Technical Report
TTIC-TR-2009-5

July 2009

A profile entropy dependent scoring function for protein threading

Jian Peng
TTI–Chicago
Jinbo Xu
TTI–Chicago
j3xu@tti-c.org

A profile entropy dependent scoring function for protein threading

Jian Peng and Jinbo Xu[1]
Toyota Technological Institute at Chicago, IL USA 60637

Abstract

Proteins play fundamental roles in all biological processes. Akin to the complete sequencing of genomes, complete descriptions of protein structures is a fundamental step towards understanding biological life, and is also highly relevant in the development of therapeutics and drugs. Computational prediction methods, especially template-based modeling, can quickly generate crude but useful structure models at a large scale. The challenge of template-based modeling lies in the recognition of correct templates and the generation of accurate sequence-template alignments. Evolutionary information (i.e., sequence profiles) has proved to be very powerful in detecting remote homologs, as demonstrated by the state-of-the-art profile-profile alignment method HHpred. However, there are still a lot of proteins without good sequence profiles. Here, we present a new protein threading method for proteins without good sequence profiles by nonlinearly combining evolutionary and non-evolutionary information. In particular, we model protein threading using a probabilistic graphical model Conditional (Markov) Random Fields (CRF) and training the model using a gradient tree boosting algorithm. The resultant threading model guides sequence-template alignment using a nonlinear scoring function consisting of a collection of regression trees. Each regression tree models a type of nonlinear relationship among different protein information. Experimental results indicate that when evolutionary information is not good enough, this new threading method greatly outperforms HHpred in terms of both alignment accuracy and fold recognition rate. The paradigm presented here for the design of a nonlinear scoring function is very general. It can also be applied to protein sequence alignment and RNA alignment.

Keywords: protein threading, conditional random fields, gradient tree boosting, regression tree, nonlinear scoring function

Introduction

Various genome sequencing projects have been producing DNA sequences that encode millions of proteins. These proteins play fundamental roles in all biological processes

[1] Please address all correspondence to Dr. Jinbo Xu at the Toyota Technological Institute at Chicago, USA. Email: j3xu@tti-c.org, Phone: 773 834 2511, Fax: 773 834 9881.

Author contributions: J.X. designed and performed research and wrote the paper; J.P. performed research and analyzed data.

including the maintenance of cellular integrity, metabolism, transcription/translation, and cell-cell communication. Akin to the complete sequencing of genomes, complete descriptions of protein structures is a fundamental step towards understanding biological life, and is also highly relevant medically in the development of therapeutics and drugs. However, there is still little knowledge of most protein structures because the experimental determination methods are costly, time-consuming and sometimes technically difficult.

Computational prediction methods, especially template-based modeling, can quickly generate crude but useful structure models at a large scale. Template-based modeling is becoming more powerful and important for structure prediction along with the increase of available experimental structures. Current PDB may contain all templates for single-domain proteins according to the seminal studies in [1]. Among 128 CASP8 targets (The 8th Critical Assessment of Structure Prediction), only 12 domains are officially considered as free modeling targets. These observations imply that the structures of many new proteins can be predicted using template-based methods. Recent CASP8 results also demonstrate that template-based modeling is the major technique for automated structure prediction. Zhang-Server [2, 3] achieves the best accuracy by integrating the predictions of nine template-based programs and refining models using contact and distance constraints extracted from multiple templates. Second to Zhang-Server, the mainly threading-based program RAPTOR [4-6] performs slightly better than Skolnick's TASSER [7, 8] and Baker's Robetta [9], although RAPTOR has no refinement module while the latter two programs extensively refined template-based models.

The error of a template-based model comes from fold recognition and sequence-template alignment, in addition to the structure difference between the sequence and the template. At higher sequence identity (>50%), template-based models can be accurate enough to be useful in virtual ligand screening [10, 11], designing site-directed mutagenesis experiments [12, 13], small ligand docking prediction [14, 15], and function prediction [16, 17]. When the sequence identity is below 30%, it is difficult to recognize the best template and generate accurate sequence-template alignments, so the resultant models have a wide range of accuracies [18, 19]. In their automated comparative modeling of all known protein sequences, Pieper et al have shown that 76% of all the models in MODBASE are from alignments in which the sequence and template share less than 30% sequence identity [20]. Therefore, to greatly enlarge the pool of useful structure models, it is essential to improve fold recognition and alignment method for the sequence and template with less than 30% sequence identity. Considering that currently there are millions of proteins without experimental structures, even a slight improvement of prediction accuracy can have a significant impact on large-scale automated structure prediction and its applications. As reported in [21], even 1% improvement in the accuracy of fold assessment for the ~4.2 million models in MODBASE can result in ~42,000 more models being correctly identified.

Given a template, the quality of a template-based model mainly depends on the alignment from which the model is built. The alignment accuracy depends on a scoring function, which is used to guide sequence-template alignment. A scoring function can

include both evolutionary and non-evolutionary information. Simple methods, such as BLAST [22] and FASTA [23], align two proteins using only primary sequence. These methods work for proteins with close homologs in the PDB and can only assign the fold for ~30% genes in microbial genomes [24]. The utilization of evolutionary information (i.e., sequence profiles) has clearly made a significant impact on the field of protein structure prediction. Almost all state-of-the-art homology modeling methods use sequence profiles [25-33]. HHpred [34], a method mainly using HMM-based sequence profiles and secondary structure, outperformed many protein threading methods in recent CASP events. Sequence profiles have also been combined with non-evolutionary/structure information to further improve alignment accuracy. For example, several leading methods such as SPARKS [35-38] and RAPTOR [5, 6] use a linear combination of structure information (e.g., secondary structure and solvent accessibility) and sequence profiles as their scoring functions. Zhang et al have shown that by using five structure features plus sequence profile, their threading program MUSTER [39] outperforms their profile-profile alignment program PPA [40]. However, MUSTER is slightly worse than HHpred in the CASP8 event.

The CASP8 result [2] also shows that HHpred did not perform as well as RAPTOR and MUSTER on the FR (Fold Recognition) targets. This indicates that in some cases, evolutionary information is not sufficient to align two proteins, especially when they are remote homologs. A close examination of these FR targets shows that their sequence profiles are not diverse enough, i.e., there are not many non-redundant homologs in the NCBI NR sequence database for these targets. We hypothesize that evolutionary information alone is sufficient to align the sequence and template if their sequence profiles are diverse enough. Otherwise, non-evolutionary information is necessary to achieve better alignment accuracy.

