Posted online 4 Sep 2017 *bioRxiv, 2017,* ***1–6***

Optimizing scoring function of dynamic programming of pairwise profile alignment using derivative free neural network

Kazunori D Yamada [1,2*]

[1]Graduate School of Information Sciences, Tohoku University, Sendai, Japan [2]Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

ABSTRACT

A profile comparison method with position-specific scoring matrix (PSSM) is one of the most accurate alignment methods. Currently, cosine similarity and correlation coefficient are used as scoring functions of dynamic programming to calculate similarity between PSSMs. However, it is unclear that these functions are optimal for profile alignment methods. At least, by definition, these functions cannot capture non-linear relationships between profiles. Therefore, in this study, we attempted to discover a novel scoring function, which was more suitable for the profile comparison method than the existing ones. Firstly we implemented a new derivative free neural network by combining the conventional neural network with evolutionary strategy optimization method. Next, using the framework, the scoring function was optimized for aligning remote sequence pairs. Nepal, the pairwise profile aligner with the novel scoring function significantly improved both alignment sensitivity and precision, compared to aligners with the existing functions. Nepal improved alignment quality because of adaptation to remote sequence alignment and increasing the expressive power of similarity score. The novel scoring function can be realized using a simple matrix operation and easily incorporated into other aligners. With our scoring function, the performance of homology detection and/or multiple sequence alignment for remote homologous sequences would be further improved.

INTRODUCTION

The profile comparison alignment method with a position-specific scoring matrix (PSSM) [1] is one of the most accurate alignment methods. The PSSM is a two dimensional vector (matrix) for sequence length. Each element in the vector consists of a 20 dimensional numerical vector, in which each value represents the likelihood of the existence of each amino acid position in a biological sequence. Here, we designed the vector inside PSSM as a position-specific scoring vector (PSSV). In a profile alignment, cosine similarity or correlation coefficient is generally calculated against the PSSVs to calculate similarity or dissimilarity between the two sites in the sequences of interest on dynamic programming (DP) [2, 3]. Profile alignment methods using these functions have been successful for a long time [4], although cosine similarity or correlation coefficient cannot capture the non-linear relationship between two vectors and the similarity between two sites is not always expressed by linear relationships.

The performance of profile sequence alignment has been improved by various studies in the past decades. For example, HHalign improved alignment quality using profiles constructed with the hidden Markov model, which provided more information than PSSM [5], MUSTER incorporated protein structural information in a profile [3], and MRFalign utilized the Markov random fields to improve alignment quality [6]. Although various methods have been devised from different perspectives, studies to develop the scoring function itself with sophisticated technologies are lacking.

Neural networks are computing system, which mimic biological nervous system of animal brains. Theoretically, it can approximate any function regardless of linearity of the functions [7]. Neural networks are attracting attention from various areas of research, including bioinformatics, due to the availability of improved computational methods and the explosive increase in available data. In recent years, these algorithms have been vigorously applied to bioinformatics. For example, several studies applied a deep neural network model to predict protein-protein interaction [8, 9], protein structure [10, 11] and various other biological conditions such as residue contact map, backbone angles, and solvent accessibility [12, 13]. These algorithms basically used the backpropagation method, which requires derivation of a cost function for searching optimal parameters, and few studies implemented derivative free neural network.

In this study, we utilized the neural network to optimize a scoring function. In the process, we first combined two PSSVs (for which we wanted to calculate similarity) derived from two sites and set it as an input vector. A target vector was required to implement supervised learning. However, in this case, we did not have the target vector because the ideal function and an ideal similarity score for each site were unknown, and thus, the scoring function could not be directly optimized. Instead, we calculated the entire DP table for the input sequences and the difference between the resultant alignment and the correct alignment was used for calculating cost. In this case, we could not use the backpropagation method for optimal weight search because we lacked the derivation of the cost function required for this search. Namely, we could not incorporate our

*To whom correspondence should be addressed. Tel: +81 22 795 7161; Email: kyamada@ecei.tohoku.ac.jp

idea in the conventional neural network framework. Therefore, we newly utilized the covariance matrix adaptation evolution strategy (CMA-ES) [14], which is an adaptive optimization method modifying the basic evolutionary strategy [15], as the search method for neural network to realize derivative free neural network calculation. Using this framework, we attempted to produce higher performance scoring function for remote sequence alignment in this study.

