Published in final edited form as:
Proteins. 2011 ; 79(Suppl 10): 161–171. doi:10.1002/prot.23175.

RaptorX: exploiting structure information for protein alignment by statistical inference

Jian Peng and **Jinbo Xu**[*]
Toyota Technological Institute at Chicago

Abstract

This paper presents RaptorX, a statistical method for template-based protein modeling that improves alignment accuracy by exploiting structural information in a single or multiple templates. RaptorX consists of three major components: single-template threading, alignment quality prediction and multiple-template threading. This paper summarizes the methods employed by RaptorX and presents its CASP9 result analysis, aiming to identify major bottlenecks with RaptorX and template-based modeling and hopefully directions for further study. Our results show that template structural information helps a lot with both single-template and multiple-template protein threading especially when closely-related templates are unavailable and there is still large room for improvement in both alignment and template selection. The RaptorX web server is available at http://raptorx.uchicago.edu.

Keywords

single-template threading; multiple-template threading; alignment quality prediction; probabilistic alignment; multiple protein alignment; CASP

Introduction

RaptorX is totally different from our previous threading program RAPTOR, which aligns a sequence to a template by employing linear programming to minimize a given threading scoring function[1,2]. By contrast, RaptorX employs a statistical learning method to design a new threading scoring function, aiming at better measuring the compatibility between a target sequence and a template structure. In addition to single-template threading, RaptorX also has a multiple-template threading component and contains a new module for alignment quality prediction. Our results show that RaptorX indeed has much better alignment accuracy than RAPTOR[3,4].

RaptorX is designed to address two "alignment" challenges facing template-based protein modeling. One is how to align a target to its template when they have a sparse sequence profile (i.e., no sufficient amount of information in homologs). In this case, a profile-based alignment method may not work well. The other is how to improve sequence-template alignment accuracy using more reliable template structural alignments as bridge when at least two similar templates are available for a target. RaptorX addresses these two challenges by exploiting template structure information in several unique ways.

Many homology modeling and protein threading methods have been developed for sequence-template alignment[5–20]. These methods have two major issues in dealing with

[*] Correspondence: Name: Jinbo Xu, Address: Toyota Technological Institute at Chicago, 6045 S. Kenwood Ave. Chicago, IL, US 60637, Phone: 773 834 2511, Fax: 773 834 9881, j3xu@ttic.edu.

distantly-related sequence and template. One is that these methods use a linear scoring function to guide the alignment of a sequence to its template. A linear function cannot deal well with correlation among protein features, although many features are indeed correlated (e.g., secondary structure vs. solvent accessibility). The other issue is that these methods heavily depend on sequence profile. Sequence profile has proved to be very powerful in detecting remote homologs and generating accurate alignments, as demonstrated by the excellent HHpred program[7]. However, information in a sequence profile may not be sufficient especially when the profile is sparse. To address these two issues, RaptorX employs a nonlinear scoring function to combine homologous information (i.e., sequence profile) and structure information in a very flexible way. When proteins under consideration have high-quality sequence profile, RaptorX counts more on profile information, otherwise on structure information to improve alignment accuracy.

When multiple similar templates are available for a single target, RaptorX improves pairwise target-template alignment accuracy through exploiting template structural similarity (i.e. geometrical information), in addition to increase alignment coverage for the target (by copying the most similar regions from different templates). Nearly all existing multiple-template methods generate an alignment between the target and its multiple templates by either simply assembling pairwise alignments into a single multiple alignment using the target as an anchor[21,22] or employing some multiple sequence alignment tools such as T-Coffee[23], MUSCLE[24] and ProbCons[25]. Neither way effectively uses template structural alignments, which are more reliable than sequence-template alignments, as bridge to build sequence-to-multiple-template alignments, so they usually do not fare much better than when only a single template is available for the target. By contrast, RaptorX takes advantage of template structural alignments to align a target sequence simultaneously to multiple templates through a novel statistical inference method. Two multiple protein alignment programs 3DCoffee[26] and PROMAL3D[27], which can also be used to align a single sequence to multiple templates, indeed take advantage of template structural alignments in building a multiple alignment, but our experimental results show that they do not fare well when the target and templates are not closely-related[28].

Methods

Following PSI-BLAST and HHpred, we use NEFF to measure the amount of information in the sequence profile of a protein. NEFF ranges from 1 to 20 and can be interpreted as the expected number of amino acid substitutions at each sequence position. A sparse sequence profile (i.e., a profile with a small NEFF value) usually leads to less accurate secondary structure prediction and less accurate alignment [29].

Single-template protein threading

We use a probabilistic model to formulate the pairwise sequence-template alignment problem. See Peng and Xu[3,4] for the technical details. Our probabilistic model uses a regression-tree-based nonlinear scoring function to measure the similarity between two proteins. A regression tree consists of a collection of rules to calculate the probability of an alignment. One rule can be as simple as “if (mutation score < −50), then the log-likelihood of two residues being aligned 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 two residues being aligned is ln 0.7 ”.

Our scoring function is much more sensitive than the widely-used linear function since our function can model protein feature correlation. Our method also enables us to 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.