We test this hypothesis by developing a nonlinear scoring function for protein threading. This scoring function nonlinearly combines both evolutionary and non-evolutionary information to guide sequence-template alignment. A nonlinear scoring function is much more flexible than a linear function. When evolutionary information is strong enough, our nonlinear function will rely more on sequence profiles. Otherwise, we will count more on non-evolutionary information. A nonlinear scoring function is also good for the sequence and template with both much conserved and less conserved regions. In the highly conserved regions, we will use only primary sequence since it may worsen their alignment by using other information. In the less conserved regions, sequence signal is weak and we will use structure information to help with alignment. A nonlinear function can also effectively model correlation among protein features. Many protein features used in threading are highly correlated, e.g., predicted secondary structure vs. sequence profiles since the former is usually predicted from the latter.

We develop this nonlinear scoring function by modeling protein threading using a probabilistic graphical model Conditional (Markov) Random Fields (CRFs) [41] and training the model using a gradient tree boosting algorithm [42]. The resultant scoring

[2] http://prodata.swmed.edu/CASP8/evaluation/DomainsFR.First.html#tabl

function consists of only dozens of regression trees [3], which are automatically constructed during model training process to capture the nonlinear relationship among sequence and structure features. Each regression tree models a type of nonlinear relationship among various protein features. Although our nonlinear scoring function is much more flexible and powerful in guiding protein alignment, it can still be optimized quickly using a dynamic programming algorithm.

Experimental results indicate that by using a regression-tree-based nonlinear scoring function, we can effectively combine evolutionary and non-evolutionary information and greatly improve alignment accuracy and fold recognition rate for proteins without good sequence profiles. This paper analyzes the relationship between alignment accuracy and the diversity of sequence profiles. Our conclusion is that when sequence profiles are diverse enough, our method has the same performance as the best profile-profile alignment method HHpred. However, the less diverse the sequence profiles, the more advantage our method has than HHpred.

Results

Testing alignment accuracy by ProSup and SALIGN benchmarks

We tested the alignment accuracy of our new threading method using the Prosup [43] and SALIGN benchmarks [30]. The Prosup benchmark has 127 protein pairs with structural alignment generated by Prosup. The SALIGN benchmark contains 200 protein pairs with alignments generated by TM-align [44]. On average, two proteins in a pair share 20% sequence identity and 65% of structurally equivalent Cα atoms can be superposed with RMSD 3.5Å. The SALIGN benchmark is more difficult since it includes many pairs of proteins with very different sizes.

Table 1. Alignment accuracy (%) of BoostThreader and other methods on two datasets.

Prosup			SALIGN		
Methods	Exact	4-offset	Methods	Exact	4-offset
SPARKS	57.20		SPARKS	53.10	
SSALGN	58.30		SALIGN	56.40	
RAPTOR	61.30	79.32	RAPTOR	40.20	59.80
SP3	65.30	82.20	SP3	56.30	56.60
SP5	68.70		SP5	59.70	
HHpred	69.04	79.18	HHpred	62.98	75.93
BoostThreader	**76.08**	**90.20**	BoostThreader	**64.40**	**78.93**

To evaluate the alignment quality, we use the exact match accuracy which is computed as the percentage of one-to-one match positions in the benchmark pairs. We also evaluate the 4-offset match accuracy, which is defined as the percentage of the matches

[3] Please refer to http://en.wikipedia.org/wiki/Decision_tree or http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf for a brief introduction to regression trees.

within 4 positions shift away from one-to-one match. Table 1 compares the performance of various alignment methods on the two benchmarks. Our new method, denoted as BoostThreader, shows a significant improvement over the others. If only exact match accuracy is considered, the absolute improvement over RAPTOR is at least 15%. BoostThreader is also better than three leading threading programs SPARKS/SP3/SP5. SP5 uses a linear combination of sequence and structure features and a position-specific gap penalty. The relative improvements of BootThreader over SP3 and SP5 are approximately 16% and 10%, respectively. The relative improvements of BootThreader over HHpred, the best profile-profile alignment method, are approximately 10% and 2.2% on the Prosup and SALIGN benchmarks, respectively. The improvement of BoostThreader over HHpred on the SALIGN benchmark is not so significant because 1) many proteins in the SALIGN set have very good sequence profiles; and 2) the big size difference between a protein pair in SALIGN also makes it challenging to achieve much better alignment accuracy. Note that the results of SPARKS/SP3/SP5 are taken from [36] and the results of RAPTOR, HHpred and BoostThreader are generated by us using the same NCBI NR sequence database.

Alignment quality with respect to the diversity of sequence profiles

We compared the alignment quality of BoostThreader with HHpred [34] with respect to the diversity of sequence profiles. HHpred is a leading protein alignment method, which aligns two proteins using sequence profiles and secondary structure. HHpred uses an NEFF value to measure the diversity of the multiple sequence alignment from which the sequence profile is derived. Roughly speaking, the NEFF value of a protein can be interpreted as the number of non-redundant homologs in the NCBI NR sequence database. NEFF is calculated as the exponential of negative entropy averaged over all columns of the alignment, so NEFF is a real value ranging from 1 to 20 (i.e., the number of amino acid types in nature). The bigger NEFF is, the more diverse the sequence profiles and the more effective number of homologs in the database. We evaluate the quality of an alignment by first generating a 3D model from this alignment using MODELLER [45] and then calculating the TM score [44] of this 3D model. TM score measures the quality of a 3D model, ranging from 0 to 1. The higher the TM score, the better quality the 3D model.

Tested on the Prosup and SALIGN benchmarks, BoostThreader is much better than HHpred when either the target or the template does not have a very good NEFF value. As shown in Figures 1 and 2, when either the sequence or template have a small NEFF value (≤7), BoostThreader can generate much better alignment than HHpred in terms of the TM score of the 3D model derived from the alignment. When the NEFF values are between 7 and 9, BoostThreader is slightly better than HHpred on the Prosup benchmark. When both the sequence and template have an NEFF value at least 9, BoostThreader has similar performance as HHpred. The SALIGN benchmark contains some pairs of proteins with very different sizes. This makes it very challenging to achieve very good alignment accuracy. When NEFF≥4, the advantage of BoostThreader over HHpred on the SALIGN benchmark is not as significant as on the Prosup

benchmark. Figure 3 shows the distribution of the NEFF values of all the ~18,000 templates in the HHpred template database. Among ~18,000 HHpred templates, ~48% have NEFF no bigger than 7. This indicates that non-evolutionary information will be useful for about half of the templates.

