


RESEARCH ARTICLE

rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments

Claudio Mirabello, Björn Wallner *

IFM Bioinformatics, Linköping University, Linköping, Sweden

* bjorn.wallner@liu.se

Abstract

In the last decades, huge efforts have been made in the bioinformatics community to develop machine learning-based methods for the prediction of structural features of proteins in the hope of answering fundamental questions about the way proteins function and their involvement in several illnesses. The recent advent of Deep Learning has renewed the interest in neural networks, with dozens of methods being developed taking advantage of these new architectures. However, most methods are still heavily based pre-processing of the input data, as well as extraction and integration of multiple hand-picked, and manually designed features. Multiple Sequence Alignments (MSA) are the most common source of information in *de novo* prediction methods. Deep Networks that automatically refine the MSA and extract useful features from it would be immensely powerful. In this work, we propose a new paradigm for the prediction of protein structural features called rawMSA. The core idea behind rawMSA is borrowed from the field of natural language processing to map amino acid sequences into an adaptively learned continuous space. This allows the whole MSA to be input into a Deep Network, thus rendering pre-calculated features such as sequence profiles and other features calculated from MSA obsolete. We showcased the rawMSA methodology on three different prediction problems: secondary structure, relative solvent accessibility and inter-residue contact maps. We have rigorously trained and benchmarked rawMSA on a large set of proteins and have determined that it outperforms classical methods based on position-specific scoring matrices (PSSM) when predicting secondary structure and solvent accessibility, while performing on par with methods using more pre-calculated features in the inter-residue contact map prediction category in CASP12 and CASP13. Clearly demonstrating that rawMSA represents a promising development that can pave the way for improved methods using rawMSA instead of sequence profiles to represent evolutionary information in the coming years. **Availability:** datasets, dataset generation code, evaluation code and models are available at: <https://bitbucket.org/clami66/rawmsa>.

 OPEN ACCESS

Citation: Mirabello C, Wallner B (2019) rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments. PLoS ONE 14(8): e0220182. <https://doi.org/10.1371/journal.pone.0220182>

Editor: Yang Zhang, University of Michigan, UNITED STATES

Received: January 17, 2019

Accepted: July 10, 2019

Published: August 15, 2019

Copyright: © 2019 Mirabello, Wallner. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The data underlying the results presented in the study are available from <https://bitbucket.org/clami66/rawmsa>.

Funding: BW was supported by Swedish Research Council grant, 2016-05369. CM was supported by The Foundation Blanceflor Boncompagni Ludovisi, née Bildt (no grant number). CM and BW were supported by Nvidia Corporation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

1 Introduction

Predicting the 3D-structure of a protein from its amino acid sequence has been one of the main objectives of structural bioinformatics for decades now [1], yet a definite solution has not

Competing interests: The authors declare the following competing interest: The fact that we have received support from Nvidia Corporation does not alter our adherence to PLOS ONE policies on sharing data and materials.

been found yet. The most reliable approaches currently involve *homology modeling*, which allows a known protein structure to be assigned an unknown protein provided that there is a detectable sequence similarity between the two. When homology modeling is not viable, *de novo* techniques are needed, either based on physical-based potentials [2] or knowledge-based potentials [3–6]. In the first case, an energy function is used to estimate the free energy of a given protein conformation along with a search function that tries different conformations in order to minimize the energy function [7]. Unfortunately, even small relatively small proteins have many degrees of freedom making it prohibitively expensive to fold them even on customized computer hardware [8]. Knowledge-based potentials, on the other hand, can be learned using statistics [3] or machine learning [9] to infer useful information from known examples of protein structures. This information can be used to constrain the problem, thus greatly reducing the conformational search space and enable prediction of larger proteins and complexes.

In the last couple of decades, a variety of machine learning methods have been developed to predict a number of structural properties of proteins: secondary structure (SS) [10–15], relative solvent accessibility (RSA) [15–18], backbone dihedrals [19], disorder [15, 20, 21], disorder-to-order transition [22, 23], contact maps [24–27], and model quality [9, 28–30].

The most important information used by most (if not all) methods above is a multiple sequence alignment (MSA) of sequences homologous to the target protein. The MSA consists of aligned sequences and to allow for comparisons and analysis of MSAs, they are often compressed into position-specific scoring matrices (PSSM), also called sequence profiles, using the fraction of occurrences of different amino acids in the alignment for each position in the sequence. The sequence profile describes the available evolutionary information of the target protein and is better than a single sequence representation, often providing a significant increase in prediction accuracy [31, 32]. An obvious limitation of compressing an MSA into a PSSM is the loss of information that could be useful to obtain better predictions. Another potential issue is that whenever the MSA contains few sequences, the statistics encoded in the PSSM will not be as reliable and the prediction system may not be able to distinguish between a reliable and an unreliable PSSM.

SS, RSA and similar structural properties are sometimes used as intermediate features to constrain and guide the prediction of more complex properties in a number of methods [33–35]. An example of this comes from the methods used for the prediction of inter-residue contact maps, where evolutionary profiles are integrated with predicted SS and RSA to improve performance [36–38].

More recently, contact map prediction methods have been at the center of renewed interest after the development of a number of techniques to analyze MSAs in search of direct evolutionary couplings. These methods have led to a big leap in the state of the art [39–41]. However, their impressive performance is correlated with the number of sequences in the MSA, and is not as reliable when few sequences are related to the target. This means that evolutionary coupling methods have not completely replaced older machine learning-based systems, but have been integrated, usually in the form of extra inputs, along with the previously mentioned sequence profiles, SS and RSA, into even more complex machine learning systems. At the same time, the Deep Learning has proved to be a useful tool for better integrating the growing number and complexity of input features [42–45].

However, one might argue that this kind of integrative approach, combining individually derived features, ignores a key aspect of deep learning, i.e. that features should be automatically extracted by the network rather than being provided to the network as inputs [46]. If we wanted to take full advantage of deep learning by using it in the same way it is employed for tasks such as image classification, one idea could be to provide a raw MSA input. Since the

MSA is the most basic, lowest level input that methods use, it would make sense not to compress it into profiles, but instead let the deep network extract features as part of the training. However, a MSA is not an image or an audio track, and there is no native way of feeding such a large block of strings as input to a deep network.

In this work we try to overcome this hurdle and introduce a new system for the *de novo* prediction of structural properties of proteins called rawMSA. The core idea behind rawMSA borrowed from the field of natural language processing a technique called *embedding* [47], which we use to convert each residue character in the MSA into a floating-point vector of variable size. This way of representing residues is adaptively learned by the network based on *context*, i.e. the structural property that we are trying to predict. To showcase the idea, we designed and tested several deep neural networks based on this concept to predict SS, RSA, and Residue-Residue Contact Maps (CMAP).

