Sequence Analysis - Protein Folding

Author: Manuel Concepción Brito

1. Introduction

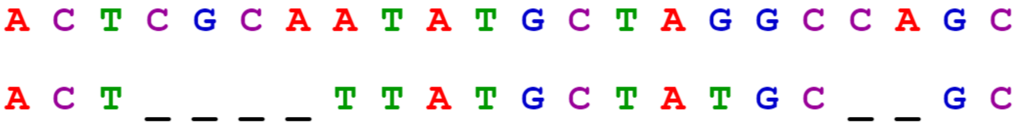
If cells are the building blocks of life, proteins are the workers inside, outside and in between cells and other organisms. Proteins are made out of building blocks, called amino acids or residues. In nature, there exist 20 amino acids which elevate the combination of possible proteins to 20^L, where L is the length of a protein. Due to interactions between amino acids, proteins adopt a unique 3D shape referred to as the protein fold. Protein folding determination has long been one of the unsolved challenges of modern science. The folding of a protein largely determines their biological activity and function within living organisms. Therefore, determining protein folding has huge implications in drug discovery and development.

The similarity between sequences, the complexity of the problem and the amount of data available makes protein folding to fall in the realm of deep learning. The objective of this review is to present the similarities between three state of the art methods: AlphaFold [1], RaptorX [2] and TripletRes [3]. The text is divided into five parts: an introduction to the protein folding problem, an extensive comparison on the differences between AlphaFold, RaptorX and TripletRes; the results of these methods in the CASP13 competition, a section exploring further work in the literature, and finally a conclusion and outlook to the future of this field.

2. Preface on the Problem of Protein Folding

Proteins are macromolecules consisting of chains of amino acids. Amino acids within a protein sequence, often called residues, appear in a distinctive order due to their physical properties [4]. Even though protein space is extremely large, there some similarities that allow for the study and approximation of their 3D shape.

Organisms construct proteins from DNA sequences in a two-step process. The first step of this process, called transcription, transforms DNA sequences into complementary messenger RNA (mRNA) sequences. Later, these mRNA sequences are converted to proteins in a step called translation. Consequently, changes in DNA alter subsequent proteins. These changes, called mutations, are relatively common in organisms, acting as the motor for evolution. Therefore, there exist groups of related proteins (families) with just a few different amino acids. The similarity among these proteins causes their 3D shape to be extremely similar. As a consequence, the study of protein similarity to produce protein folds is one of the pillars of modern methods.



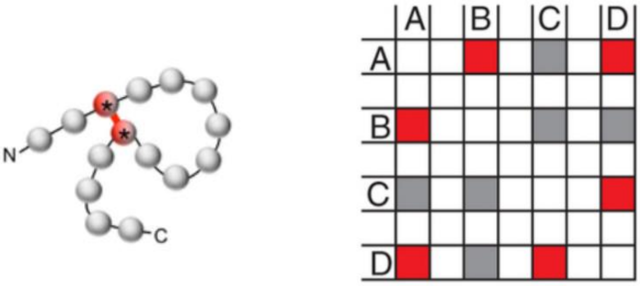
A C T C G C A A T A T G C T A G G C C A G C

A C T _ _ _ _ T T A T G C T A T G C _ _ G C

Figure 1 – Alignment between two DNA sequences. The process is equal for protein sequences. Source: [5]

Given a protein sequence, we can query protein databases in order to find homologs (similar proteins). The comparison between sequences is done using alignment methods. Figure 1 shows an example of an alignment between two sequences of DNA; however, the process is similar for protein sequences. By aligning several sequences, one can estimate the degree of similarity between them. Aligning multiple sequences for a multiple sequence alignment (MSA). An MSA is a series of alignments between three or more proteins from which homology can be inferred [6].

When a protein folds, residues come together as a consequence of hydrogen bonding, Coulombic and van der Waals interactions [7]. These residues are said to be in contact, which means that they are closer than a certain threshold, normally under 8 angstroms (Å) [8]. Figure 2 (left) shows an illustration of two residues in contact (highlighted in red). Contact between residues constraint the final 3D shape of a protein. Therefore, the



	A		B		C		D
A			🟥		▨		🟥
B	🟥				▨		▨
C	▨		▨				🟥
D	🟥		▨		🟥		

Figure 2 – Left: Two residues in contact are highlighted in red. Right: Contact map illustration. Source: [22]

aim of many methods is to generate a contact map which highlight contacts for every pair of residues. Figure 2 (right) shows a contact map example for a protein with four residues

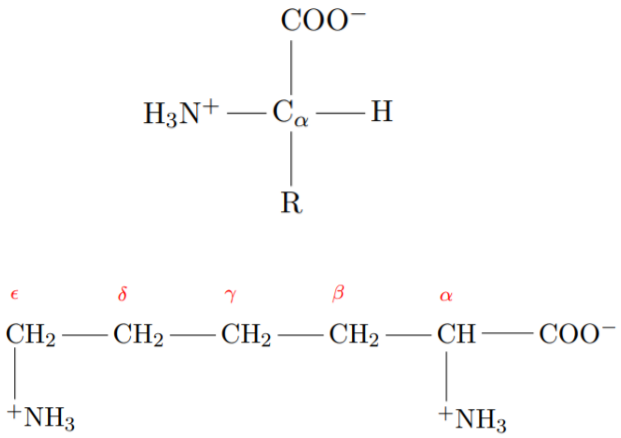


Figure 3 – Top: Basic structure of an amino acid, R denotes the side chain or variable group. Bottom: Lysine amino acid, additional carbons in the R-group are labelled from the C-alpha. Source: [34]

(A, B, C and D). Red squares indicate that two residues are in contact; meanwhile, grey squares mean residues are not in contact. Nevertheless, an amino acid is composed of multiple atoms as shown in Figure 3 (top), in which of them should we focus to predict contacts? C_β or the first carbon in the R-group, as shown in Figure 3 (bottom), is the preferred atom to study inter residue contacts since it constitute the most accurate representation of the 3D structure [9].