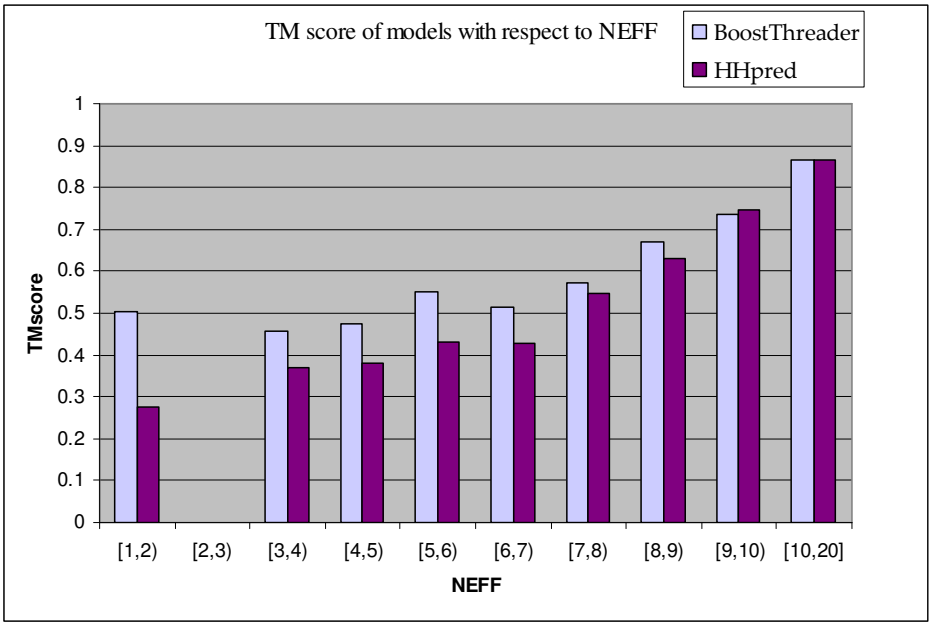

Figure 1. The average TM score of the 3D models with respect to the NEFF value. The models are generated by BoostThreader and HHpred for the protein pairs in the Prosup benchmark. The NEFF value of a protein pair is the minimum NEFF of the sequence and template.

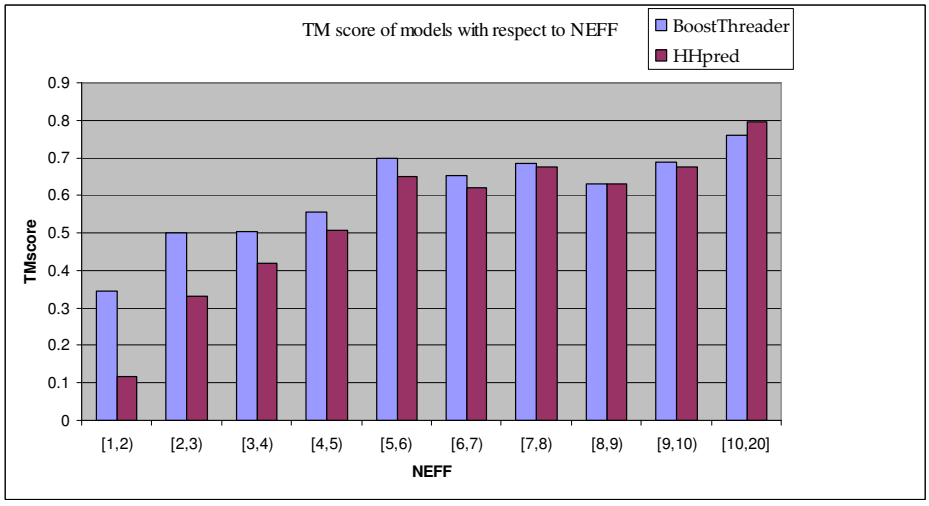

Figure 2. The average TM score of the 3D models with respect to the NEFF value. The models are generated by BoostThreader and HHpred for the protein pairs in the SALIGN benchmark. The NEFF value of a protein pair is the minimum NEFF of the sequence and template.

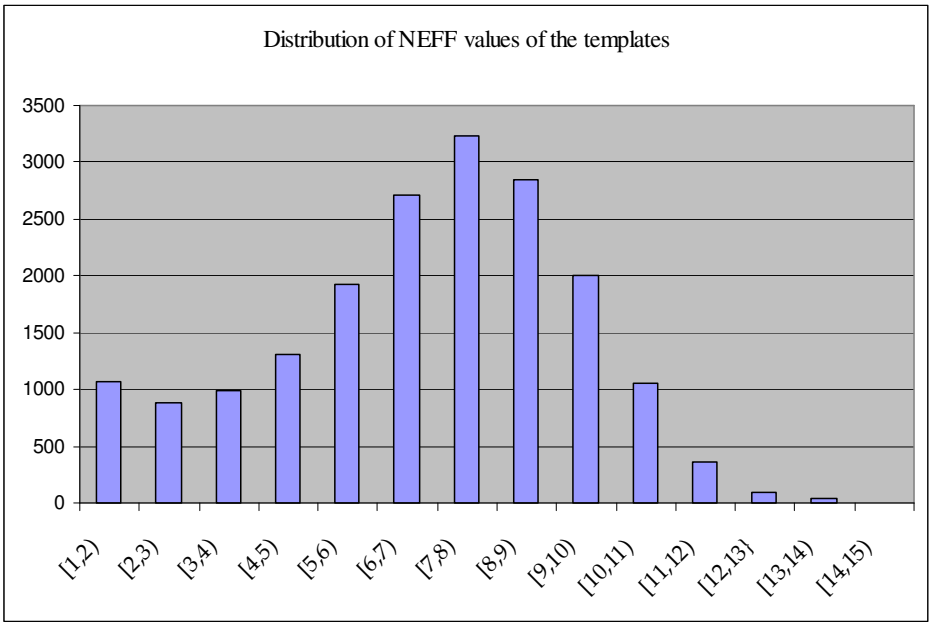

Figure 3. The NEFF distribution of the templates in the HHpred template database.

Testing model quality by CASP8 targets

To further demonstrate the advantage of BoostThreader when good sequence profiles are unavailable, we compared BoostThreader with HHpred2 on 17 CASP8 targets. Most of these targets have a small NEFF value. The models of HHpred2 are downloaded from the CASP8 website[4]. BoostThreader builds a 3D model for a target by first generating all sequence-template alignments and then choosing the best alignment using a model quality assessment method, which is used by RAPTOR in CASP8 (see RAPTOR abstract in the CASP8 abstract book). To do a fair comparison, BoostThreader used an NR database and a template database generated before CASP8 started (i.e., May 2008). As shown in Table 2, we examined the performance of HHpred2 and BoostThreader on 17 CM medium and hard CASP8 targets. BoostThreader performs better than HHpred2 on 10 out of 17 targets while HHpred2 is better on only 2 targets. BoostThreader has an average TM score of 0.541, which is 0.044 (i.e., 8.85%) better than HHpred2. In terms of the average GDT-TS score, BoostThreader is better than HHpred2 by 0.046 (i.e., 10.36%). The improvement of BoostThreader over HHpred2 comes from the improvement of both sequence-template alignment and template selection. The result in this table indicates that by using non-evolutionary information, BoostThreader can generate better models for those targets without good evolutionary information.