METHODS

Dataset

We downloaded the non-redundant subset of SCOP40 (1.75 release) [16], in which sequence identity between any sequence pair is less than 40%, from ASTRAL [17]. We selected the remote sequence subset since we wanted to improve the remote sequence alignment quality. The SCOP is a protein domain dataset where sequences are classified in hierarchical manner by class, fold, superfamily, and family. All notations of the superfamily in the dataset were sorted by alphabetical order and all superfamilies, the ordered numbers of which were multiples of three, were classified into a learning dataset, whereas the others were classified into a test dataset. We obtained 3,726 and 6,843 sequences in the learning and test datasets, respectively. Next, we randomly extracted a maximum of 10 pairs of sequences from each superfamily to negate a bias induced by different volumes of each superfamily and used these sequence pairs for subsequence construction of PSSM. We confirmed that sequences in each pair were from the same family to obtain decent reference alignment. Finally, we obtained 1,721 and 3,195 sequence pairs in the learning and test datasets.

Construction of profiles and reference alignments

We constructed PSSMs for all sequences in the learning and test datasets using DELTA-BLAST version 2.2.30+ with the Conserved Domain Database for DELTA-BLAST version 3.12 [18]. Reference alignments were constructed through structural alignment of protein steric structures, which corresponded to sequences of interest using TM-align [19]. All structure data were also downloaded from ASTRAL [17].

Learning network

Figure 1 shows the learning network computed in this study. We calculated similarity scores between two PSSVs using the neural network. At first, the summation of matrix products between xa (the PSSV A) and $\boldsymbol{W}_{1a}$, $\boldsymbol{x}_b$ (the other PSSV B) and $\boldsymbol{W}_{1b}$, and 1 (bias) and $\boldsymbol{b}_1$ in the neural network were calculated. The resultant vector was transformed by an activating function, $\phi()$. Finally, the summation of the dot products between the transformed vector and $\boldsymbol{w}_2$, and 1 and b_2 was calculated. The resultant value was used as the similarity score for the two sites. Namely, the forward calculation was computed by the following equation. Here, y is the similarity score.

$$y = \boldsymbol{w}_2 \phi(\boldsymbol{x}_a \boldsymbol{W}_{1a} + \boldsymbol{x}_b \boldsymbol{W}_{1b} + \boldsymbol{b}_1) + b_2$$

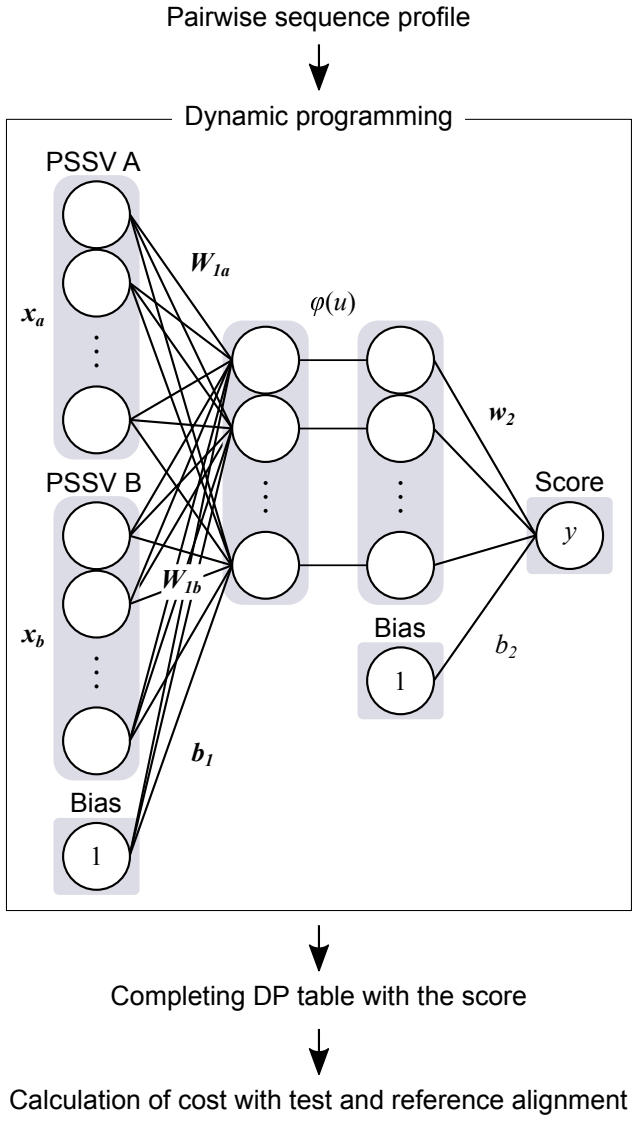

Figure 1. Schematic diagram of learning network developed in this study. The upper case letters in italics and bold face, the lower case letters in italics and bold face, and the lower case letters in italics represent matrix, vector, and scalar values, respectively. The activating function is represented by $\phi()$.

