Chem. 2008;29: 1460–1465.

30. Aszódi A, Taylor WR. Secondary structure formation in model polypeptide chains. Protein Eng. 1994;7: 633–644. doi:10.1093/protein/7.5.633

31. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2019. pp. 8026–8037.

32. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv [cs.LG]. 2014. Available: http://arxiv.org/abs/1412.6980

33. Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision. 2017. pp. 2980–2988.

34. Theobald DL. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. Acta Crystallogr A. 2005;61: 478–480. doi:10.1107/S0108767305015266

35. AlQuraishi M. End-to-End Differentiable Learning of Protein Structure. Cell Syst. 2019;8: 292–301.e3. doi:10.1016/j.cels.2019.03.006

36. Ingraham J, Riesselman AJ, Sander C, Marks DS. Learning Protein Structure with a Differentiable Simulator. ICLR. 2019. Available: https://openreview.net/forum?id=Byg3y3C9Km

37. Tay Y, Dehghani M, Bahri D, Metzler D. Efficient Transformers: A Survey. arXiv [cs.LG]. 2020. Available: http://arxiv.org/abs/2009.06732

38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. arXiv [cs.CL]. 2017. Available: http://arxiv.org/abs/1706.03762

39. Alley E, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. Nature Methods. 2019;16: 1315–1322. doi:10.21203/rs.2.13774/v1

40. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Guo D, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. bioRxiv. 2019. Available: https://www.biorxiv.org/content/10.1101/622803v3

41. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. bioRxiv. 2020. Available: https://www.biorxiv.org/content/10.1101/2020.07.12.199554v2