All of the methods that we will analyze in this review, AlphaFold, RaptorX and TripletRes, consist of four main steps. First, the input sequence is compared against databases to generate an MSA. From the MSA, features are extracted that will serve as the input to the third step, a deep neural network. The neural network will predict either contacts between residues or the exact distance depending on the method. Finally, the contact or distance information will be used to generate the protein fold. In the next section, we will briefly describe each of the methods before jumping into a comparison between them.

3. Method Description

Here, we will shortly describe AlphaFold, RaptorX and TripletRes as an introduction before jumping straight into the comparison between these methods.

3.1. AlphaFold

AlphaFold is a method developed by DeepMind focused on predicting the distances between the carbon atoms of the protein backbone. The distance map can be used to develop the 3D model of a certain protein with a high degree of accuracy. AlphaFold is a non-end-to-end differentiable method that consists of two steps. First, the prediction of the C_β-C_β distance and the torsion angles ϕ and ψ of each residue. Second, a structure realization step in which a potential based on the distance and torsion angle distribution is minimized by gradient descent until reaching the final protein conformation.

3.2. RaptorX

RaptorX encompasses several approaches, namely RaptorX-TBM, RaptorX-Contact and RaptorX-DeepModeller. These different approaches vary slightly on the databases used to construct the input features as well as the preprocessing of the input features. However, during this section we will consider only RaptorX-Contact. First, because it achieves the best results during the contact prediction competition in CASP13 out of the three RaptorX methods. And secondly, because it would be tedious for the reader to follow the subsequent discussion.

In Figure 4, we can see the main steps for protein structure prediction in the RaptorX pipeline. This pipeline can be divided into three main steps: feature extraction from protein sequence, learning step (marked in red) and the use of Crystallography & NMR System (CNS) software to generate the final protein structure prediction.

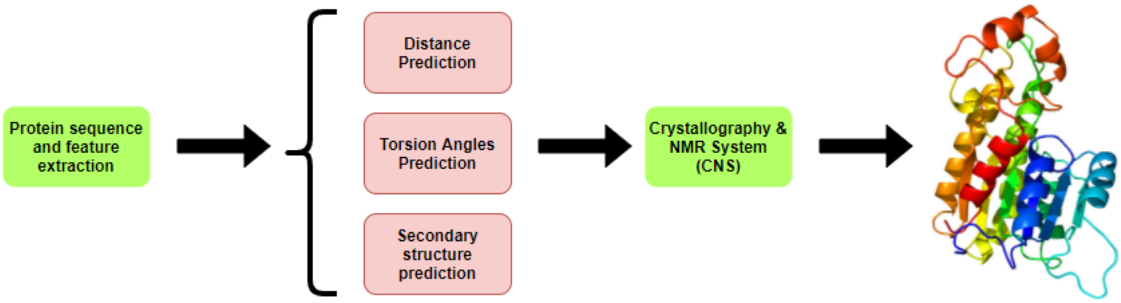

Protein sequence and feature extraction
Distance Prediction
Torsion Angles Prediction
Secondary structure prediction
Crystallography & NMR System (CNS)

Figure 4 – RaptorX pipeline for protein structure prediction. Source: Own figure

3.3. TripletRes

The method described in [3], introduces both TripletRes and ResTriplet. Figure 5 provides an overview of both methods. These are used to predict the contacts between residues of a protein sequence. However, we will only discuss TripletRes from here on, as it was the outperforming method in the CASP13 method out of the two. The pipeline consist of the extraction of coevolutionary features using homologs search, and an end-to-end learning step which outputs the contact matrix for every residue pair. Finally, the I-Tasser folding engine is used to generate the protein fold [10].

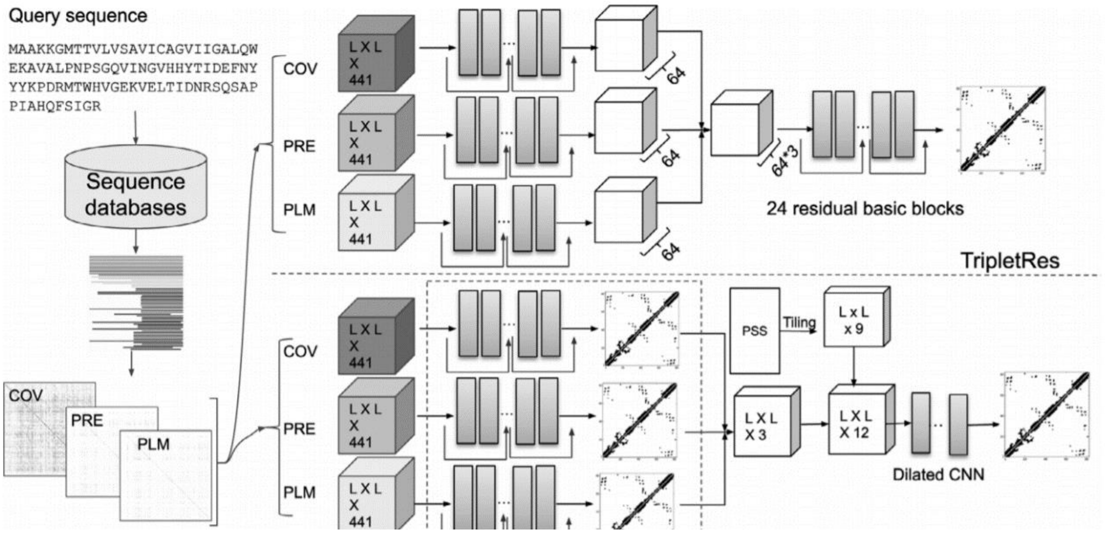


Figure 5 – Pipelines for the TripletRes and ResTriplet methods. Source: [3]

4. Comparison

In this section, we will compare the methods based on their inputs, outputs, architectures, and training schemes. Table 1 shows a comparative between these methods based on several criteria which will be useful along this review.