The complete DP table was calculated using the similarity score and a final pairwise alignment was produced. The pairwise alignment and its corresponding reference alignment were compared to each other and an alignment sensitivity score, described below, was calculated. The subtraction of the alignment sensitivity score from 1 was used as cost for searching optimum weight by the neural network with CMA-ES.

We set the weights $\boldsymbol{W}_{1a}$ and $\boldsymbol{W}_{1b}$ equal to each other (shared weight) so that the network outputs same value even though the input order of the two PSSVs were opposite. The number of units of the middle layer was set to 144. The rectified linear unit was utilized as the activation function. We set σ, λ, and μ as 0.032, 70, and 35, respectively, as parameters for CMA-ES. Here, σ is almost equivalent to step

size of the gradient descent method, and λ and μ indicate the number of descendant and survival individuals in evolutional process. In actual learning, we read training datasets in batch manner. The learning loop was stopped using the early stopping criteria by checking the dissociation between the training and validating curves. The initial weight was derived from parameters that mimicked the correlation coefficient. To generate the initial weight, we randomly generated 200,000 PSSM pairs and learned them using multilayer perceptron with hyperparameters (the dimension of weight and activating function) identical to the above hyperparameters. In addition to the weights, we simultaneously optimized the open and extension gap penalties. The initial values of open and extension gap penalties were set as -1.5 and -0.1.

Alignment algorithm

In this study, we implemented the semi-global alignment method, namely global alignment with free end-gaps method [20, 21].

Metrics of alignment quality

The alignment quality was evaluated using alignment sensitivity and precision [22]. The alignment sensitivity was calculated by dividing the number of correctly aligned sites by the number of non-gapped sites in a reference alignment. In contrast, alignment precision was calculated by dividing the number of correctly aligned sites by the number of non-gapped sites in a test alignment.

Calculation of residue interior propensity

The relative accessible surface area (rASA) for residues of all proteins in the learning and test dataset was calculated by areaimol in CCP4 package version 6.5.0 [23]. The residues of which rASA is less than 0.25 were counted as an interior residue and the other residues were counted as surface residue, according to a previous study [24]. We divided the ratio of the interior residues by the background probability of residues to calculate the residue interior propensity. The residue interior propensity is the likelihood of a residue existing inside a protein. Namely, propensity greater than 1 signifies that the probability of the residue to be inside the protein is high.

RESULTS AND DISCUSSION

Gap optimization of existing functions

At first, we conducted gap penalty optimization of the existing scoring functions such as cosine similarity and correlation coefficient on the learning dataset. We computed both alignment sensitivity and precision for aligners using these functions, changing open and extension gap penalties by 0.1 increments from -2.0 to -0.6 and from -0.4 to -0.1, respectively. The best alignment sensitivity was selected as the optimum combination among the combinations of open and extension gap penalties. As shown in Table 1, the best gap penalty combination for cosine similarity and correlation coefficient was (-1.0, -0.1) and (-1.5, -0.1).

Optimization of scoring function of the neural network

Next, we conducted optimization of scoring function on the neural network with CMA-ES. During learning, we randomly divided the learning dataset into two subsets, namely, the training and validation datasets, which included 1,536 and 160 pairwise PSSV sets and its corresponding reference alignments as targets, respectively. Since calculation of CMA-ES in our parameter settings requires more than 100,000 times DP (the size of training dataset $\times$ λ) per epoch, the consumption of computer resources was large and calculation time was long even when 24 threads were used with the C++ program; therefore, we set the maximum limit for epoch to a small number such as 150. We selected the best scores from the validation scores of the last fifth part of an entire epoch (which was derived from 145th epoch) and obtained final weight and bias matrices, namely, the substance of a novel scoring function and optimal gap penalty combination, respectively. As a result, optimal combination of open and extension gap penalty for the final weight and bias matrix were approximately -1.7 and -0.2.