2 Methods

2.1 Inputs

Unlike the classical machine learning methods for the prediction of protein features, rawMSA does not compress the Multiple Sequence Alignment into a profile but, rather, uses the raw aligned sequences as input and devolves the task of extracting useful features to the deep network. The input to the deep network is a flat FASTA alignment. Before it is passed to the input layer of the neural network, each letter in the input is mapped to an integer ranging from 1 to 25 (20 standard residues plus the non-standard residues *B*, *U*, *Z*, *X* and *-* for gaps). If the alignment of a protein of length L contains N sequences, including the target, or “master” sequence, it translates to an array of $L \times N$ integers. The master sequence occupies the first row of the array, while the following rows contain all the aligned sequences, in the order of output determined by the alignment software. Since MSAs for large protein families can contain up to tens of thousands of sequences, a threshold is set so that no more than Y sequences are used. For details on the alignment depth threshold, see the “Architecture” section.

When training on or predicting SS or RSA, a sliding window of width 31 is applied to the MSA so that L separate windows of size $31 \times Y$, one for each residue in the master sequence, are passed to the network. The central column in the window is occupied by the residue in the master sequence for which a prediction is being made and the corresponding aligned residues from the other sequences. Zero-padding is applied at the N- and C- terminals of the master sequence or if the master sequence is shorter than the window size, and at the bottom if the number of aligned sequences is smaller than the maximum alignment depth Y . Note that residues are mapped to integers larger than zero and do not interfere with zero-padding.

2.2 Architecture

We developed two different architectures for three different applications SS-RSA for the SS and RSA prediction and CMAP for the contact map prediction. In Fig 1 we show an example of the network architecture. The networks trained in this work might use different numbers of convolutional, fully connected or BRNN layers, as well as slightly different parameters, but they all share this same basic overall structure.

Since with rawMSA we abandoned the use of sequence profiles, which are also useful to represent the amino acid information in a computer-friendly format, i.e. a matrix of floating points, we needed to come up with a way of representing the input. In this case, where the inputs can be very large (up to hundreds of thousands of amino acids), categorical data cannot be translated with sparse, memory inefficient techniques such as *one-hot encoding*.

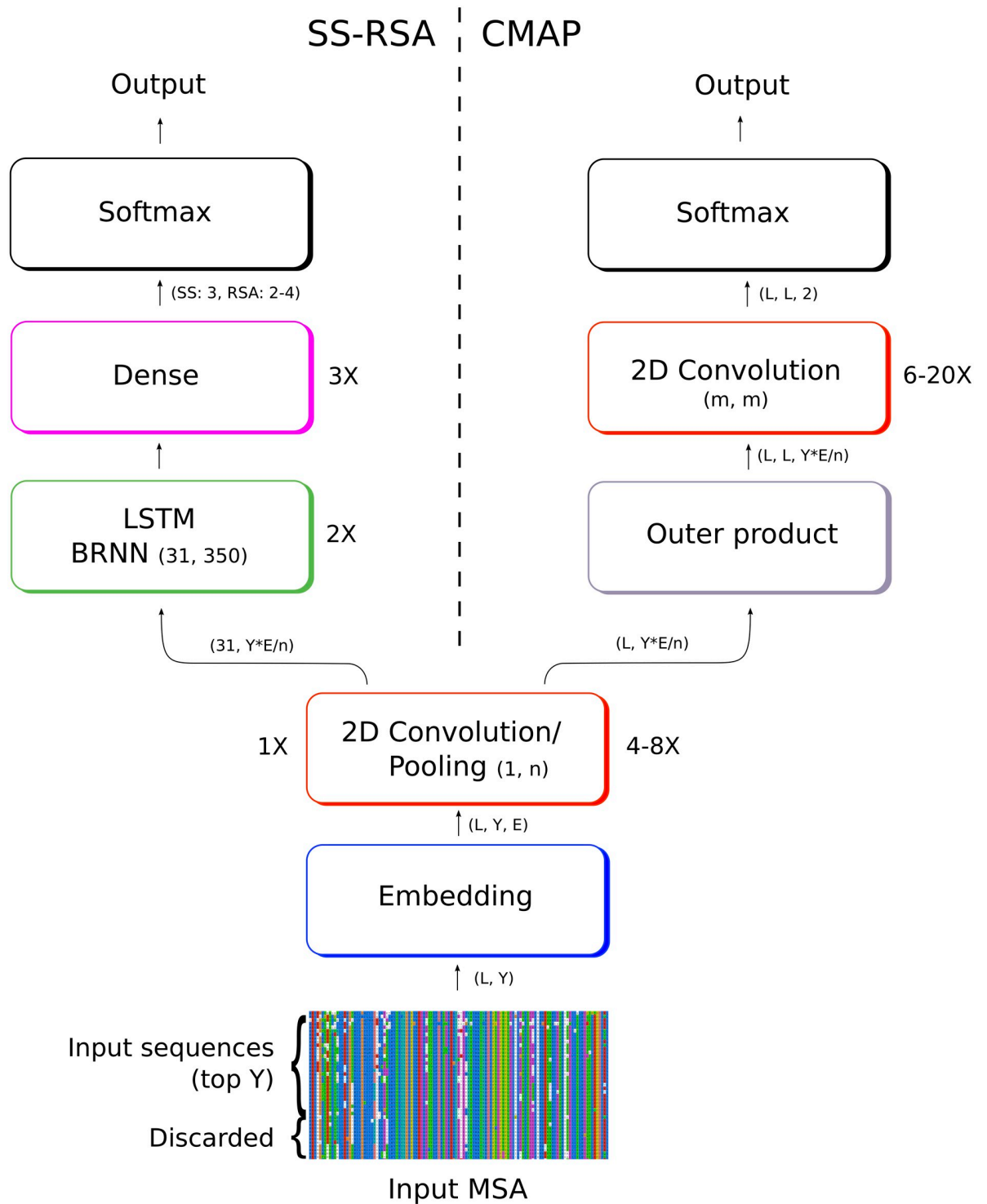


Fig 1. Network architecture for two rawMSA networks. On the left, the *SS-RSA* network predicts the secondary structure and relative solvent accessibility of each amino acid; on the right the *CMAP* network predicts the full contact map of the protein. The first layers are in common between the *SS-RSA* and *CMAP* architectures, although with slightly different settings, and provide the basis for the rawMSA approach.