Table 2. Performance of BoostThreader and HHpred2 on 17 CASP8 targets.

Target	NEFF	HHpred2		BoostThreader	
		TMscore	GDT-TS	TMscore	GDT-TS
T0414	4.8	0.501	0.437	**0.520**	**0.470**

[4] http://predictioncenter.org/casp8/.

T0417	7.7	0.724	0.648	0.733	0.641
T0420	4.3	0.711	0.552	**0.772**	**0.629**
T0421	4.6	0.416	0.284	**0.490**	**0.354**
T0434	4.9	0.587	0.535	0.588	0.539
T0436	8.0	0.832	0.585	0.822	0.594
T0460	1.2	0.223	0.207	**0.284**	**0.277**
T0464	4.2	**0.438**	**0.455**	0.407	0.429
T0466	3.5	0.193	0.162	**0.238**	**0.220**
T0467	5.5	0.228	0.240	**0.414**	**0.405**
T0468	4.2	0.181	0.177	**0.369**	**0.349**
T0471	3.3	0.569	0.481	**0.664**	**0.587**
T0473	5.5	0.622	0.673	0.635	0.680
T0474	4.3	0.477	0.494	0.465	0.472
T0495	3.3	**0.475**	**0.418**	0.427	0.366
T0502	6.5	0.719	0.712	**0.733**	**0.735**
T0507	8.7	0.561	0.491	**0.643**	**0.574**
Average	4.97	0.497	0.444	0.541	0.490

Testing fold recognition with Lindahl benchmark

We also evaluated the fold recognition rate of our method BoostThreader on the Lindahl benchmark [46], which contains 976 proteins. Any two proteins in this set share less than 40% sequence identity. All-against-all threading of these proteins can generate 976× 975 pairs. After generating the alignments of all the pairs using BoostThreader, we rank all the templates for each sequence using a method similar to [47] and then evaluate the fold recognition rate of our method. When evaluating the performance in the superfamily level, all the templates similar in the family level are ignored. Similarly, when evaluating the performance in the fold level, all the templates similar in the superfamily or family level are ignored. As shown in Table 3, BoosThreader is much better than SP3/SP5, HHpred and RAPTOR, especially in the superfamily and fold levels. These three programs performed well in recent CASP events.

Table 3. Fold recognition rate (%) of BoostThreader and others. The PSI-BLAST, SPARKS, SP3, SP5 and HHpred results are from [36]. The FOLDpro results are from [48]. The RAPTOR and PROSPECT-II results are from [47].

	Family		Superfamily		Fold	
	Top1	Top5	Top1	Top5	Top1	Top5
PSIBLAST	71.2	72.3	27.4	27.9	4.0	4.7
PROSPECT-II	84.1	88.2	52.6	64.8	27.7	50.3
SPARKS	81.6	88.1	52.5	69.1	24.3	47.7
SP3	81.6	86.8	55.3	67.7	28.7	47.4
FOLDpro	85.0	89.9	55.5	70.0	26.5	48.3
SP5	81.6	87.0	59.9	70.2	37.4	58.6

HHpred	82.9	87.1	58.8	70.0	25.2	39.4
RAPTOR	86.6	89.3	56.3	69.0	38.2	58.7
BoostThreader	86.5	**90.5**	**66.1**	**76.4**	**42.6**	57.4

Note that we used the NCBI NR database before May 2008 to generate sequence profiles while the results of HHpred and SP3/SP5 were published in June 2008 by Zhang et al [36]. Therefore, BoostThreader's superior performance over HHpred and SP3/SP5 cannot be simply explained by the different versions of the NR database.

Discussion

Evolutionary information is much more powerful than primary sequence in detecting remote homologs, as evidenced by the HHpred method, which performed better than or as well as several top threading methods in recent CASP events. Although previous studies indicate that alignment accuracy can be improved by combining evolutionary information and structure information, it is unclear when non-evolutionary/structure information will help improve protein alignment accuracy. This paper studies the relationship between alignment accuracy and the diversity of sequence profiles and has shown that when good evolutionary information is unavailable from current sequence databases, we can improve alignment accuracy by using non-evolutionary information. When both the sequence and template have very good sequence profiles, it will not help much by using non-evolutionary information.

It is challenging to effectively combine evolutionary and non-evolutionary information to achieve the maximum alignment accuracy. This paper resolves this issue by formulating protein threading using a probabilistic graphical model Conditional (Markov) Random Fields (CRF) and regression trees. By using regression trees to represent the threading scoring function, our CRF-based threading method can make use of as many sequence and structure features as possible and accurately model their nonlinear interactions. Although nonlinear, such a scoring function can still be efficiently optimized by a dynamic programming algorithm. It takes less than half a second to generate the optimal alignment between a typical protein pair. Experimental results also demonstrate that by nonlinearly combining evolutionary and non-evolutionary information, we can greatly improve alignment accuracy over the leading profile-based alignment method HHpred [34]. The improved alignment accuracy also leads to the improvement of fold recognition rate and final model quality.

Currently, our threading model only considers state transition between two adjacent positions. A straight-forward generalization is to model state dependency among three adjacent positions. We can also model pairwise interaction between two non-adjacent positions. The challenge of modeling non-local interactions is that it is computationally hard to train and test such a model. Some approximation algorithms may be resorted.

In summary, the paradigm of nonlinearly combining various protein features offers greatly improved alignment quality and fold recognition rate, especially when good evolutionary information is unavailable. We believe that the result in this paper is

sufficient to warrant utilization of non-evolutionary information in protein modeling until all proteins can have a very good sequence profile. The paradigm presented here should be easily transferable to protein sequence alignment or even RNA alignment.