Finally, we implemented the pairwise profile aligner with the weight and bias matrices as novel scoring function and named it as neural network enhanced profile alignment library (Nepal). Our aligner and scoring function (weight and bias matrices) can be downloaded from https://github.com/yamada-kd/nepal.

Benchmark of Nepal and other aligners with existing function on the test dataset

Next, we conducted benchmark test of Nepal and other aligners with existing functions on the test dataset. In addition to profile comparison methods, we examined the performance of sequence comparison aligners with difference substitution matrices such as BLOSUM62 [25] and MIQS [26] for reference. We used -10 and -2 as open and extension gap penalties, respectively, based on a previous study [26]. When calculating alignment qualities, the test dataset was further categorized into remote and medium subset depending on pairwise sequence identity of the reference alignments. The remote and medium subset includes sequence pairs, of which each sequence identity was not lower than 0% and less than 20%, and not lower than 20% and less than 40%, respectively. Generally, a pairwise alignment between sequences of lower identity such as those in the twilight zone is more difficult [27].

Table 2 shows alignment quality scores for each method. Results show that among the existing methods, including sequence comparison methods, the method with the best performance from all perspectives was the profile

Table 1. Gap optimization of the existing scoring function

	Open	Extension	Sensitivity	Precision
Cosine	-1.0	-0.1	0.6837	0.6550
CC	-1.5	-0.1	0.6882	0.6613

Open and Extension indicate optimized open and extension gap penalties, respectively, and Cosine and CC represent aligners with a cosine similarity and correlation coefficient as scoring functions, respectively.

comparison method with correlation coefficient scoring function. In contrast, Nepal improved both alignment sensitivity and precision compared to this method. Actually, these improvements were statistically significant according to Wilcoxon signed rank test with Bonferroni correction even when significance level (α) is set to 0.01. Comparison between sequence-based methods with different substitution matrices such as MIQS and BLOSUM62 showed that the gain of improvement of MIQS compared to BLOSUM62 was more significant for the remote subset than the medium subset. This was expected since MIQS was originally developed to improve remote homology alignment. This trend was observed regarding the relationship between Nepal and correlation coefficient implemented aligner, where Nepal improved both alignment sensitivity and precision by about 4% and 1% in remote and medium subsets, respectively. This indicated that the novel scoring function was optimized for remote sequence alignment. This is expected because sequence alignment between sequences with closer identities was easier than those with remote identities. Therefore, during optimization, the novel scoring function would be optimized to be naturally advantageous for remote sequence alignments. Since the problem regarding remote relationship holds true for sequence similarity search [26, 28], the novel scoring function of our method could be useful for improving the performance of remote similarity search methods.

Importance of attributes using the connection weight method

Finally, we calculated the importance of 20 attributes using the connection weight method [29]. As shown in Figure 2A, the connection weights against each attribute, namely each amino acid, were distributed to various values. This indicated that our developed scoring function discerned the importance of the attributes depending on the variety of amino acids.

According to the results, the connection weight of hydrophobic residues such as Leu, Ile, and Val were of higher value. These residues are located mostly inside the hydrophobic cores of proteins. In addition, as shown in Figure 2B, the other residues which also tend to locate inside proteins, such as Ala, Cys, and Tyr, were of higher importance. In contrast, residues which tend to locate on protein surface, such as Asp, Pro, Lys, and Asn, were of lower importance. The Spearman's rank correlation coefficient between the connection weight and interior propensity was approximately 0.6 and the value was statistically significant (p-value < 0.05). While residues which are exposed on the protein surface are subject to higher mutation pressures, interior residues are less susceptible to mutation [30]. This is because the protein structure is disrupted if mutations in the interior residues collapse the hydrophobic core [31]. The scoring function constructed in this study was optimized for alignment of remote homologous sequences. According to the previous study based on substitution matrices [32], hydrophobicity of residues was the dominant property of remote sequence substitution rather than simple mutability. This fact partially represents that for remote sequence alignment, residues occupying interior locations in a protein higher order structure