<https://doi.org/10.1371/journal.pone.0220182.g001>

To resolve this issue, the first layer of rawMSA is a trained, shallow two-layer neural network called an *embedding layer*. Embeddings are a compact way of representing a discrete set of inputs into a continuous space [47]. This technique is widely used in natural language processing where the inputs are made of a sequence of words taken from a dictionary and mapped to an n -dimensional space in a vector of floats of size n . When dealing with word embeddings in natural language, words that represent similar concepts, at least in a certain context, will be in close proximity in the output space. A similar idea might also be very useful when dealing with the discrete set of amino acids [48], since they also share context-dependent similarities. For example, when it comes to the context of secondary structure, if we look at the Chou-Fasman amino acid propensities table [49], glutamic acid and methionine are both strongly associated with alpha-helices, so it might be useful that such amino acids are represented by similar vectors when predicting secondary structure.

The embedding layers in rawMSA output a vector of size E ranging from from 10 to 30, depending on the model, for each input residue in the alignment. In general, we have found that larger embedding vectors tend to give better results. The embedding layer is used both in the SS-RSA and the CMAP networks.

2.2.1 SS-RSA. In the case of SS-RSA, a 2D convolutional layer is stacked on top of the embedding layer, followed by a max pooling layer. The convolutional layer has a number of filters equal to the dimensionality of the embedding space. The convolution filters have the shape of column vectors, rather than square matrices as is usually the case, thus the size of the convolution windows varies between 1×10 to 1×30 depending on the model. This means that convolution is performed along each column in the MSA and the information does not spread across columns (i.e. across adjacent residues in the input sequence). Pooling is performed selecting the maximum value in a window of the same size. In this way, if the dimension of the input is 31 residues by 500 alignments before embedding and $31 \times 500 \times 10$ after embedding, this is reduced to a vector of size $31 \times 50 \times 10$ following the convolution and pooling layers, if the convolution and pooling windows are of size 1×10 . The convolutional and pooling layers are followed by a stack of two Long Short-Term Memory (LSTM) bidirectional recurrent layers, where each LSTM module contains 350 hidden units. The final three layers are fully connected, with softmax layers to output the classification prediction for SS (3 classes: Helix, Strand, Coil) or RSA (4 or 2 classes), depending on the model. Dropout is applied after each recurrent or dense layer to avoid overfitting, with variable fractions of neurons being dropped depending on the model (0.4 to 0.5). All the convolutional layers have ReLU activations and the outputs are zero-padded to match the two first dimensions of the inputs.

2.2.2 CMAP. For CMAP, the network predicts a whole contact map of size $L \times L$ for a protein of length L . The input, in this case, is not split in windows, but we use the whole width MSA at once, while the depth is cut at the Y top alignments. The network for the first part of CMAP is similar to SS-RSA, with an embedding layer followed by up to six (rather than only one) 2D convolution/max pooling layers along each MSA column. In this case though, the output of the network is a contact map with shape $L \times L$, so the preceding layers should represent the interaction between pairs of residues. This change of dimensionality is performed with a custom layer that performs the outer product from the output of the first stack of convolutional layers (H):

$$OP = \hat{H} \bar{H} \quad (1)$$

Where H has dimensionality $(L \times F \times S)$, \hat{H} and \bar{H} are obtained by adding singleton dimensions to H ($(L \times 1 \times F \times S)$ and $(1 \times L \times F \times S)$ respectively).

This operation generates a four-dimensional hidden tensor OP of shape $L \times L \times F \times S$, where F and S are the last two dimensions of the hidden tensor before the outer product. This output is then reshaped to a three-dimensional tensor of shape $L \times L \times (F * S)$ and passed to a new stack of six to 20 (depending on the model) 2D convolutional layers with squared convolutions of varied size (3x3, 5x5, 10x10) and number of filters (10 to 50). The last convolutional layer has shape $L \times L \times 2$ and is followed by a softmax activation layer that output the contact prediction with a probability from 0 to 1. All the convolutional layers have ReLU activations and the outputs are zero-padded to match the two first dimensions of the inputs. Batch normalization is performed in the outputs of the convolutional layers in the CMAP network.

2.3 Training

rawMSA was implemented in Python using the Keras library [50] with TensorFlow backend [51]. Training and testing were performed on computers equipped with NVIDIA GeForce 1080Ti, Tesla K80, and Quadro P6000 GPUs.

The training procedure was run including one protein in each batch block, regardless of the size, and using an RMSprop optimizer with sparse categorical cross-entropy as loss function. The SS-RSA network was trained for five epochs, while the CMAP network was trained for up to 200 epochs. During training, a random 10% of the training samples were reserved for validation, while the rest were used for training. After training, the model with the highest accuracy on the validation data was used for testing.

2.4 Data sets

The data set is composed of protein chains extracted from a 70% redundancy-reduced version of PDB compiled by PISCES [52] in April 2017 with minimum resolution of 3.0 Å and R-factor 1.0. This set contains 29,653 protein chains.

2.4.1 Avoiding homolog contamination. When training our networks, we want to make sure that the testing and training sets are rigorously separated so that no protein in the test set is too similar to any protein that the network has already “seen” during the training phase.

While most secondary structure and solvent accessibility prediction methods have been using 25–30% sequence identity as the threshold to separate testing and training sets [53–56], this practice has been discouraged as it has been shown that it is not sufficient to avoid information leakage [57]. This is apparently also valid for raw MSA inputs and ur tests separating sets at 25% sequence identity yields higher accuracies compared to our final results (data not shown).

To correctly split training and testing sets, we used two databases based on a structural classification of the proteins: ECOD [58] and SCOPe [59] databases to assign one or more superfamilies to each of the protein chains in the initial set. Then, we removed any chains that were related to more than one superfamily. The set generated from ECOD contains 16,675 proteins (ECOD set), while the one generated from SCOPe contains 9885 proteins (SCOPe set). The SCOPe set contains fewer proteins than the ECOD set since SCOPe has a lower coverage of the PDB. We split each set into five subsets by making sure that no two proteins from the same superfamily were placed in two separate subsets. This ensured that the respective MSA inputs would not be too similar to each other and is the recommended practice when training neural networks using sequence profiles, which are extracted from MSAs [57].

We used the SCOPe subsets to perform five-fold cross-validation on SS-RSA. We also used one of the ECOD subsets to train and validate CMAP, where the validation set was used to determine when to stop training to avoid overfitting, and to select the models that would be ensembled and tested, i.e. the models with the lowest validation error.

2.4.2 Multiple Sequence Alignments. The MSAs for both SS-RSA and CMAP were obtained with HHblits [60] by searching with the master sequence against the HMM database clustered at 20% sequence identity from February, 2016 for three iterations, with 50% minimum coverage, 99% sequence similarity threshold, and 0.001 maximum E-value. We also obtained a second set of MSAs by running JackHMMER [61], for three iterations and $1e - 3$ maximum E-value on the UniRef100 database from February 2016. The HHblits alignments were used to train and test the SS-RSA networks. The HHblits alignments were also used to train the CMAP network, while both the HHblits and the JackHMMER alignments were used as inputs when ensembling CMAP networks (see Ensembling section), as it improved the prediction accuracy. This approach was also tried for the SS-RSA network, but no improvement was observed.