Materials and Methods

Regression-tree-based CRF threading model

We formulate the protein threading problem using a probabilistic graphical model Conditional (Markov) Random Fields (CRF) [41] and measure the sequence-template similarity using a set of regression trees, which take as input protein features and output the log-likelihood of an alignment state (i.e., match or gap). A regression tree consists of many paths, each specifying a rule to calculate the probability of an alignment state. One path can be as simple as "if (mutation score < -50), then the log-likelihood of a match state is $\ln 0.9$" or as complex as "if (-50 < mutation score < -10) and (secondary structure score > 0.9) and (solvent accessibility score > 0.6), then the log-likelihood of a match state is $\ln 0.7$". Regression trees can use different criteria to align different regions of the sequence and template. This is analogous to the position specific scoring matrix, which has different mutation potentials for the same amino acid at different positions. In addition, regression trees can also model the nonlinear relationship between an alignment state and protein features. By contrast, a simple linear scoring function used in existing threading methods is lack of these good characteristics.

Let s denote the target protein and its associated features, e.g., sequence profile, predicted secondary structure and solvent accessibility. Let t denote the template and its associated information, e.g., position-specific scoring matrix, solvent accessibility and secondary structure. Let $X = \{M, I_s, I_t\}$ be a set of three possible alignment states. Meanwhile, M indicates that two positions are matched and I_s and I_t indicate insertion at sequence and template, respectively. Let $a = \{a_1, a_2, ..., a_L\}$ ($a_i \in X$) denote an alignment between s and t where a_i represents the state at position i. Our CRF-based threading model defines the conditional probability of a given s and t as follows.

$$p(a \mid s,t) = \exp(\sum_i F(a_{i-1} \to a_i \mid s,t)) / Z(s,t)$$

Where $Z(s,t)$ is a normalizing factor. $F(a_{i-1} \to a_i \mid s,t)$ is a function that calculates the log-likelihood of the state transition from a_{i-1} to a_i given target and template information at position i. To model the nonlinear relationship between an alignment state and protein features, we represent $F(a_{i-1} \to a_i \mid s,t)$ as a linear combination of regression trees. Each regression tree is a nonlinear function of protein features, so the scoring function of this new threading model is nonlinear. This model is much more powerful than existing threading methods because a state transition in this model depends on a complex function of protein features while existing methods use only a linear function. Since this CRF model considers only state transition between two adjacent positions, the optimal alignment can still be efficiently calculated using dynamic programming.

Building regression trees

We train this CRF threading model by maximizing the occurring probability of a set of training alignments. To build the regression trees, we need to calculate the functional gradient of $p(a \mid s,t)$ with respect to $F(a_{i-1} \to a_i \mid s,t)$. Let u and v denote two alignment states. Using a similar technique as in [42], we can prove that the functional gradient of $\ln p(a \mid s,t)$ with respect to $F(u \to v \mid s,t)$ is given by

$$\frac{\partial \ln p(a \mid s,t)}{\partial F(u \to v \mid s,t)} = I(a_{i-1} = u, a_i = v) - P(a_{i-1} = u, a_i = v \mid s,t)$$

where $I(a_{i-1} = u, a_i = v)$ is a 0-1 function. Its value equals to 1 if and only if in the training alignment the state transition from $i-1$ and i is $u \to v$. $P(a_{i-1} = u, a_i = v \mid s,t)$ is the predicted probability of the state transition $u \to v$ under current threading model. $P(a_{i-1} = u, a_i = v \mid s,t)$ can be calculated using a forward-backward method (see section Implementation details). The functional gradient is easy to interpret. Given a training alignment, if the transition $u \to v$ is observed at position i, ideally the predicted probability $P(a_{i-1} = u, a_i = v \mid s,t)$ should be 1 in order to make $\frac{\partial \ln p(a \mid s,t)}{\partial F(u \to v \mid s,t)}$ be 0 and thus, to maximize $p(a \mid s,t)$. Similarly, if the transition is not observed, the predicted probability should be 0 to maximize $p(a \mid s,t)$.

Given an initial $F(u \to v \mid s,t)$, to maximize $p(a \mid s,t)$, we need to move $F(u \to v \mid s,t)$ along the gradient direction defined by the difference between $I(a_{i-1} = u, a_i = v)$ and $P(a_{i-1} = u, a_i = v \mid s,t)$. Since $F(u \to v \mid s,t)$ is a function taking as input protein features at each alignment position, the gradient direction is also a function with the same input variables. We can use a function $T(u \to v \mid s,t)$ to fit $I(a_{i-1} = u, a_i = v) - P(a_{i-1} = u, a_i = v \mid s,t)$ with the corresponding input values being the protein features at position i. Then $F(u \to v \mid s,t)$ is updated by $F(u \to v \mid s,t) + T(u \to v \mid s,t)$ where $T(u \to v \mid s,t)$ is the gradient direction. We can fit a given set of data using mathematical tools as simple as linear regression or as complex as neural networks. We use regression trees because they not only can capture nonlinear correlation among variables, but also are easy to interpret and computationally efficient.

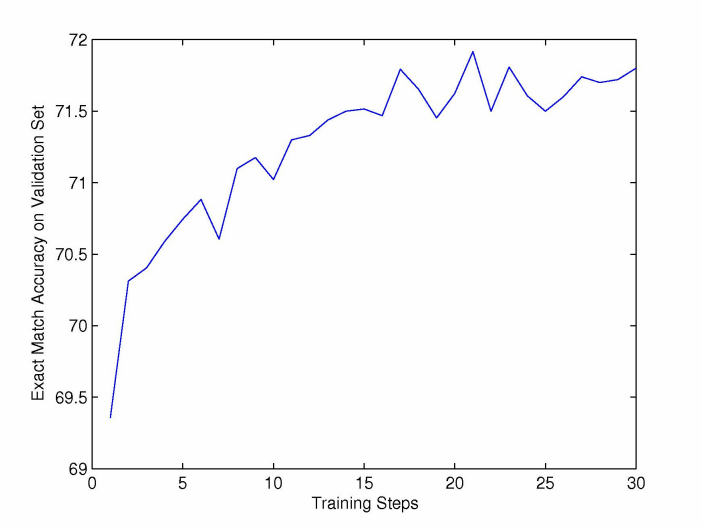

Figure 4. Training process of the CRF-based threading model.

We choose 66 protein pairs from the PDB as the training set and 50 pairs as the validation set. The NEFF (i.e., the diversity of sequence profiles) values of these 66 pairs of proteins are distributed uniformly between 1 and 11. In the training set, 46 pairs are in the same fold but different superfamily level by the SCOP classification [49]. The other 20 pairs are in the same superfamily but different family level. Any two proteins in the training and validation set have sequence identity less than 30%. The proteins used for model training and validation have no high sequence identity (30%) with the proteins in the Prosup [43] and SALIGN [30] benchmarks and the CASP8 targets. We use the structure alignment program TM-align [44] to build reference alignments for the training and validation protein pairs. The maximum training accuracy can be achieved by iteratively updating $F(u \to v \mid s,t)$ around 20 times, as shown in Figure 4. The training process is very efficient. It takes approximately two minutes to run a single training iteration. More training iterations will lead to more running time for aligning a protein pair. As a result, we choose the model trained after 21 iterations as our final threading model. For each state transition, the model has twenty-one regression trees with an average depth four.