	AlphaFold	RaptorX	TripletRes
Inputs	MSA	MSA	MSA
Training instances	31,247 domains	11,410 proteins	7,671 domains
Architecture	2D ResNet	1D + 2D ResNet	2D ResNet
Network depth	220 blocks	7 (1D) + 60 (2D)	24*4 blocks
Outputs (pre-folding)	Distance map	Distance map	Contact map
Folding Engine	Gradient-based	CNS	I-Tasser

Table 1 – Comparison between AlphaFold, RaptorX and TripletRes. Source: Own Table

4.1. Inputs

Protein structure prediction depends heavily on sequence search of similar proteins. As seen in Table 1, the three methods use MSA to construct their inputs. However, there are differences on how the MSA is constructed as well as the features which are produced using the MSA.

While constructing the MSA, there are some design options that need to be addressed. For example, which database to use or how similar should sequences be to be included in the MSA. Another question to tackle is whether to construct the MSA for full proteins or domains. Certain proteins can be divided into domains which are short sections of proteins which can evolve, function and fold independently [11]. Since domains are shorter subsections of proteins, studying them separately allows to find more sequence homologs than for the whole protein. AlphaFold and TripletRes construct the MSA using domains while RaptorX uses whole sequences. AlphaFold extracts proteins from the Protein Data Bank (PDB) [12]. Then it extracts domains with less than 35% similarity between them using the CATH database [13]. On the contrary, RaptorX generates its training data from the PDB25, a protein database where sequences are no more than 25% similar. TripletRes extracts domains consisting of 30-400 residues and with less than 30% similarity from the SCOPe 2.07 database [14]. We can notice the abysmal difference between the training domains of AlphaFold versus TripletRes. Although both methods search for domains with less ~30% of redundancy, AlphaFold uses the CATH database which contain 500,238 domains while TripletRes uses the SCOPe database which contains approximately half of domains (276,231). Therefore, database choice may have had an impact in the final performance of each model.

Once sequences are collected, the MSA is generated for each training instance. AlphaFold searches for homologous sequences in Uniclust30 [15] using the HHblits [16] searching tool. RaptorX approach is similar, although four different MSA are generated for each training sequence. This are later used to generate four different predictions, which are averaged to obtain the final result. MSAs are generated searching in UniProt and UniClust30 using HHBlits and Jackhmmer [17], with different E-values (interpreted as the probability of observing homology matches by chance). On the contrary, TripletRes uses a more complex approach called DeepMSA [18]. DeepMSA is a three-step method which searches through three different databases. Depending on the number of effective sequences found at each step the method proceeds to the next step or not.

After MSAs are constructed, several features are generated that will serve as inputs to the deep learning step. TripletRes and AlphaFold create a Potts model from the MSA as input feature. The Potts model is a generalization of the Ising Model. It takes the following form:

$$H(S) = \sum_{i=1}^{L} h_i(s_i) + \sum_{i=1}^{L} \sum_{i<j}^{L} J_{ij}(s_i, s_j)$$

The sequence S encodes 21-states (20 states accounting for each aminoacid plus a state for a gap). $h_i(s_i)$ encodes the one-site marginal probabilities of the MSA while $J_{ij}(s_i, s_j)$ two-site marginal probabilities (i.e., the marginal probability of a pair of residues *i* and *j*) [19],[20], [21]. In Figure 6 (left), we can see an example of the AlphaFold input which represents a single slice in the z-direction of the Potts Model.

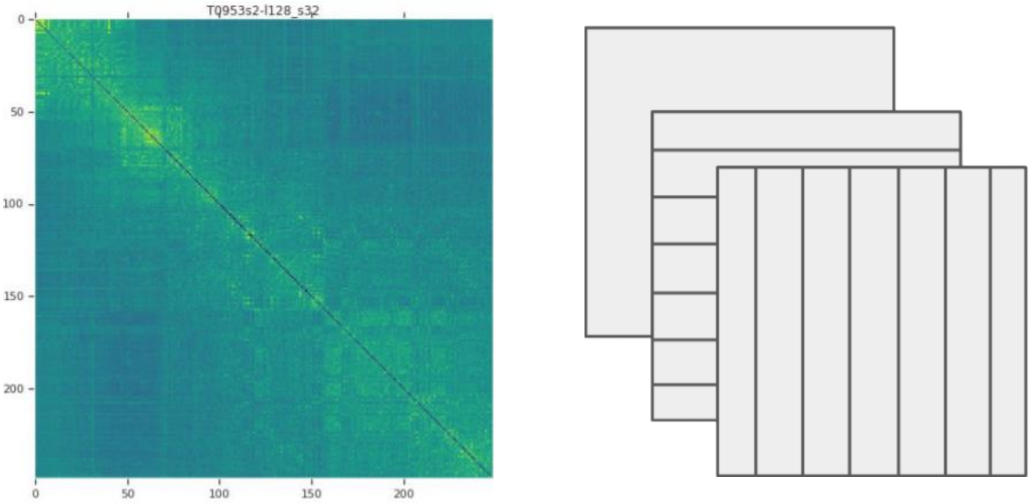


Figure 6 Left: Example of an input slice to the network. Right: One-dimensional features are repeated in x and y and concatenated to the rest of the input features. Source: [1]

As seen in Figure 5, TripletRes also uses as input the covariance matrix (COV) and its inverse, the precision matrix (PRE). The covariance matrix is defined as:

$$S_{ij}^{ab} = f_{i,j}(a,b) - f_i(a) f_j(b)$$

where $f_{i,j}(a,b)$ is the observed frequency of residue pair a and b at position i and j and $f_i(a)$ is the frequency of occurrence of a residue type a at position i. On the other hand, RaptorX uses the mutual information (MI) to study the correlation between residues. The MI is defined as:

$$MI = \sum_{i,j} f\left(A_i B_j\right) \frac{f\left(A_i B_j\right)}{f(A_i) f\left(B_j\right)}$$

where $f\left(A_i B_j\right)$ measures the combined frequency of two pairs of amino acids and $f(A_i)$ measures the frequency of a single amino acid. Mutual information has the disadvantage that even when residues are not connected in the 3D structure, there is some indirect mutual information between them [22].