2.4.3 Test sets. We labeled the CMAP data sets by assigning a native contact map to each protein. A contact was assigned to a pair of residues in a protein if the Euclidean distance between their $C\beta$ atoms ($C\alpha$ for Gly) in the crystal structure from the PDB was lower than 8 Å. Otherwise, the two residues were assigned the non-contact label.

We tested CMAP on the CASP12 RR (Residue-Residue) benchmark [62], which is composed of 37 protein chains/domains of the Free Modeling class (FM), i.e. protein targets for which no obvious protein homologs could be found at the time of the experiment (May-August 2016). To ensure a fair comparison with the predictors which participated in CASP12, we performed the benchmark in the same conditions to which all the other predictors were subjected at the time of the CASP experiment. We made sure that all protein structures (from Apr, 2016) in the training set (Apr, 2016) and sequences (from Feb, 2016) in the HHblit HMM and UniRef100 databases were released before CASP12 started.

To test SS-RSA, we calculated the secondary structure (SS) and the Relative accessible Surface Area (RSA) with DSSP 2.0.4 [63]. We reduced the eight SS classes (G, H, I, E, B, S, T, C) to the more common three classes: Coil, Helix, Extended (C, H, E). We used the theoretical Maximum Accessibility Surface Area (Max ASA) defined in [64] to calculate the RSA from the absolute surface areas (ASA) in the DSSP output and we used [0, 0.04], (0.04, 0.25], (0.25, 0.5], (0.5, 1] as thresholds for the four-class RSA predictions (Buried, Partially Buried, Partially Accessible, Accessible), and [0, 0.25], (0.25, 1] as thresholds for the two-class RSA predictions (Buried, Accessible). We discarded the proteins for which DSSP could not produce an output, as well as those that had irregularities in their PDB formats. The final set contained 9,680 protein chains.

2.4.4 Quality measures. The measure of the performance of the trained ensemble of SS-RSA networks is the three-class accuracy (Q3) for SS and the four-class and two-class accuracy for RSA, which are calculated by dividing the number of correctly classified residues by the total number of residues in the dataset.

CMAP predictions for the CASP12 RR benchmark set were evaluated in accordance with the CASP criteria by calculating the accuracy of the top $L/5$ predicted long-range contacts, where L is the length of the protein, and the long-range contacts are contacts between residues with sequence separation distance over 23.

2.4.5 Ensembling. Ensembling models usually yield a consensus model that performs better than any of the networks included in the ensemble [65]. Several networks both for CMAP and SS-RSA have been trained with different parameters (see “Results” section). Even though some models have worse performances on average, they are still saved. All the saved models that have been trained on the same set are used at testing time. The outputs from each model are ensembled to determine the final output. This is done by averaging all outputs from the softmax units and selecting the final class by picking the class with the highest average probability.

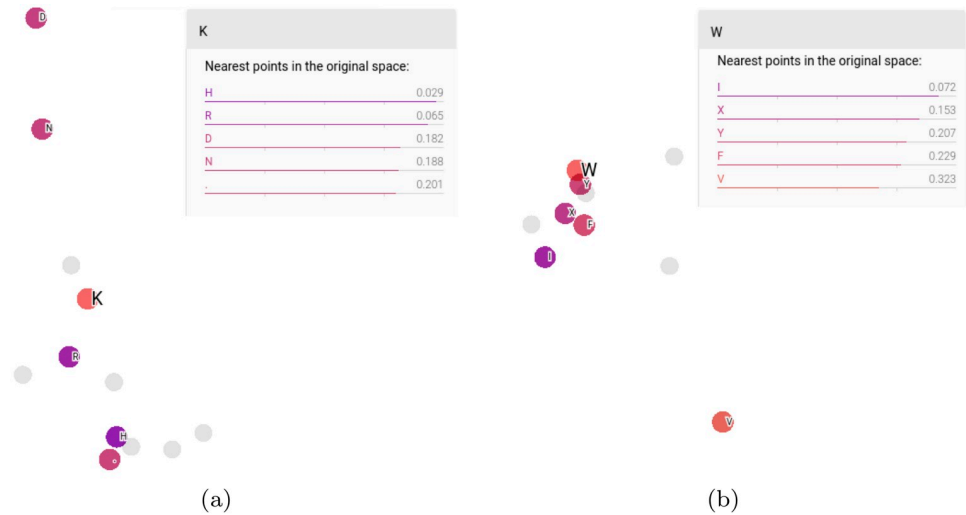


Fig 2. 2D PCA of the space of the embedded vectors representing the single residues. In this example, we show the embedding outputs of a simpler network where the original space has a dimensionality of four. The residues that are closest (lower cosine between the 4D vectors) to (a) lysine and (b) tryptophan are colored (the closer the residue, the darker the hue).

<https://doi.org/10.1371/journal.pone.0220182.g002>

In the CMAP case, each model in the ensemble is used to make two predictions for each target using either HHblits or JackHMMER alignment. Although the CMAP network is trained only on HHblits alignments, using the JackHMMER alignments in the ensemble improved the overall accuracy of the predictor.

3 Results and discussion

3.1 Embeddings

Tensorboard in Tensorflow can be used to visualize the output of the embeddings layer to see how each residue type is mapped on the output space. In Fig 2 we show the embeddings outputs for an early version of the SS-RSA network where each amino acid type is mapped on a 4D space. The 4D vectors are projected onto a 2D space by principal component analysis on Tensorboard. In the figure, the amino acids that are the closest (lowest cosine between the 4D vectors) to lysine (Fig 2a) and tryptophan (Fig 2b) are highlighted. The amino acids closest to lysine (K) are histidine (H) and arginine (R), which makes sense, since they can all be positively charged. Similarly, the residues closest to the hydrophobic tryptophan (W) are also hydrophobic, indicating that the embeddings can discriminate between different kinds of amino acids and map them onto a space that makes sense from a chemical point of view.