It is challenging to build the regression trees due to the extremely unbalanced number of positive and negative examples. A training example is positive if its response value is positive, otherwise negative. Given a training pair with 200 residues in each protein and 150 aligned positions, the ratio between the number of positive examples and that of negative ones is approximately $\frac{150}{200 \times 200 \times 3}$. This will result in serious bias in regression tree training. We employed two strategies to resolve this issue. One is to add more weight to the positive examples and the other is that we randomly sample a small subset of negative examples for training [50]. To avoid overfitting the training alignments, we control the depth of a regression tree. We use an internal 5-fold cross-validation procedure to determine the best tree depth. The average tree depth is 4.

Sequence and structure features

We use both evolutionary information and non-evolutionary information to build regression trees for our CRF threading model. We generate position specific score matrix (PSSM) for a template and position specific frequency matrix (PSFM) for a target using PSI-BLAST with five iterations and E-value 0.001 [51]. Let $PSSM(i,a)$ denote the mutation potential for amino acid a at template position i and $PSFM(j,b)$ the occurring frequency of amino acid b at target position j. The secondary structure and solvent accessibility of a template is calculated by the DSSP program [52]. For a target protein, we use PSIPRED [53] and SSpro [54] to predict its secondary structure and solvent accessibility, respectively. We use NEFF to measure the diversity of sequence profiles, which can be calculated by the HHpred package.

Building regression trees for a match state. In addition to its left state, we use the following features to estimate the probability of template position i being aligned to target position j.

1. Sequence profile similarity. The profile similarity score between two aligned positions is calculated as $\sum_a PSSM(i,a) \times PSFM(j,a)$.
2. In order for the regression trees to tell the relative importance of evolutionary and non-evolutionary information, the diversity values (i.e., NEFF) of the sequence profiles are fed into the regression trees. When NEFF is large, regression trees will count more on sequence profile similarity. Otherwise, regression trees will also make use of non-evolutionary information to estimate the probability of an alignment state.
3. Structure-based score matrices. These score matrices have been studied by the Kihara group for protein alignment [55]. The first matrix is the correlation matrix of contact potential values. Each entry of the matrix is computed as the correlation coefficient of the pairwise contact potentials of two amino acids. The second matrix is the structure-derived substitution matrix [56, 55]. This matrix is calculated by the same procedure as the BLOSUM matrices [57, 58], based upon the structure alignments of structurally similar protein pairs. When the sequence or template does not have very good sequence profiles, the non-homology information in these two matrices can help improve alignment.
4. Contact capacity score. The contact capacity potential measures the capability of a residue making a certain number of contacts with other residues in a protein. The two residues are in physical contact if the spatial distance between their C_β atoms is smaller than 8Å. Let $CC(a,k)$ denote the contact potential of amino acid a having k contacts (see Section 3 in [47]). The contact capacity score is calculated by $\sum_a CC(a,c) \times PSFM(j,a)$ where c is the number of contacts at template position i.
5. Environmental fitness score. This score measures how well it is to align one target residue to a template local environment, which is defined by a combination of three secondary structure types and three solvent accessibility states. Let $F(env,a)$ denote the environment fitness potential for amino acid a being in a local environment env

(see Section 3 in [47]). The environment fitness score is given by $\sum_a F(env,a) \times PSFM(j,a)$.

6. Secondary structure consistency score. Supposing the secondary structure type at template position i is ss, the predicted likelihood of ss at target position j is used as the secondary structure consistency score.
7. Solvent accessibility consistency score. This is a binary feature used to indicate if the template position and the target position are in the same solvent accessibility state.

Building regression trees for a gap state. The simplest gap penalty model is an affine function, which specifies that gap open and extension at any position has equal probability. Some studies indicate that the probability of a gap is related to its local sequence and structure context. For example, SSALIGN [59] uses a context-specific gap penalty model, in which the probability of a gap depends on secondary structure and solvent accessibility. Some methods use a gap penalty model derived from evolutionary information. For example, HHpred [34], SP5 [36] and Ellrott et al [60] use a position-specific gap penalty model, which is derived from statistical analysis of gaps in a multiple sequence alignment. These studies have shown that the probability of a gap is related to multiple factors. In this article, we use the following features to estimate the probability of a gap state.

In addition to its left state, the occurring probability of an insertion state at the template depends on the following features: secondary structure type, solvent accessibility, amino acid identity and hydropathy count [61]. Similarly, the occurring probability of an insertion state at the target depends on the following features: predicted secondary structure likelihood scores, predicted solvent accessibility, amino acid identity and hydropathy count. We also use position-specific gap frequency as one feature, which is extracted from multiple sequence alignment. The probability of a gap event is calculated as the ratio between the number of the gap events and the number of sequences in the multiple sequence alignment.

Implementation details

Once a CRF model has been trained, we can find the best alignment a by maximizing $P(a \mid s,t)$ using a dynamic programming algorithm. This step is similar to all the HMM-based sequence alignment procedure. The best sequence-template alignment can be computed by the well-known Viterbi algorithm [62], which has the advantage that it does not need to compute the normalizer $Z(s,t)$.

We can calculate $P(a_{i-1} = u, a_i = v \mid s,t)$ using a forward-backward method. Let $\alpha(v,i)$ and $\beta(v,i)$ denote the probabilities of reaching state v at position i, starting from the N-terminal and C-terminal of the alignment, respectively. Both $\alpha(v,i)$ and $\beta(v,i)$ can be recursively calculated as follows.

$$\alpha(v,1) = \exp(F(\phi \to v \mid s,t)) \ , \ \alpha(v,i) = \sum_u \exp(F(u \to v \mid s,t))\alpha(u,i-1)$$

Where ϕ represents a dummy state.

$$\beta(v,i) = 1,\ \beta(v,i) = \sum_u \exp(F(v \to u \mid s,t))\beta(u,i+1)$$

Then $P(a_{i-1} = u, a_i = v \mid s,t)$ can be calculated as $\frac{\alpha(u,i-1)\exp(F(u \to v \mid s,t)\beta(v,i)}{Z(s,t)}$ and $Z(s,t)$ can be calculated as $\sum_u \alpha(u,0)\beta(u,0)$.