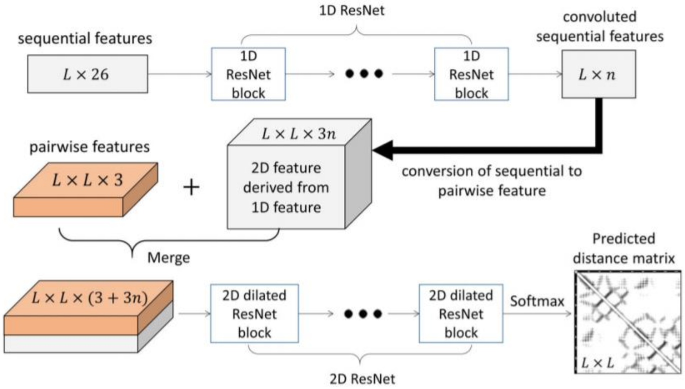


Figure 7 – RaptorX's deep learning distance prediction pipeline. Source: [2]

In contrast to TripletRes, AlphaFold and RaptorX use sequential features (1D features) on top of the pairwise features that have just been described. The biggest difference is how these 1D inputs are processed. While RaptorX uses a 1D residual network to process them as shown in Figure 7, AlphaFold repeats the 1D features in x and y before concatenating them to the pairwise (2D) features (Figure 6 left).

One important difference is how inputs are fed into the networks. RaptorX and TripletRes input the whole feature into the network. Meanwhile, AlphaFold uses a patch-based approach. The input is split in patches of 64x64 pixels. By sampling these patches with random offsets, each training sample can generate thousands of patches which acts as a mechanism against overfitting. Finally, at inference time, patches are averaged together, with patches around the center having a higher weight.

4.2. Architectures

As previously mentioned, RaptorX uses a combination of 1D and 2D neural networks to produce its final result. RaptorX sequential features are processed by a 1D ResNet block consisting of 7 convolutional layers with a kernel size of 15. This stage produces a set of L features, namely $v = v_1, v_2, \ldots, v_L$. These features are then converted to a 2D representation. For each pair of residues, i and j, the features $v_i, v_{(i+j)/2}, v_j$ are concatenated. Repeating this step for each pair of residues produces a LxLx3n feature map. Then, these features are processed by a 60-blocks neural network that produces the final output. Meanwhile, TripletRes handle each of its inputs by a separate residual network with 24 blocks each. These are followed by another 24-block network which outputs the final result. AlphaFold uses the deepest network, 220 blocks, which result in 21 M parameters. As it can be observed, all presented methods use residual networks as their architecture of choice. The building blocks of these residual networks vary from method to method. For example, AlphaFold makes use of three convolutional layers while TripletRes and RaptorX only use two per block. However, these differences are slight and prone to have been decided by a trial-and-error approach.

4.3. Output and Training Details

The output of the networks is where some of the biggest differences between methods lie. During the introduction, we have presented how contacts can help determine the 3D structure of a protein. However, contact determination is not as informative as it could be. Determining whether two residues are in contact or not, gives no information on the exact distance between them. We know they should be closer than a certain threshold, but not how close. Therefore, predicting the distance between each pair of residues would be much more informative. AlphaFold and RaptorX opt for this approach meanwhile TripletRes predicts the contact maps. Figure 8 (left) shows an example of the output of the network for a single pair of residues. AlphaFold method predicts the marginal distance distribution of distances between every pair of residues. To do so, the output of the network is divided into 64 equal bins in the range of 2-22 Å. Figure 8 (right) shows the mean of the distance histograms (distograms) for all residues. RaptorX uses a similar method. However, less bins are used to encode the final output. The predicted distance distribution between residues is discretized using 25 bins (<4.5Å, 4.5-5Å, 5-5.5Å, …, 15.5-16 Å, >16Å). Therefore, the output is encoded as a LxLx25 matrix.

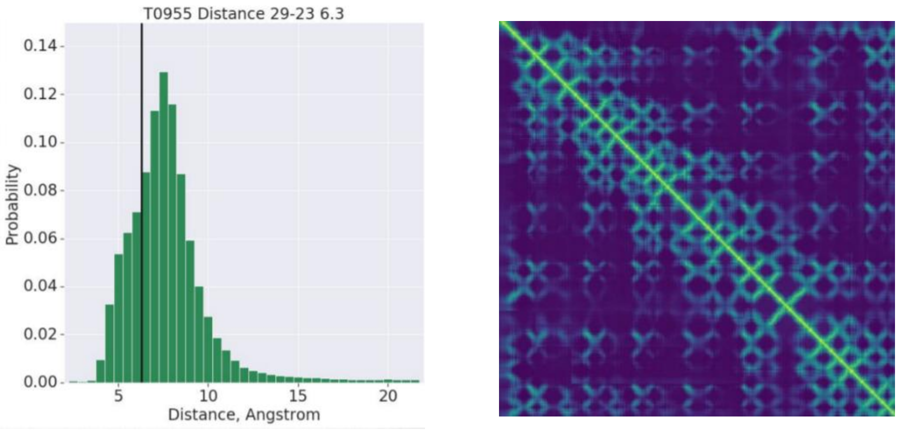


Figure 8 – Left: Distance distribution in AlphaFold for a pair of residues in protein T0955. Right: Mean of the distance distribution for all residue pairs (AlphaFold). Source: [1]

Apart from the distances, AlphaFold and RaptorX predict several other outputs that will be used in the protein folding step. However, while AlphaFold predicts these secondary outputs from the final activations of a unique network, RaptorX uses individual networks for each output. Torsion angles of each residue are predicted by both AlphaFold and RaptorX. As shown in Figure 9 (left), the torsion angles specify the position of the atoms with respect to each other. In both methods, only ϕ and ψ are predicted since ω is 180º in a majority of the cases. AlphaFold predicts the distribution of the torsion angles from the final activations. The probability of ψ and φ taking a certain value in the range -180º to + 180º is discretized in 10ºx10º bins, which results in 1,296 values. On the other hand, RaptorX uses an additional 1D ResNet to predict the torsion angles and the secondary structure. Although AlphaFold also predicts the secondary structure, this information is not used in their pipeline to produce the final folding.