3.2 SS and RSA predictions

We have used the five-fold cross-validation results to determine the testing accuracy for SS-RSA. To compare the rawMSA approach against a classic profile-based method, we trained a separate network by removing the bottom layers from the SS-RSA network (embedding and first 2D convolution/Pooling layer) and we trained it by using the PSSMs calculated from the HHblits alignments as inputs (PSSM network). In order to assess the usefulness of amino acid embeddings, we also train a rawMSA neural network without embedding layer, where the input alignments are transformed using a simple one-hot encoding of each amino acid (One-hot100 network). In this case, because of limits imposed by the amount of memory available

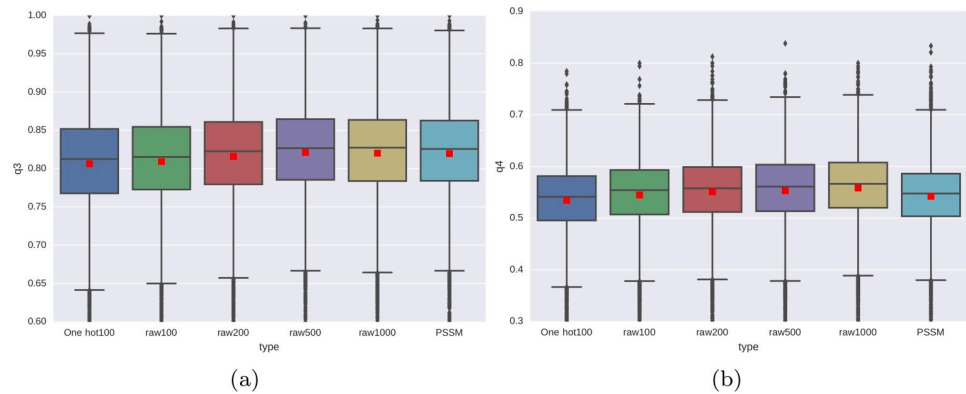


Fig 3. Per target secondary structure (a) and four-class solvent accessibility (b) accuracy for predictions using one hot encoding, a number of rawMSA networks, and a classical PSSM network trained and tested on the same dataset. One hot100 skips the rawMSA embedding step and encodes the alignments using one hot encoding, limited to the top100 alignments for memory reasons. Four different rawMSA networks are tested at variable MSA depths, using top 100, 200, 500 or 1000 alignments from the MSAs as input to the SS-RSA network. The average accuracies are shown as red squares.

<https://doi.org/10.1371/journal.pone.0220182.g003>

on our testing machine, we had to limit the one-hot network to 100 alignments only. We tested and trained the PSSM and one-hot network in the same way we trained the other networks, both for SS and RSA. In Fig 3 we compare the performance of the PSSM and one-hot network against several rawMSA networks trained on different numbers of input sequences (100 to 1000 MSA sequences). The boxplot shows how the rawMSA networks with more input sequences perform generally better, with the rawMSA500 and rawMSA1000 networks performing slightly better than the classic PSSM network in predicting secondary structure, and all the rawMSA networks outperforming the PSSM network in predicting solvent accessibility. We also show that the rawMSA100 network outperforms the One-hot100 network both in the SS and RSA experiments.

The final SS-RSA network is an ensemble of six networks trained in five-fold cross-validation on 100 to 3000 input MSA sequences per target.

The results are shown in Table 1.

It is difficult to make a direct comparison of rawMSA against other predictors in literature because of inevitable differences in the datasets. One example of this comes from secondary structure prediction systems, which have recently been reported to predict at accuracies (Q3) of up to 84% [66], yet we have not been able to find a recent study where the reported accuracy is supported by a proper splitting of the training and testing sets (see “Avoiding homolog contamination” paragraph). Running local versions of existing software does not solve that problem since it is not clear exactly which sequences were used for training. Also, in many cases a final network is trained using all available sequence, which means that any test is bound to be contaminated by homologous information. However, given the very large size of our test set, the rigorosity of our experimental setup, and the fact that rawMSA outperforms our own PSSM-based method, we believe that rawMSA compares favorably against the state of the art.

The convolutional layers of rawMSA depends on the order of the input sequences, since that will change the block of aligned positions from which features are extracted. To estimate the degree if this dependence we trained rawMSA with sequences sorted by sequence similarity using the BLOSUM62 similarity matrix (rawMSA1000 SS BLOSUM62 sort) and with randomly shuffled sequences (rawMSA1000 SS shuffle), see Table 1). Even though the performance decrease is small, neither of these two approaches worked as well as using the MSA as

Table 1. Results for the SS-RSA networks trained to predict secondary structure and solvent accessibility.

Predictor		Accuracy
rawMSA SS ensemble		83.4
One-hot100 SS		80.5
rawMSA100 SS		80.7
rawMSA1000 SS		81.8
rawMSA1000 SS BLOSUM62 sort		80.3
rawMSA1000 SS shuffle		79.8
PSSM SS		81.7
rawMSA RSA ensemble	(4-class)	57.7
One-hot100 RSA	(4-class)	53.7
rawMSA100 RSA	(4-class)	55.0
rawMSA1000 RSA	(4-class)	56.1
PSSM RSA	(4-class)	54.1
rawMSA RSA ensemble	(2-class)	81.2

The number in the predictor name refers to the number of sequences from the MSA that were used; *BLOSUM62 sort* and *shuffle* refers to different sorting of the sequences in the MSA (see text for details).

<https://doi.org/10.1371/journal.pone.0220182.t001>

outputted by HHblits, most likely because it is easier to learn from blocks of similar positions with smoother mutational transitions.

3.3 CMAP predictions

The final CMAP network is an ensemble of 10 networks trained on 10 to 1000 input sequences and varying numbers of layers (10 to 24 convolutional layers). The CMAP predictions for each target have been sorted by the contact probability measure output by the ensemble, then the top $L/5$ long-range contacts have been evaluated against the native contact map. The final accuracy has been calculated as the average of the accuracies for all targets. In order to make a fair comparison against the other predictors, we have downloaded all of the predictions made in CASP12 and evaluated them with the same system. In Table 2 we compare the top $L/5$ long-range accuracy of rawMSA CMAP against the top 5 CASP12 predictors.

rawMSA outperforms the top predictors in CASP12 under the same testing conditions. This is unexpected, since it is the only top predictor not to use any kind of explicit coevolution-based features, or any other inputs than the MSA. On the other hand, CASP12 was held in 2016 and the field has made rapid progress since then. For example, the group behind RaptorX-Contact has reported an improvement of roughly 12 percentage points on this same

Table 2. Comparison of rawMSA against the top 5 contact prediction methods in CASP12.

Predictor	Domain Count	L/5 LR Accuracy
rawMSA CMAP	37	43.8
RaptorX-Contact	37	43.0
iFold_1	36	42.3
Deepfold-Contact	37	38.6
MetaPSICOV	37	38.4
MULTICOM-CLUSTER	37	37.9

All predictions from CASP12 have been re-evaluated to ensure a fair comparison. The accuracy is calculated on the top $L/5$ long-range contacts.