Appendix: Availability of Datasets. All benchmark datasets and training datasets can be downloaded from http://ttic.uchicago.edu/~jinbo/BoostThreader/

Acknowledgements

The authors are grateful to Liefeng Bo and Kristian Kersting for their help with the gradient tree boosting technique and to Johannes Soding for his help with HHpred.

References

1. Zhang, Y. and J. Skolnick, *The protein structure prediction problem could be solved using the current PDB library.* Proceedings of National Academy Sciences, USA, 2005. **102**(4): p. 1029-1034.
2. Zhang, Y., *Template-based modeling and free modeling by I-TASSER in CASP7.* Proteins-Structure Function and Bioinformatics, 2007. **69**: p. 108-117.
3. Zhang, Y., *I-TASSER server for protein 3D structure prediction.* Bmc Bioinformatics, 2008. **9**: p. -.
4. Xu, J. and M. Li, *Assessment of RAPTOR's linear programming approach in CAFASP3.* Proteins-Structure Function and Genetics, 2003. **53**(6): p. 579-584.
5. Xu, J., M. Li, D. Kim, and Y. Xu, *RAPTOR: optimal protein threading by linear programming.* Journal of Bioinformatics and Computational Biology, 2003. **1**(1): p. 95-117.
6. Xu, J., M. Li, G. Lin, D. Kim, and Y. Xu, *Protein threading by linear programming.* Pac Symp Biocomput, 2003: p. 264-275.
7. Zhang, Y., A.K. Arakaki, and J.R. Skolnick, *TASSER: An automated method for the prediction of protein tertiary structures in CASP6.* Proteins: Structure Function and Bioinformatics, 2005. **61**: p. 91-98.
8. Zhang, Y. and J. Skolnick, *Automated structure prediction of weakly homologous proteins on a genomic scale.* PNAS, 2004. **101**(20): p. 7594-7599.
9. Kim, D.E., D. Chivian, and D. Baker, *Protein structure prediction and analysis using the Robetta server.* Nucleic Acids Research, 2004. **32**: p. W526-W531.
10. Bjelic, S. and J. Aqvist, *Computational prediction of structure, substrate binding mode, mechanism, and rate for a malaria protease with a novel type of active site.* Biochemistry, 2004. **43**(46): p. 14521-14528.
11. Caffrey, C.R., L. Placha, C. Barinka, M. Hradilek, J. Dostal, M. Sajid, J.H. McKerrow, P. Majer, J. Konvalinka, and J. Vondrasek, *Homology modeling and SAR analysis of Schistosoma japonicum cathepsin D (SjCD). with statin inhibitors identify a unique active site steric barrier with potential for the design of specific inhibitors.* Biological Chemistry, 2005. **386**(4): p. 339-349.

12. Skowronek, K.J., J. Kosinski, and J.M. Bujnicki, *Theoretical model of restriction endonuclease HpaI in complex with DNA, predicted by fold recognition and validated by site-directed mutagenesis.* Proteins-Structure Function and Bioinformatics, 2006. **63**(4): p. 1059-1068.
13. Wells, G.A., L.M. Birkholtz, F. Joubert, R.D. Walter, and A.I. Louw, *Novel properties of malarial S-adenosylmethionine decarboxylase as revealed by structural modelling.* Journal of Molecular Graphics & Modelling, 2006. **24**(4): p. 307-318.
14. Vakser, I.A., *Low-resolution docking: Prediction of complexes for underdetermined structures.* Biopolymers, 1996. **39**(3): p. 455-464.
15. Wojciechowski, M. and J. Skolnick, *Docking of small ligands to low-resolution and theoretically predicted receptor structures.* Journal of Computational Chemistry, 2002. **23**(1): p. 189-197.
16. Baker, D. and A. Sali, *Protein structure prediction and structural genomics.* Science, 2001. **294**(5540): p. 93-96.
17. Skolnick, J., J.S. Fetrow, and A. Kolinski, *Structural genomics and its importance for gene function analysis.* Nature Biotechnology, 2000. **18**(3): p. 283-287.
18. Chakravarty, S., S. Godbole, B. Zhang, S. Berger, and R. Sanchez, *Systematic analysis of the effect of multiple templates on the accuracy of comparative models of protein structure.* Bmc Structural Biology, 2008. **8**: p. -.
19. Sanchez, R., U. Pieper, F. Melo, N. Eswar, M.A. Marti-Renom, M.S. Madhusudhan, N. Mirkovic, and A. Sali, *Protein structure modeling for structural genomics.* Nature Structural Biology, 2000. **7**: p. 986-990.
20. Pieper, U., N. Eswar, F.P. Davis, H. Braberg, M.S. Madhusudhan, A. Rossi, M. Marti-Renom, R. Karchin, B.M. Webb, D. Eramian, M.Y. Shen, L. Kelly, F. Melo, and A. Sali, *MODBASE: a database of annotated comparative protein structure models and associated resources.* Nucleic Acids Research, 2006. **34**: p. D291-D295.
21. Melo, F. and A. Sali, *Fold assessment for comparative protein structure modeling.* Protein Science, 2007. **16**(11): p. 2412-2426.
22. Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, *Basic Local Alignment Search Tool.* Journal of Molecular Biology, 1990. **215**(3): p. 403-410.
23. Pearson, W.R. and D.J. Lipman, *Improved Tools for Biological Sequence Comparison.* Proceedings of the National Academy of Sciences of the United States of America, 1988. **85**(8): p. 2444-2448.
24. Wang, Y.L., S. Bryant, R. Tatusov, and T. Tatusova, *Links from genome proteins to known 3-D structures.* Genome Research, 2000. **10**(10): p. 1643-1647.
25. Ginalski, K., M.v. Grotthuss, N.V. Grishin, and L. Rychlewski, *Detecting distant homology with Meta-BASIC.* Nucleic Acids Research, 2004(32): p. W576-W581.
26. Ginalski, K., J. Pas, L.S. Wyrwicz, M. von Grotthuss, J.M. Bujnicki, and L. Rychlewski, *ORFeus: detection of distant homology using sequence profiles and predicted secondary structure.* Nucleic Acids Research, 2003. **31**(13): p. 3804-3807.
27. Gough, J., K. Karplus, R. Hughey, and C. Chothia, *Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure.* Journal of Molecular Biology, 2001. **313**(4): p. 903-919.