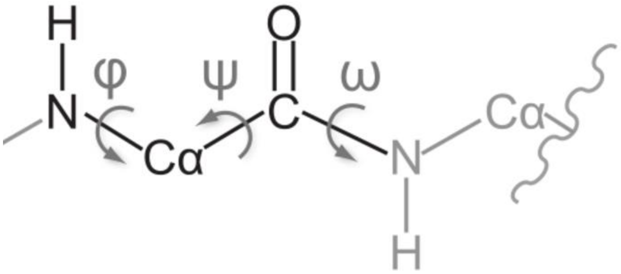


Figure 9 – Left: Torsion angles of a residue. Source: [35].

Additionally, to the distance between C_β-C_β atom pairs, RaptorX also predicts distances between other three atom pairs for each residue (i.e., C_α-C_α, C_g-C_g and N-O). Consequently, the final output contains more information on how the protein should fold. The distance distribution of atom pairs is done individually. Therefore, RaptorX repeats the pipeline shown in Figure 7 four times, one for each distance distribution.

Regarding their training scheme, AlphaFold and RaptorX use SGD as optimizer while TripletRes uses the Adam optimizer [23]. TripletRes and RaptorX are trained for ~300,000 steps while AlphaFold is trains for double, 600,000 steps. This could show that TripletRes and RaptorX are prematurely ended or more hyperparameter tweaking was needed. TripletRes performs a binary prediction (contact or not contact). Therefore, it the loss used is binary cross-entropy. AlphaFold and RaptorX use as cross-entropy as their loss. However, RaptorX uses a weighted version. The weighting scheme is introduced to compensate for the small number of contacts among all residue pairs.

All methods use ensemble techniques to improve accuracy and make the predictions more robust. AlphaFold uses four different networks with slightly different hyperparameters which are averaged together. On the contrary, RaptorX opts to generate slightly different MSAs (four) which are used to train different networks. TripletRes uses a 10-fold cross validation scheme to produce its final output.

Regarding computational power, AlphaFold is trained using a batch size of 32 split into 8 different GPUs. The training took 5 days for 600,000 steps. Meanwhile, TripletRes requires 4 GPUs with a varying batch size depending on the length of the protein. Unfortunately, this information was not available for RaptorX.

In summary, the biggest differences in this section are the prediction of distances by AlphaFold and RaptorX in comparison with the prediction of contacts by TripletRes. The networks present a marked difference in complexity. However, whereas AlphaFold uses one network to predict all their outputs, RaptorX trains five different networks for four atom distance pair prediction (i.e., C_β-C_β, C_α-C_α, C_g-C_g and N-O) plus one for torsion angle prediction. Therefore, a direct comparison in terms of complexity is not as straightforward. Nevertheless, a better metric to judge the complexity of each network would be the number of parameters; unfortunately, only AlphaFold discloses this information.

4.4. Protein Folding

Finally, we will analyze the folding mechanism of the three groups. Both RaptorX and TripletRes use folding engines to produce the final protein folding. These engines use the information generated by the neural networks in the previous step as constraints in their minimization pipeline. RaptorX constructs the final protein structure using the Crystallography & NMR System (CNS) which is a software for protein structure determination commonly used for computational biology. The learned distance distributions, backbone torsion angles and secondary structure are used as constraints for the CNS when predicting the protein folding. For each protein, out of the L^2 predicted distances between residues, only 7L pairs with the highest likelihood of having a distance <15Å are used as inputs for the CNS step. For these 7L residue pairs, lower (mean minus standard deviation) and upper (mean plus standard deviation) bounds are defined to constraint the final folding of the protein. For every protein, CNS creates 200 possible 3D models. The five with the least violation of restraints are chosen as the final models. In contrast, TripletRes uses the physical-based folding engine I-Tasser. In comparison to the CNS, the I-Tasser engine is more powerful producing better folds as a result. Both of these methods minimize a global energy. On the other hand, AlphaFold constructs a protein-specific potential using the information from the deep learning step (distance and torsion angles distributions) which is then minimized using gradient descent. In order to construct a differentiable potential, both the distance and the torsion angle distribution should be continuous functions. Therefore, AlphaFold constructs a smooth potential $V_{potential}$ from the distance distributions. To do so, the distance distributions are fitted to a cubic spline. A reference distribution which predicts the distance between residues is calculated in order to produce an unbiased potential. This reference state is trained using only the overall length of the protein and a binary feature $\delta_{\alpha\beta}$ if the residue is glycine or not (since glycine does not have a β carbon). Thus, the normalized distance potential is:

$$V_{distance}(\boldsymbol{x}) = -\sum_{i,j\ i\neq j} log\,P\left(d_{ij}\middle|S, MSA(S)\right) - log\,P\left(d_{ij}\middle|length, \delta\alpha\beta\right)$$

Then, a von Mises distribution is fitted to the marginal distribution of the torsion angles. The last term of the potential accounts for the van der Waals forces among residues. This term, named V_{score2_smooth}, prevents unnatural overlaps between two non-bonding atoms (known as called steric clashes) in the final protein structure. This term is introduces using the open-source framework Rosetta [24]. Therefore, the final potential is:

$$V_{total} = V_{distance}\big(G(\phi,\psi)\big) + V_{torsion}(\phi,\psi) + V_{score2\_smooth}\big(G(\phi,\psi)\big)$$