<https://doi.org/10.1371/journal.pone.0220182.t002>

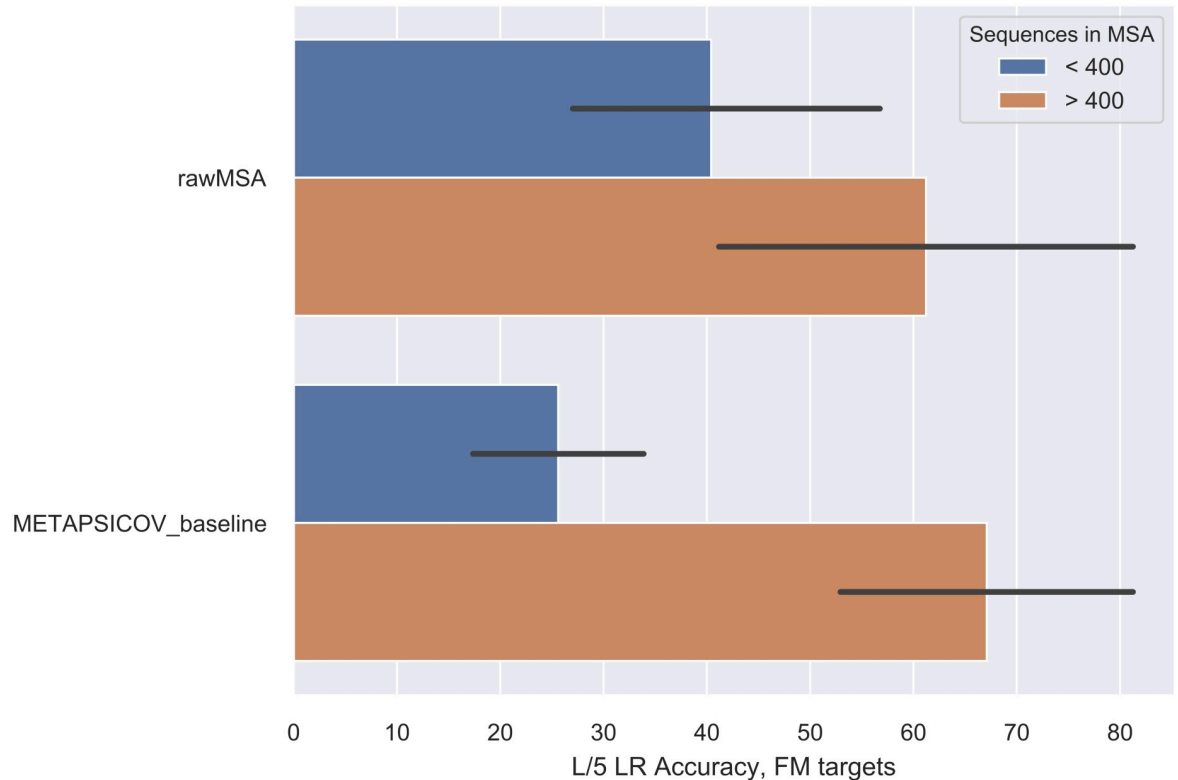


Fig 4. Impact of the number of sequences in the MSA on performance for rawMSA and METAPSICOV_baseline in the CASP13 FM targets.

<https://doi.org/10.1371/journal.pone.0220182.g004>

CASP12 test set only months after the experiment was closed with a new deeper version of their neural network [44].

Moreover, we expect rawMSA to be better than coevolution-based methods only whenever a relatively small number of sequences can be found in an MSA for a given target sequence, since only up to 1000 input MSA sequences could be used in training and testing because of limits in the amount of GPU RAM available (<25GB). Coevolution-based methods are more accurate as the number of sequences in the MSA increases and better metagenomic datasets [67] will produce larger MSAs for more target sequences. In the latest CASP13 experiment [68], where we participated with a prototype of rawMSA trained on a smaller ensemble of simpler models (up to 400 sequences per MSA) using a smaller sequence databases (Uniclust30), rawMSA was not among the best predictors in the contact prediction category. Nevertheless, the rawMSA prototype still outperformed a number of other coevolution-based methods, in particular the METAPSICOV_baseline method for MSAs with < 400 sequences, see Fig 4. For sequences with > 400 sequences in the MSA METAPSICOV_baseline method is still better. However, since we observe a clear correlation in testing accuracy and the number of sequences used as input, it is reasonable to expect that rawMSA will benefit from training on GPUs with larger memory allowing more sequences and deeper architectures to be used. In addition, rawMSA CMAP could also be improved by predicting distances, which seem to be direction in which the field is heading.

4 Conclusion

We have presented a new paradigm for the prediction of structural features of proteins called rawMSA, which involves using raw multiple sequence alignments (MSA) of proteins as input

instead of compressing them into protein sequence profiles, as is common practice today. Furthermore, rawMSA does not need any other manually designed or otherwise hand-picked extra feature as input, but instead exploits the capability that deep networks have of automatically extracting any relevant feature from the raw data.

To convert MSAs, which could be described as categorical data, to a more machine-friendly format, rawMSA adopts embeddings, a technique from the field of Natural Language Processing to adaptively map discrete inputs from a dictionary of symbols into vectors in a continuous space.

To showcase our novel representation of the MSA, we developed a few different flavors of rawMSA to predict secondary structure, relative solvent accessibility and inter-residue contact maps. All these networks use the same and only kind of input, i.e. the MSA. After rigorous testing, we show how rawMSA SS-RSA sets a new state of the art for these kinds of predictions, and rawMSA CMAP performs on par with methods using more pre-calculated features in the inter-residue contact map prediction category in CASP12 and CASP13. Clearly demonstrating that rawMSA represents a promising development that can pave the way for improved methods using rawMSA instead of sequence profiles to represent evolutionary information in the coming years.

Acknowledgments

The Swedish e-Science Research Center. Computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the National Supercomputer Centre (NSC) in Linköping and at the High Performance Computing Center North (HPC2N) in Umeå and at Hops (www.hops.io). We also thank Isak Johansson-Åkhe for helpful discussions.

Author Contributions

Conceptualization: Claudio Mirabello.

Funding acquisition: Claudio Mirabello, Björn Wallner.

Investigation: Claudio Mirabello.

Methodology: Claudio Mirabello.

Resources: Björn Wallner.

Software: Claudio Mirabello.

Supervision: Björn Wallner.

Visualization: Claudio Mirabello.

Writing – original draft: Claudio Mirabello.

Writing – review & editing: Claudio Mirabello, Björn Wallner.