28. Han, S.J., B.C. Lee, S.T. Yu, C.S. Jeong, S. Lee, and D. Kim, *Fold recognition by combining profile-profile alignment and support vector machine.* Bioinformatics, 2005. **21**(11): p. 2667-2673.
29. Karplus, K., C. Barrett, M. Cline, M. Diekhans, L. Grate, and R. Hughey, *Predicting protein structure using only sequence information.* Proteins: Structure, Function and Genetics, 1999: p. 121-125.
30. Marti-Renom, M.A., M.S. Madhusudhan, and A. Sali, *Alignment of protein sequences by their profiles.* Protein Science, 2004. **13**(4): p. 1071-1087.
31. Rychlewski, L., L. Jaroszewski, W.Z. Li, and A. Godzik, *Comparison of sequence profiles. Strategies for structural predictions using sequence information.* Protein Science, 2000. **9**(2): p. 232-241.
32. Tomii, K. and Y. Akiyama, *FORTE: a profile-profile comparison tool for protein fold recognition.* Bioinformatics, 2004. **20**(4): p. 594-595.
33. Yona, G. and M. Levitt, *Within the twilight zone: A sensitive profile-profile comparison tool based on information theory.* Journal of Molecular Biology, 2002. **315**(5): p. 1257-1275.
34. Soding, J., *Protein homology detection by HMM-HMM comparison.* Bioinformatics, 2005. **21**(7): p. 951-960.
35. Zhang, C., S. Liu, H. Zhou, and Y. Zhou, *An accurate, residue-level, pair potential of mean force for folding and binding based on the distance-scaled, ideal-gas reference state.* Protein Sci, 2004. **13**(2): p. 400-411.
36. Zhang, W., S. Liu, and Y. Zhou, *SP5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model.* PLoS ONE, 2008. **3**(6).
37. Zhou, H.Y. and Y.Q. Zhou, *Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition.* Proteins-Structure Function and Bioinformatics, 2004. **55**(4): p. 1005-1013.
38. Zhou, H.Y. and Y.Q. Zhou, *Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments.* Proteins-Structure Function and Bioinformatics, 2005. **58**(2): p. 321-328.
39. Wu, S.T. and Y. Zhang, *MUSTER: Improving protein sequence profile-profile alignments by using multiple sources of structure information.* Proteins: Structure, Function and Bioinformatics, 2008. **72**(2): p. 547-556.
40. Wu, S.T., J. Skolnick, and Y. Zhang, *Ab initio modeling of small proteins by iterative TASSER simulations.* Bmc Biology, 2007. **5**: p. -.
41. Lafferty, J., A. McCallum, and F. Pereira. *Conditional Random Fields: probabilistic models for segmenting and labeling sequence data*. in *Proc. 18th International Conf. on Machine Learning*. 2001: Morgan Kaufmann, San Francisco, CA
42. Dietterich, T., A. Ashenfelter, and Y. Bulatov. *Training conditional random fields via gradient tree boosting*. in *In Proceedings of the 21th International Conference on Machine Learning (ICML)*. 2004
43. Lackner, P., W.A. Koppensteiner, M.J. Sippl, and F.S. Domingues, *ProSup: a refined tool for protein structure alignment.* Protein Engineering, 2000. **13**(11): p. 745-752.

44. Zhang, Y. and J. Skolnick, *TM-align: a protein structure alignment algorithm based on the TM-score.* Nucleic Acids Research, 2005. **33**(7): p. 2302-2309.
45. Sali, A., *Comparative Protein Modeling by Satisfaction of Spatial Restraints.* Molecular Medicine Today, 1995. **1**(6): p. 270-277.
46. Lindahl, E. and A. Elofsson, *Identification of related proteins on family, superfamily and fold level.* Journal of Molecular Biology, 2000. **295**(3): p. 613-625.
47. Xu, J., *Protein fold recognition by predicted alignment accuracy.* IEEE/ACM Trans. on Computational Biology and Bioinformatics, 2005. **2**(2): p. 157-165.
48. Cheng, J.L. and P. Baldi, *A machine learning information retrieval approach to protein fold recognition.* Bioinformatics, 2006. **22**(12): p. 1456-1463.
49. Murzin, A.G., S.E. Brenner, T. Hubbard, and C. Chothia, *SCOP: a structural classification of proteins database for the investigation of sequences and structures.* J Mol Biol, 1995. **247**(4): p. 536-40.
50. Gutmann, B. and K. Kersting, *Stratified Gradient Boosting for Fast Training of Conditional Random Fields*, in *the 6th International Workshop on Multi-Relational Data Mining*. 2007: Warsaw, Poland. p. 56-68.
51. Altschul, S.F., T.L. Madden, A.A. Schaffer, J.H. 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, 1997. **25**(17): p. 3389-3402.
52. Kabsch, W. and C. Sander, *Dictionary of Protein Secondary Structure - Pattern-Recognition of Hydrogen-Bonded and Geometrical Features.* Biopolymers, 1983. **22**(12): p. 2577-2637.
53. Jones, D.T., *Protein secondary structure prediction based on position-specific scoring matrices.* Journal of Molecular Biology, 1999. **292**(2): p. 195-202.
54. Cheng, J., A.Z. Randall, M.J. Sweredoski, and P. Baldi, *SCRATCH: a protein structure and structural feature prediction server.* Nucleic Acids Research, 2005. **33**: p. W72-W76.
55. Tan, Y.H., H. Huang, and D. Kihara, *Statistical potential-based amino acid similarity matrices for aligning distantly related protein sequences.* Proteins-Structure Function and Bioinformatics, 2006. **64**(3): p. 587-600.
56. Prlic, A., F.S. Domingues, and M.J. Sippl, *Structure-derived substitution matrices for alignment of distantly related sequences.* Protein Engineering, 2000. **13**(8): p. 545-550.
57. Henikoff, S. and J.G. Henikoff, *Amino acid substitution matrices from protein blocks.* Proc Natl Acad Sci U S A, 1992. **89**(22): p. 10915-10919.
58. Henikoff, S. and J.G. Henikoff, *Performance evaluation of amino acid substitution matrices.* Proteins, 1993. **17**(1): p. 49-61.
59. Qiu, J. and R. Elber, *SSALN: An alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs.* Proteins-Structure Function and Bioinformatics, 2006. **62**(4): p. 881-891.
60. Ellrott, K., J.T. Guo, V. Olman, and Y. Xu, *Improvement in protein sequence-structure alignment using insertion/deletion frequency arrays.* Computational systems bioinformatics / Life Sciences Society. Computational Systems Bioinformatics Conference, 2007. **6**: p. 335-342.

61. Do, C.B., S.S. Gross, and S. Batzoglou, *CONTRAlign: Discriminative training for protein sequence alignment*, in *The 10th Annual International Conference on Research in Computational Molecular Biology*. 2006, Springer: Venice, Italy. p. 160-174.
62. Rabiner, L.R., *A Tutorial on Hidden Markov-Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.