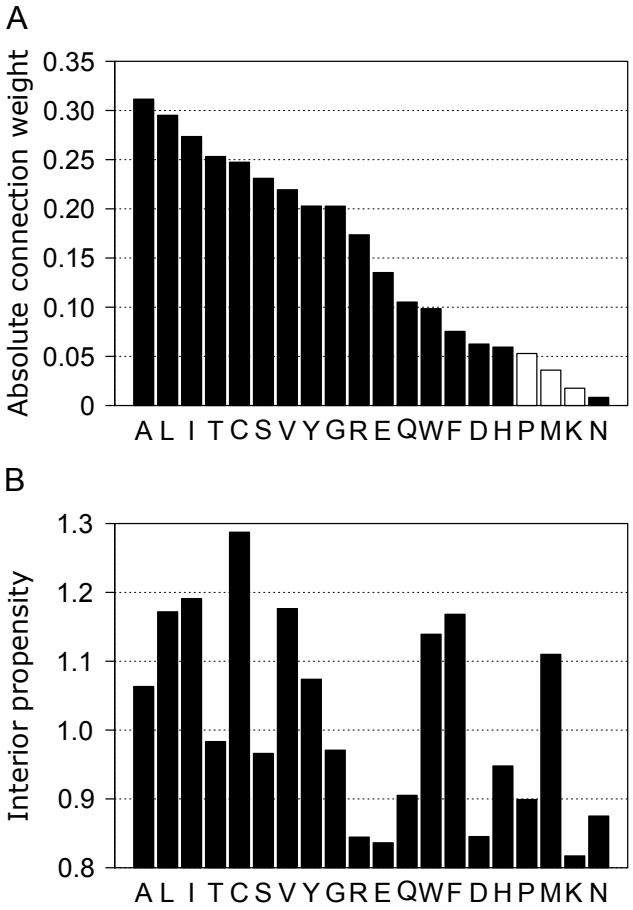

Figure 2. (A) Absolute connection weight for each attribute, which corresponds to the profile value of each amino acid. Filled and open bar represents positive and negative sign of original connection weight, respectively. (B) The residue interior propensity against whole data in the study.

Table 2. Comparison of Nepal with other alignment methods

	Remote [0,20) (1,405 files)	Medium [20,40) (1,790 files)	All [0,40) (3,195 files)
Sensitivity			
Nepal	**0.5317**	**0.8343**	**0.7012**
Cosine	0.5045**	0.8246**	0.6838**
CC	0.5135**	0.8269**	0.6891**
MIQS	0.2775**	0.7316**	0.5319**
BL62	0.2333**	0.6955**	0.4923**
Precision			
Nepal	**0.5031**	**0.8102**	**0.6751**
Cosine	0.4753**	0.7999**	0.6571**
CC	0.4858**	0.8032**	0.6636**
MIQS	0.2654**	0.7134**	0.5164**
BL62	0.2317**	0.6902**	0.4885**

The interval on the second line in the header represents sequence identity (%) of each division. Methods such as Cosine, CC, MIQS, and BL62 indicate profile comparison methods with cosine similarity and correlation coefficient and sequence comparison methods with MIQS and BLOSUM62. The double asterisks on the score (**) indicate p-value < 0.01 on Wilcoxon signed rank test with Bonferroni correction when the method is compared to Nepal.

with less susceptibility to mutation pressure are considered more meaningful. Since our scoring function was also optimized for remote sequence alignment, the above property would be observed and this fact paradoxically suggests that our scoring function was optimized for remote sequence alignment. Collectively, this property is one of the reasons for the superiority of our method to the existing ones.

In addition, although the connection weight consisted of various values, it would at least contribute to increasing the expressive power of the novel scoring function. For example, we wanted to calculate the similarity score between PSSV A (a) and B (b) as shown in Figure 3. The original scores are 0.488207 and 0.387911 when calculated using the correlation coefficient and Nepal score, respectively, (middle panel Figure 3). The scores calculated by correlation coefficient did not change when the 1st and 18th sites or the 4th and 19th sites were swapped. This was unexpected since the converted PSSV obtained after swapping was not identical to the original one. This could be one of the drawbacks of using unweighted linear function such as cosine similarity and correlation coefficient. In contrast, Nepal scores changed after the swapping, which varied with the change in PSSV. Actually, there were about 290,000 overlaps when we calculated similarity score to six places of decimal against randomly generated one million PSSVs using correlation coefficient, whereas there were approximately 180,000 overlaps when Nepal was used. These overlaps would negatively affect DP computation because higher overlap scores would cause difficulty in deciding the correct path, especially during the computation of maximum three values derived from up, diagonal, and left side of the DP cell.