References

1. Dill KA, Ozkan SB, Weikl TR, Chodera JD, Voelz VA. The protein folding problem: when will it be solved? *Current Opinion in Structural Biology*. 2007; 17(3):342–346. <https://doi.org/10.1016/j.sbi.2007.06.001> PMID: 17572080
2. Shell MS, Ozkan SB, Voelz V, Wu GA, Dill KA. Blind test of physics-based prediction of protein structures. *Biophysical Journal*. 2009; 96(3):917–924. <https://doi.org/10.1016/j.bpj.2008.11.009> PMID: 19186130

3. Sippl MJ. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. *Journal of Molecular Biology*. 1990; 213(4):859–883. [https://doi.org/10.1016/s0022-2836\(05\)80269-4](https://doi.org/10.1016/s0022-2836(05)80269-4) PMID: 2359125
4. Jones DT, Taylor WR, Thornton JM. A new approach to protein fold recognition. *Nature*. 1992; 358(6381):86–89. <https://doi.org/10.1038/358086a0> PMID: 1614539
5. Sippl MJ. Knowledge-based potentials for proteins. *Current Opinion in Structural Biology*. 1995; 5(2):229–235. [https://doi.org/10.1016/0959-440X\(95\)80081-6](https://doi.org/10.1016/0959-440X(95)80081-6) PMID: 7648326
6. Lazaridis T, Karplus M. Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology*. 2000; 10(2):139–145. [https://doi.org/10.1016/S0959-440X\(00\)00063-4](https://doi.org/10.1016/S0959-440X(00)00063-4) PMID: 10753811
7. Simons KT, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology*. 1997; 268(1):209–225. <https://doi.org/10.1006/jmbi.1997.0959> PMID: 9149153
8. Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, et al. Atomic-level characterization of the structural dynamics of proteins. *Science (New York, NY)*. 2010; 330(6002):341–346. <https://doi.org/10.1126/science.1187409>
9. Wallner B, Elofsson A. Can correct protein models be identified? *Protein Science*. 2003; 12(5):1073–1086. <https://doi.org/10.1110/ps.0236803> PMID: 12717029
10. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices1. *Journal of Molecular Biology*. 1999; 292(2):195–202. <https://doi.org/10.1006/jmbi.1999.3091> PMID: 10493868
11. Cuff JA, Clamp ME, Siddiqui AS, Finlay M, Barton GJ. JPred: a consensus secondary structure prediction server. *Bioinformatics*. 1998; 14(10):892–893. <https://doi.org/10.1093/bioinformatics/14.10.892> PMID: 9927721
12. Pollastri G, Przybylski D, Rost B, Baldi P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*. 2002; 47(2):228–235. <https://doi.org/10.1002/prot.10082>
13. Pollastri G, McIysaght A. Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*. 2004; 21(8):1719–1720. <https://doi.org/10.1093/bioinformatics/bti203> PMID: 15585524
14. Drozdetskiy A, Cole C, Procter J, Barton GJ. JPred4: a protein secondary structure prediction server. *Nucleic acids research*. 2015; 43(W1):W389–94. <https://doi.org/10.1093/nar/gkv332> PMID: 25883141
15. Wang S, Li W, Liu S, Xu J. RaptorX-Property: a web server for protein structure property prediction. *Nucleic acids research*. 2016; 44(W1):W430–5. <https://doi.org/10.1093/nar/gkw306> PMID: 27112573
16. Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. *Proteins: Structure, Function, and Bioinformatics*. 1994; 20(3):216–226. <https://doi.org/10.1002/prot.340200303>
17. Pollastri G, Baldi P, Fariselli P, Casadio R. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins: Structure, Function, and Bioinformatics*. 2002; 47(2):142–153. <https://doi.org/10.1002/prot.10069>
18. Adamczak R, Porollo A, Meller J. Accurate prediction of solvent accessibility using neural networks–based regression. *Proteins: Structure, Function, and Bioinformatics*. 2004; 56(4):753–767. <https://doi.org/10.1002/prot.20176>
19. Gao Y, Wang S, Deng M, Xu J. RaptorX-Angle: real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning. *BMC Bioinformatics*. 2018; 19(Suppl 4):100. <https://doi.org/10.1186/s12859-018-2065-x> PMID: 29745828
20. Linding R, Jensen LJ, Diella F, Bork P, Gibson TJ, Russell RB. Protein disorder prediction: implications for structural proteomics. *Structure*. 2003; 11(11):1453–1459. <https://doi.org/10.1016/j.str.2003.10.002> PMID: 14604535
21. Ward JJ, McGuffin LJ, Bryson K, Buxton BF, Jones DT. The DISOPRED server for the prediction of protein disorder. *Bioinformatics*. 2004; 20(13):2138–2139. <https://doi.org/10.1093/bioinformatics/bth195> PMID: 15044227
22. Jones DT, Cozzetto D. DISOPRED3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics*. 2015; 31(6):857–863. <https://doi.org/10.1093/bioinformatics/btu744> PMID: 25391399
23. Basu Sankar, Söderquist Fredrik, Wallner Björn. Proteus: a random forest classifier to predict disorder-to-order transitioning binding regions in intrinsically disordered proteins. *Journal of computer-aided molecular design*. 2017; 31(5):453–466. <https://doi.org/10.1007/s10822-017-0020-y> PMID: 28365882
24. Fariselli P, Casadio R. A neural network based predictor of residue contacts in proteins. *Protein engineering*. 1999; 12(1):15–21. <https://doi.org/10.1093/protein/12.1.15> PMID: 10065706
25. Punta M, Rost B. PROFcon: novel prediction of long-range contacts. *Bioinformatics*. 2005; 21(13):2960–2968. <https://doi.org/10.1093/bioinformatics/bti454> PMID: 15890748