In the equation above, the term $G(\phi,\psi)$ represents the parametrization from the torsion angles to the backbone atom coordinates such that $\boldsymbol{x} = G(\phi,\psi)$. As a result, the potential defined by Equation 2 is fully differentiable with respect to the torsion angles. Therefore, it can be minimized using L-BFGS [25], a gradient descent algorithm. Since the minimization is conditional on the sampling from the distance and torsion distributions, sampling from the initial distributions is repeated to avoid getting stuck in local minima. In comparison to AlphaFold, RaptorX only uses the mean plus/minus the standard deviation instead of the whole predicted distribution, throwing away valuable information in the folding step. Both the CNS and I-Tasser minimize a global (not protein specific) function whereas AlphaFold potential is derived explicitly for the MSA of a given protein; then, it could have more expressivity in order to reach the final protein fold.

5. CASP13 Results Comparison

Throughout this review, we have explored three different methods for protein folding using deep learning, namely AlphaFold, RaptorX, and TripletRes. All these methods participated in the CASP13 protein folding competition making its comparison easier and more objective.

As a foreword, protein folding result comparison is a challenging topic. This is because metrics are not as straightforward as with other machine learning tasks such as image recognition. CASP, the leading competition for protein folding, is an example of this. In every competition, is not until after the submission when metrics and z-scores are defined by assessors. This is due to the fact that the proteins in which the assessment is performed, can be divided into Free Modelling (FM) or proteins which have few homologs and Template-Based Modelling (TBM) or proteins with a large number of homologs. However, this distinction is not clear, and there are several categories in between (e.g., TBM easy, TBM hard, TBM/FM, etc.). Therefore, until all proteins are analyzed by experts, proper and unbiased z-scores cannot be defined.

Additionally, the CASP13 competition has several rankings (e.g., regular targets, multimeric targets, contact predictions, etc.) and not all the three methods competed in all of them at the same time. For that reason, the methods will only be compared in the regular target section (which is the main competition ranking) and contact predictions[1].

[1] AlphaFold did not submit its results for the contact prediction in CASP13. However, they are described in [1].

In this section, we will analyze and compare the three presented methods according to their results in the CASP13 competition. In Figure 10, we see the results for AlphaFold, TripletRes and RaptorX in the regular target section of the CASP13 competition. Next to the method name, in between brackets, the position achieved by each method is shown.

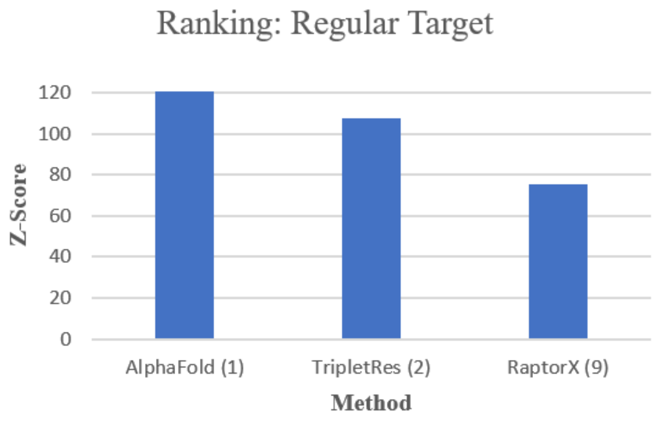
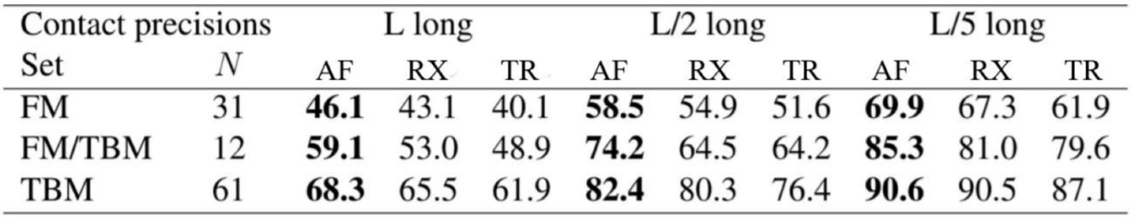


Figure 10 – Result of AlphaFold, TripletRes and RaptorX (DeepModeller) on the CASP13 competition. Source: Own figure based on CASP13 results

The regular target ranking considers the performance of these methods on 104 proteins. The concrete metric used varies depending if the proteins are suitable for FM, TBM or in between. AlphaFold achieves the first position obtaining an improve of ~12% over TripletRes (which achieved second position) and a 60% difference against RaptorX which achieved ninth position in this category. When only analyzing TBM models, TripletRes achieves the first position with a z-score of 63.27 followed by AlphaFold (second position) with a score of 62.47 and RaptorX (eighth position) with a score of 51.92. This shows that deep learning methods that do not use template information (like AlphaFold and TripletRes) can predict with high accuracy 3D protein structure. However, it could be argued that deep learning methods encode this information for all training proteins as long as it is complex enough. For FM targets, AlphaFold achieved the first position in terms of z-score and expert assessment. Human manual assessment concluded that AlphaFold predicted the best 3D configuration twice as many times as the next competitor.

Contact prediction is key when constructing the fold of a protein. For a large number of proteins, estimating as few as 8% is sufficient to reconstruct their folding [26], [27]. In order to make metrics length-invariant, it is usual to evaluate the most probable *L/l* contacts, where L is the length of the sequence. Therefore, if a protein has 150 residues, the L/5 metric will only evaluate the most probable 30 contacts. The second difficulty when evaluating contacts is that short-range contacts are easier to predict than long-range contacts. That is, predicting a contact between two residues closer together is easier than for two residues further apart in the sequence. Thus, contact evaluation is divided in short, medium or long-range predictions, being the latter the most used metric. In Figure 11, we can see the predicted long-term contacts (top L, L/2 and L/5) for the 104 proteins in the test set of the CASP13 competition. It can be seen that AlphaFold, RaptorX and TripletRes consistently occupy the first, second and third position for every metric and protein group.