Collectively, the different weights based on amino acid variety presented by the connection weight method is one of the reasons why Nepal score improved the alignment quality compared to the existing scoring functions.

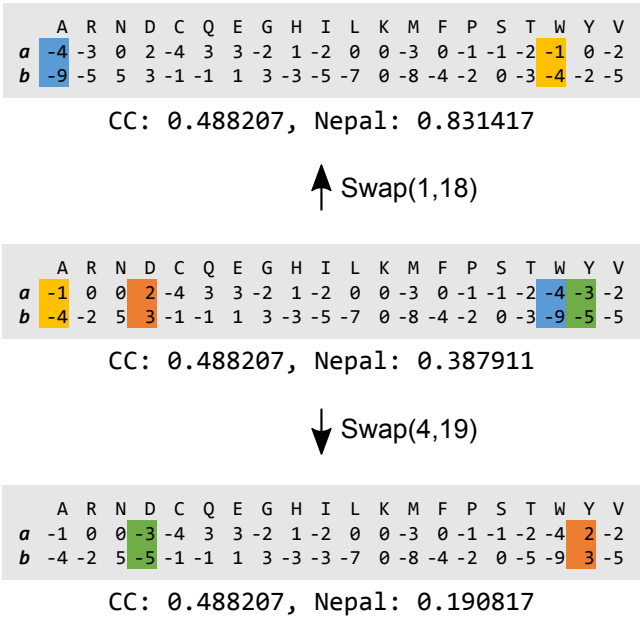

Figure 3. Transition of similarity scores depending on site swapping. In each panel, a and b stand for PSSV A and B respectively. The middle panel represents an original PSSV and similarity scores calculated by correlation coefficient (CC) and Nepal. The top and bottom panel stands for the resultant PSSVs and the similarity scores.

CONCLUSION

In this study, we developed a new derivative free neural network with CMA-ES. Using this framework, we developed a novel scoring function for profile comparison and Nepal, a pairwise profile aligner with the scoring function. Large computational resources were required by our learning procedure with the derivative free neural network; thus, we could not examine whether the learning was converged enough because of our limited computational environment. Nevertheless, Nepal significantly improved alignment quality of profile alignment, especially for alignment of remote relationships, compared to the existing scoring functions. Nepal improved alignment quality because of adaptation to remote sequence alignment and increasing the expressive power of similarity score. The novel scoring function can be realized using a simple matrix operation and the parameters are provided on https://github.com/yamada-kd/nepal. In future, the performance of distant homology detection method or that of multiple sequence alignment method for remote homologous sequences may be further improved with our scoring function.

ADDITIONAL INFORMATION

Acknowledgements

We are grateful to Dr Kentaro Tomii, Dr. Satoshi Omori and Mr. Tsukasa Nakamura for constructive discussion. Computations were partially performed on the NIG supercomputer at the ROIS National Institute of Genetics.

Funding

This work was supported in part by the Top Global University Project from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT)

Availability of data and material

The source code of Nepal and the learned parameters are available at GitHub (https://github.com/yamada-kd/nepal).

Abbreviations

CMA-ES: covariance matrix adaptation evolution strategy; DP: dynamic programming; PSSM: position-specific scoring matrix; PSSV: position-specific scoring vector

Competing interests

The authors declare that they have no competing interests.

Author Contribution

KDY did everything.