26. 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. <https://doi.org/10.1186/1471-2105-15-6> PMID: 24410833
27. Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLoS Computational Biology*. 2017; 13(1):e1005324. <https://doi.org/10.1371/journal.pcbi.1005324> PMID: 28056090
28. Ray A, Lindahl E, Wallner B. Improved model quality assessment using ProQ2. *BMC Bioinformatics*. 2012; 13(1):224. <https://doi.org/10.1186/1471-2105-13-224> PMID: 22963006
29. Uziela Karolis, Menendez Hurtado David, Shu Nanjiang, Wallner Björn, Elofsson Arne. ProQ3D: improved model quality assessments using deep learning. *Bioinformatics*. 2017; 33(10):1578–1580. <https://doi.org/10.1093/bioinformatics/btw819> PMID: 28052925
30. Cao R, Adhikari B, Bhattacharya D, Sun M, Hou J, Cheng J. QAcon: single model quality assessment using protein structural and contact information with machine learning techniques. *Bioinformatics*. 2017; 33(4):586–588. <https://doi.org/10.1093/bioinformatics/btw694> PMID: 28035027
31. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*. 1993; 232(2):584–599. <https://doi.org/10.1006/jmbi.1993.1413> PMID: 8345525
32. Cuff JA, Barton GJ. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*. 2000; 40(3):502–511. [https://doi.org/10.1002/1097-0134\(20000815\)40:3%3C502::AID-PROT170%3E3.0.CO;2-Q](https://doi.org/10.1002/1097-0134(20000815)40:3%3C502::AID-PROT170%3E3.0.CO;2-Q)
33. Rohl CA, Strauss CE, Misura KM, Baker D. Protein structure prediction using Rosetta. In: *Methods in Enzymology*. vol. 383. Elsevier; 2004. p. 66–93.
34. McGuffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. *Bioinformatics*. 2000; 16(4):404–405. <https://doi.org/10.1093/bioinformatics/16.4.404> PMID: 10869041
35. Roy A, Kucukural A, Zhang Y. I-TASSER: a unified platform for automated protein structure and function prediction. *Nature Protocols*. 2010; 5(4):725. <https://doi.org/10.1038/nprot.2010.5> PMID: 20360767
36. Baú D, Martin AJ, Mooney C, Vullo A, Walsh I, Pollastri G. Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins. *BMC Bioinformatics*. 2006; 7(1):402. <https://doi.org/10.1186/1471-2105-7-402> PMID: 16953874
37. Tegge AN, Wang Z, Eickholt J, Cheng J. NNcon: improved protein contact map prediction using 2D-recursive neural networks. *Nucleic acids research*. 2009; 37(suppl_2):W515–W518. <https://doi.org/10.1093/nar/gkp305> PMID: 19420062
38. Pollastri G, Baldi P. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*. 2002; 18(suppl_1):S62–S70. https://doi.org/10.1093/bioinformatics/18.suppl_1.s62 PMID: 12169532
39. Morcos Faruck, Pagnani Andrea, Lunt Bryan, Bertolino Arianna, Marks Debora S, Sander Chris, et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*. 2011; 108(49):E1293–301. <https://doi.org/10.1073/pnas.1111471108>
40. Jones DT, Buchan DW, Cozzetto D, Pontil M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*. 2011; 28(2):184–190. <https://doi.org/10.1093/bioinformatics/btr638> PMID: 22101153
41. Ekeberg M, Hartonen T, Aurell E. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*. 2014; 276:341–356. <https://doi.org/10.1016/j.jcp.2014.07.024>
42. Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Computational Biology*. 2017; 13(1):e1005324. <https://doi.org/10.1371/journal.pcbi.1005324> PMID: 28056090
43. Adhikari B, Hou J, Cheng J. DNCON2: Improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*. 2017.
44. Wang S, Sun S, Xu J. Analysis of deep learning methods for blind protein contact prediction in CASP12. *Proteins: Structure, Function, and Bioinformatics*. 2018; 86:67–77. <https://doi.org/10.1002/prot.25377>
45. Buchan DWA, Jones DT. Improved protein contact predictions with the MetaPSICOV2 server in CASP12. *Proteins*. 2018; 86 Suppl 1:78–83. <https://doi.org/10.1002/prot.25379> PMID: 28901583
46. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015; 521(7553):436. <https://doi.org/10.1038/nature14539> PMID: 26017442
47. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:13013781*. 2013.

48. Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*. 2015; 10(11):e0141287. <https://doi.org/10.1371/journal.pone.0141287> PMID: 26555596
49. Chou PY, Fasman GD. Conformational parameters for amino acids in helical, β -sheet, and random coil regions calculated from proteins. *Biochemistry*. 1974; 13(2):211–222. <https://doi.org/10.1021/bi00699a001> PMID: 4358939
50. Chollet F, et al. Keras; 2015. <https://github.com/fchollet/keras>.
51. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Available from: <https://www.tensorflow.org/>.
52. Wang G, Dunbrack RL Jr. PISCES: a protein sequence culling server. *Bioinformatics*. 2003; 19(12):1589–1591. <https://doi.org/10.1093/bioinformatics/btg224> PMID: 12912846
53. Fang C, Shang Y, Xu D. MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*. 2018; 86(5):592–598. <https://doi.org/10.1002/prot.25487>
54. Torrisi M, Kaleel M, Pollastri G. Porter 5: fast, state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes. *bioRxiv*. 2018; p. 289033.
55. Yang Y, Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, et al. Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. In: *Prediction of Protein Secondary Structure*. Springer; 2017. p. 55–63.
56. Wang Y, Mao H, Yi Z. Protein secondary structure prediction by using deep learning method. *Knowledge-Based Systems*. 2017; 118:115–123. <https://doi.org/10.1016/j.knsys.2016.11.015>
57. Söding J, Remmert M. Protein sequence comparison and fold recognition: progress and good-practice benchmarking. *Current Opinion in Structural Biology*. 2011; 21(3):404–411. <https://doi.org/10.1016/j.sbi.2011.03.005> PMID: 21458982
58. Cheng H, Schaeffer RD, Liao Y, Kinch LN, Pei J, Shi S, et al. ECOD: an evolutionary classification of protein domains. *PLoS Computational Biology*. 2014; 10(12):e1003926. <https://doi.org/10.1371/journal.pcbi.1003926> PMID: 25474468
59. Fox NK, Brenner SE, Chandonia JM. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic acids research*. 2013; 42(D1):D304–D309. <https://doi.org/10.1093/nar/gkt1240> PMID: 24304899
60. Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*. 2012; 9(2):173–175. <https://doi.org/10.1038/nmeth.1818>
61. Johnson L Steven, Eddy Sean R, Portugaly Elon. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*. 2010; 11(1):431. <https://doi.org/10.1186/1471-2105-11-431> PMID: 20718988
62. Schaarschmidt J, Monastyrskyy B, Kryshtafovych A, Bonvin AM. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*. 2018; 86:51–66. <https://doi.org/10.1002/prot.25407>
63. Joosten RP, Te Beek TA, Krieger E, Hekkelman ML, Hooft RW, Schneider R, et al. A series of PDB related databases for everyday needs. *Nucleic acids research*. 2010; 39(suppl_1):D411–D419. <https://doi.org/10.1093/nar/gkq1105> PMID: 21071423
64. Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO. Maximum allowed solvent accessibilities of residues in proteins. *PLoS one*. 2013; 8(11):e80635. <https://doi.org/10.1371/journal.pone.0080635> PMID: 24278298
65. Naftaly U, Intrator N, Horn D. Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*. 1997; 8(3):283–296. https://doi.org/10.1088/0954-898X_8_3_004
66. Yang Y, Gao J, Wang J, Heffernan R, Hanson J, Paliwal K, et al. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in bioinformatics*. 2016; 19(3):482–494.
67. Steinegger M, Söding J. Clustering huge protein sequence sets in linear time. *Nature communications*. 2018; 9(1):2542. <https://doi.org/10.1038/s41467-018-04964-5> PMID: 29959318
68. CASP. CASP13 Webpage; 2018. <http://predictioncenter.org/casp13>.