Contact precisions		L long			L/2 long			L/5 long		
Set	*N*	AF	RX	TR	AF	RX	TR	AF	RX	TR
FM	31	**46.1**	43.1	40.1	**58.5**	54.9	51.6	**69.9**	67.3	61.9
FM/TBM	12	**59.1**	53.0	48.9	**74.2**	64.5	64.2	**85.3**	81.0	79.6
TBM	61	**68.3**	65.5	61.9	**82.4**	80.3	76.4	**90.6**	90.5	87.1

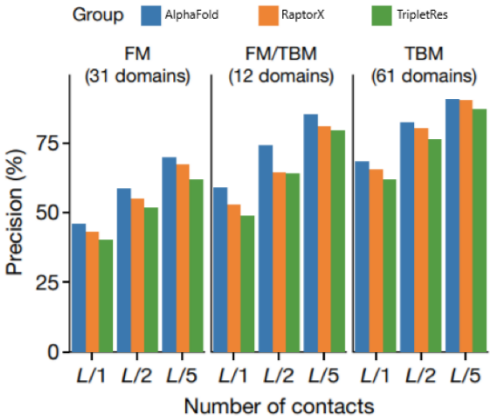


Figure 11 – Long-range contact prediction results for the CASP13 competition. Source: [1]

It is interesting that even though RaptorX contact prediction results were better than those obtained by TripletRes, their final folds perform much worse than the ones from TripletRes. From Figure 11 (above), we can see that the difference in contact prediction performance is slight between the best and worst results. However, when considering the protein fold, the performance difference is ~60%[2]. This stress the importance of the folding pipeline in the final output.

In summary, CASP13 results point out the dominance of AlphaFold in all tasks. In the next section, we will explore other work in the field of protein folding.

6. Other Work

In the CASP13 competition, there were 99 competitors. Two years later, in CASP14, the number of contenders increased to 146. This shows the increasing interest in the scientific community for the protein folding problem. In the regular target podium of CASP13, after AlphaFold and TripletRes, we find the MULTICOM method [28]. Similar to TripletRes, MULTICOM predicts the contact between residues instead of the distances. However, it uses a two-step deep learning approach. The first step predicts the probability maps at

[2] Difference between AlphaFold and RaptorX for regular targets in the CASP13 competition

different distance thresholds (i.e., 6, 7.5, 8, 8.5, and 10 Å). Then these are concatenated and processed by a convolutional neural network that outputs the final contact probability map at 8 Å distance threshold. Nevertheless, MULTICOM as the presented methods is not end-to-end differentiable. Other methods, such as NEMO (Neural energy modeling and optimization) do perform the sequence-to-structure process in an end-to-end differentiable manner [29]. Like AlphaFold, NEMO constructs protein-specific potentials. However, NEMO approach uses no co-evolutionary information (no homologs) and uses concepts of Langevin dynamics to produce the final fold. Many of the presented approaches center their efforts in predicting accurate contact or distance maps. However, as seen in the results section, it is arguable that the folding pipeline is even more important (of course given good enough contact results). Studies such as ProteinSolver [30], focus on the folding pipeline given contact maps. ProteinSolver models residues as a graph and connects them depending on the input contact map. Using Graph Neural Networks [31] the final protein fold can be generated. During this review, we have analyzed the main methods presented at CASP13. Nevertheless, it would not be complete without a mention to AlphaFold 2, the uncontested winner of CASP14. AlphaFold 2 improved on the second contender almost by a 300% [32]. Their improvement was so large, that even the organizers of CASP have called the protein folding problem solved [33]. Although there is not a publication yet, AlphaFold2 seems to be an end-to-end approach that generates the MSA features within the deep learning pipeline using a transformer model. Unfortunately, we will need to wait to hear the concrete details.

7. Conclusion

We have presented here a comparison between three methods for protein folding: AlphaFold, TripletRes and RaptorX. We have described the advances made in this field such as the introduction of more informative distance prediction (AlphaFold and RaptorX) or the use of protein-specific potentials (AlphaFold). AlphaFold has been proclaimed as the undoubted winner in this comparison. Nevertheless, RaptorX and TripletRes contribute significantly with their approaches.

In general, we have seen that the methods presented here are highly dependent on the input MSA (worse MSA leads to a worse result). Therefore, folding of proteins without any or few homologs is still a challenge that needs to be solved before any biological application (i.e., protein design). Nevertheless, AlphaFold2 seems to have tackled this problem and come up with a solution that will be ready to real life situations. With the solution of protein folding, other problems arise such as being sure of the calibration of these methods. Undoubtedly, breakthroughs in this field will lead to a better understanding of the machinery of life.

8. Bibliography

[1] A. W. Senior *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, Jan. 2020.

[2] J. Xu and S. Wang, "Analysis of distance-based protein structure prediction by deep learning in CASP13," *Proteins Struct. Funct. Bioinforma.*, vol. 87, no. 12, pp. 1069–1081, Dec. 2019.

[3] Y. Li, C. Zhang, E. W. Bell, D. Yu, and Y. Zhang, "Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13," *Proteins Struct. Funct. Bioinforma.*, vol. 87, no. 12, pp. 1082–1091, Dec. 2019.

[4] C. Branden and J. Tooze, *Introduction to protein structure*. 1991.

[5] P. J. A. Cock *et al.*, "Biopython: freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, Jun. 2009.

[6] M. Chatzou *et al.*, "Multiple sequence alignment modeling: Methods and applications," *Briefings in Bioinformatics*, vol. 17, no. 6. Oxford University Press, pp. 1009–1023, 01-Nov-2016.