REFERENCES

1. S F Altschul, T L Madden, A A Schffer, J Zhang, Z Zhang, W Miller, and D J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25:3389–3402, September 1997.
2. Kentaro Tomii and Yutaka Akiyama. Forte: a profile-profile comparison tool for protein fold recognition. *Bioinformatics (Oxford, England)*, 20:594–595, March 2004.
3. Sitao Wu and Yang Zhang. Muster: Improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins*, 72:547–556, August 2008.
4. Kentaro Tomii, Takatsugu Hirokawa, and Chie Motono. Protein structure prediction using a variety of profile libraries and 3d verification. *Proteins*, 61 Suppl 7:114–121, 2005.
5. Johannes Sding. Protein homology detection by hmm-hmm comparison. *Bioinformatics (Oxford, England)*, 21:951–960, April 2005.
6. Jianzhu Ma, Sheng Wang, Zhiyong Wang, and Jinbo Xu. Mrfalign: protein homology detection through alignment of markov random fields. *PLoS computational biology*, 10:e1003500, March 2014.
7. G Gybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
8. Xiuquan Du, Shiwei Sun, Changlin Hu, Yu Yao, Yuanting Yan, and Yanping Zhang. Deepppi: Boosting prediction of protein-protein interactions with deep neural networks. *Journal of chemical information and modeling*, 57:1499–1510, June 2017.
9. Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics*, 18(1):277, 2017.
10. Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6:18962, January 2016.
11. Matt Spencer, Jesse Eickholt, and Jianlin Cheng. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 12:103–112, 2015.
12. Pietro Di Lena, Ken Nagata, and Pierre Baldi. Deep architectures for protein contact map prediction. *Bioinformatics (Oxford, England)*, 28:2449–2457, October 2012.
13. Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. *Bioinformatics (Oxford, England)*, April 2017.
14. Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317. IEEE, 1996.
15. Hans-Georg Beyer. Toward a theory of evolution strategies: Some asymptotical results from the (1,+ λ)-theory. *Evolutionary computation*, 1(2):165–188, 1993.
16. Antonina Andreeva, Dave Howorth, John-Marc Chandonia, Steven E Brenner, Tim J P Hubbard, Cyrus Chothia, and Alexey G Murzin. Data growth and its impact on the scop database: new developments. *Nucleic acids research*, 36:D419–D425, January 2008.
17. John-Marc Chandonia, Gary Hon, Nigel S Walker, Loredana Lo Conte, Patrice Koehl, Michael Levitt, and Steven E Brenner. The astral compendium in 2004. *Nucleic acids research*, 32:D189–D192, January 2004.
18. Grzegorz M Boratyn, Alejandro A Schffer, Richa Agarwala, Stephen F Altschul, David J Lipman, and Thomas L Madden. Domain enhanced lookup time accelerated blast. *Biology direct*, 7:12, April 2012.
19. Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33:2302–2309, 2005.
20. O Gotoh. An improved algorithm for matching biological sequences. *Journal of molecular biology*, 162:705–708, December 1982.
21. S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, March 1970.
22. A Biegert and J Sding. Sequence context-specific profiles for homology searching. *Proceedings of the National Academy of Sciences of the United States of America*, 106:3770–3775, March 2009.
23. Martyn D Winn, Charles C Ballard, Kevin D Cowtan, Eleanor J Dodson, Paul Emsley, Phil R Evans, Ronan M Keegan, Eugene B Krissinel, Andrew G W Leslie, Airlie McCoy, Stuart J McNicholas, Garib N Murshudov, Navraj S Pannu, Elizabeth A Potterton, Harold R Powell, Randy J Read, Alexei Vagin, and Keith S Wilson. Overview of the ccp4 suite and current developments. *Acta crystallographica. Section D, Biological crystallography*, 67:235–242, April 2011.
24. Emmanuel D Levy. A simple definition of structural regions in proteins and its use in analyzing interface evolution. *Journal of molecular biology*, 403:660–670, November 2010.
25. S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89:10915–10919, November 1992.
26. Kazunori Yamada and Kentaro Tomii. Revisiting amino acid substitution matrices for identifying distantly related proteins. *Bioinformatics (Oxford, England)*, 30:317–325, February 2014.
27. B Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12:85–94, February 1999.
28. Piero Fariselli, Ivan Rossi, Emidio Capriotti, and Rita Casadio. The wwwh of remote homolog detection: the state of the art. *Briefings in bioinformatics*, 8:78–87, March 2007.
29. Julian D Olden, Michael K Joy, and Russell G Death. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3):389–397, 2004.
30. Jianzhi Zhang and Jian-Rong Yang. Determinants of the rate of protein sequence evolution. *Nature reviews. Genetics*, 16:409–420, July 2015.
31. S Chakravarty and R Varadarajan. Residue depth: a novel parameter for the analysis of protein structure and stability. *Structure (London, England : 1993)*, 7:723–732, July 1999.
32. Akira R Kinjo and Ken Nishikawa. Eigenvalue analysis of amino acid substitution matrices reveals a sharp transition of the mode of sequence conservation in proteins. *Bioinformatics (Oxford, England)*, 20:2504–2508, November 2004.