[7] R. W. Newberry and R. T. Raines, "Secondary Forces in Protein Folding," *ACS Chem. Biol.*, vol. 14, no. 8, pp. 1677–1686, Aug. 2019.

[8] C. S. Miller and D. Eisenberg, "Using inferred residue contacts to distinguish between correct and incorrect protein models," *Bioinformatics*, vol. 24, no. 14, pp. 1575–1582, Jul. 2008.

[9] J. M. Duarte, R. Sathyapriya, H. Stehr, I. Filippis, and M. Lappe, "Optimal contact definition for reconstruction of Contact Maps," *BMC Bioinformatics*, vol. 11, no. 1, p. 283, May 2010.

[10] C. Zhang, S. M. Mortuza, B. He, Y. Wang, and Y. Zhang, "Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12," *Proteins Struct. Funct. Bioinforma.*, vol. 86, no. Suppl 1, pp. 136–151, Mar. 2018.

[11] S. Uchida, "Databases and software to make your research life easier," in *Annotating New Genes*, Elsevier, 2012, pp. 7–47.

[12] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nat. Struct. Mol. Biol.*, vol. 10, no. 12, p. 980, 2003.

[13] M. Knudsen and C. Wiuf, "The CATH database," *Hum. Genomics*, vol. 4, no. 3, pp. 207–212, Feb. 2010.

[14] N. K. Fox, S. E. Brenner, and J. M. Chandonia, "SCOPe: Structural Classification of Proteins - Extended, integrating SCOP and ASTRAL data and classification of

new structures,” *Nucleic Acids Res.*, vol. 42, no. D1, p. D304, Jan. 2014.

[15] M. Mirdita, L. von den Driesch, C. Galiez, M. J. Martin, J. Söding, and M. Steinegger, “Uniclust databases of clustered and deeply annotated protein sequences and alignments,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D170–D176, Jan. 2017.

[16] M. Remmert, A. Biegert, A. Hauser, and J. Söding, “HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment,” *Nat. Methods*, vol. 9, no. 2, pp. 173–175, Feb. 2012.

[17] L. S. Johnson, S. R. Eddy, and E. Portugaly, “Hidden Markov model speed heuristic and iterative HMM search procedure,” *BMC Bioinformatics*, vol. 11, no. 1, p. 431, 2010.

[18] C. Zhang, W. Zheng, S. M. Mortuza, Y. Li, and Y. Zhang, “DeepMSA: Constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins,” *Bioinformatics*, vol. 36, no. 7, pp. 2105–2112, Apr. 2020.

[19] R. M. Levy, A. Haldane, and W. F. Flynn, “Potts Hamiltonian models of protein co-variation, free energy landscapes, and evolutionary fitness,” *Current Opinion in Structural Biology*, vol. 43. Elsevier Ltd, pp. 55–62, 01-Apr-2017.

[20] M. Ekeberg, “Detecting contacts in protein folds by solving the inverse Potts problem - a pseudolikelihood approach,” 2012.

[21] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, “Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 87, no. 1, p. 012707, Jan. 2013.

[22] M. Liebrand and E. Marchiori, “Improvements in structural contact prediction: opportunities in prediction diculty and pairing preference of amino acids,” *Chemistry (Easton).*, 2014.

[23] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014.

[24] R. Das and D. Baker, “Macromolecular modeling with Rosetta,” *Annual Review of Biochemistry*, vol. 77. Annu Rev Biochem, pp. 363–382, 2008.

[25] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Math. Program.*, vol. 45, no. 1–3, pp. 503–528, Aug. 1989.

[26] R. Sathyapriya, J. M. Duarte, H. Stehr, I. Filippis, and M. Lappe, “Defining an essence of structure determining residue contacts in proteins,” *PLoS Comput. Biol.*, vol. 5, no. 12, p. 1000584, Dec. 2009.

[27] B. Adhikari and J. Cheng, “Protein residue contacts and prediction methods,” in *Methods in Molecular Biology*, vol. 1415, Humana Press Inc., 2016, pp. 463–476.

[28] J. Hou, T. Wu, R. Cao, and J. Cheng, “Protein tertiary structure modeling driven

by deep learning and contact distance prediction in CASP13," *Proteins Struct. Funct. Bioinforma.*, vol. 87, no. 12, pp. 1165–1178, Dec. 2019.

[29] J. Ingraham, A. Riesselman, C. Sander, D. Marks, and H. M. School, "Learning Protein Structure With a Differentiable Simulator," *International Conference on Learning Representations*, 2019.

[30] A. Strokach, D. Becerra, C. Corbi-Verge, A. Perez-Riba, and P. M. Kim, "Fast and Flexible Protein Design Using Deep Graph Neural Networks," *Cell Syst.*, vol. 11, no. 4, pp. 402-411.e4, Oct. 2020.

[31] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv*, Jun. 2018.

[32] "Groups Analysis: zscores - CASP14." [Online]. Available: https://predictioncenter.org/casp14/zscores_final.cgi. [Accessed: 09-Dec-2020].

[33] K. Noble, "Artificial intelligence solution to a 50-year-old science challenge could 'revolutionise' medical research." [Online]. Available: https://predictioncenter.org/casp14/doc/CASP14_press_release.html. [Accessed: 14-Dec-2020].

[34] P. J. Butterworth, "Lehninger: principles of biochemistry (4th edn) D. L. Nelson and M. C. Cox, W. H. Freeman &amp; Co., New York, 1119 pp (plus 17 pp glossary), ISBN 0-7167-4339-6 (2004)," *Cell Biochem. Funct.*, vol. 23, no. 4, pp. 293–294, Jul. 2005.

[35] C. Baldauf and M. Rossi, "Going clean: Structure and dynamics of peptides in the gas phase and paths to solvation," *Journal of Physics Condensed Matter*, vol. 27, no. 49. Institute of Physics Publishing, p. 493002, 24-Nov-2015.