Our method differs from others in that we use NEFF to adjust the relative importance of homologous and structure information so that the former will not dominate the latter. When proteins have large NEFF, RaptorX counts more on sequence profile information; otherwise, structure information. Our method uses both context-specific and position-specific gap penalty and then use NEFF to determine their relative importance. If NEFF is large, we will rely more on position-specific gap penalty derived from the alignment of sequence homologs; otherwise, context-specific gap penalty. Our context-specific gap penalty depends on (predicted) secondary structure type, (predicted) solvent accessibility, amino acid identity, hydropathy count and if a residue is in the core region or not.

Alignment quality prediction

We predict the absolute quality of a pairwise sequence-template alignment using neural network and then use the predicted quality to rank all the templates for a specific target. The quality of a pairwise sequence-template alignment is defined as the TMscore[30] of the 3D model built from this alignment by MODELLER (with default parameters). However, our method does not need to actually build a 3D model in order to predict alignment quality because only information in an alignment is used for quality prediction. This saves time for 3D model building and thus, we can predict alignment quality very quickly. Our old RAPTOR program uses an SVM method to predict the number of correctly aligned positions in an alignment[31]. RaptorX differs from RAPTOR in that RaptorX predicts TMscore, a better quality measure, and also uses a better set of alignment features.

Let $A(i)$ denote the target residue aligned to the ith template residue. $A(i)$ is empty if the template residue is not aligned to any target residue. Let PSSM and PSFM denote the position specific scoring matrix and position specific frequency matrix for the template and the sequence, respectively. PSSM and PSFM are two slightly different representations of sequence profile. Let PSSM(i) and PSFM(i) denote sequence profiles at template position i and sequence position i, respectively. Both PSSM(i) and PSFM(i) are a vector of 20 real values encoding occurring frequency of the 20 amino acids.

NEFF values—See description at the beginning of this section.

Sequence profile similarity—At one aligned position, the sequence profile similarity score is calculated as the inner product of the template and target profile vectors. The total profile similarity score is the sum of scores at all the aligned positions. We also calculate the histogram distribution of the per-position profile similarity scores by dividing the score into 8 equal-width intervals. The scores are normalized by the number of aligned positions and the target length, respectively. Using these two different normalizations, we can take into consideration the impact of the gap length.

Statistical potential-based sequence similarity—We use the *CC50* matrix developed by Kihara group[32] to calculate sequence similarity. *CC50* is a symmetric matrix and derived from statistical potentials. Each element *CC50[a][b]* measures the similarity between two amino acids *a* and *b*. We use an array of 20 values, denoted as *v[20]*, to store the sequence similarity scores, each corresponding to an amino acid type. For a given amino acid *a*, *v[a]* is calculated as $\Sigma_i \partial[s(i)= a]\times CC50[a][t(i)]$where i runs over all aligned positions and *s(i)* and *t(i)* are the target and template amino acids at *i*. $\partial[s(i)= a]$ is an indicator function and equal to 1 if and only if *s(i)* is *a*. The 20 values (i.e., *v[20]*) are normalized by the target length and the number of aligned positions, respectively, to generate 40 features.

Secondary structure similarity score—For each type of secondary structure, three different scoring methods are used:

- Exact match score. It is equal to 1 if the secondary structure types at an aligned position are identical, otherwise 0.
- Log-odds score. It is calculated as $\log(P(ss))$ where ss is the secondary structure type at template position i and $P(ss)$ is the predicted likelihood of ss at sequence position $A(i)$.
- Confidence score generated by PSIPRED.

These scores are normalized by the sequence length and also the number of aligned positions.

Solvent accessibility—The total solvent accessibility score is defined as the number of aligned positions at which the template and the target have different solvent accessibility status. The score is also normalized by the number of aligned positions to generate another feature.

Contact capacity—The contact capacity potential measures the capability of a residue making a certain number of contacts with other residues in a protein. The contact capacity score is calculated as $\Sigma_a CC(a, k) PSFM(A(i),a)$ where k is the number of contacts at template position i and $CC(a, k)$ is the contact potential of amino acid a having k contacts. We also calculate the histogram distribution of this score using 8 equal-width bins. This score is also normalized by the number of aligned positions and the target length, respectively.

Environmental fitness—This score measures how well it is to align one sequence residue to a local structure environment, which is defined by a combination of three secondary structure types and three solvent accessibility states. Let $F(env, a)$ denote the potential of amino acid a being in a local environment env. The environment fitness score is calculated by $\Sigma_a F(env_i, a) PSFM(A(i), a)$. Again we calculate the histogram distribution of this score and normalize it using the number of aligned positions and the target length, respectively.

Sequence identity—It is the fraction of identical residues in the alignment.

Alignment length—We use the number of aligned positions and its ratio to the target length as two features.

Gap—All gap lengths are divided into four intervals: 1–5, 6–11, 12–20 and [20, ∞] and we also differentiate template gaps from target gaps. Given an alignment, we calculate the histogram distribution of its gap lengths. The gaps at the two ends of an alignment are ignored.

Multiple-template protein threading

Instead of simply using the star alignment algorithm to assemble all the pairwise target-template alignments into a target-to-multiple-template alignment, RaptorX realigns the target simultaneously to all the templates to fix some errors in the pairwise alignments by exploiting template structure similarity. Given two pairwise alignments S-T1 and S-T2 where S is the target sequence and T1 and T2 are two templates, an alignment between T1 and T2 can be derived from S-T1 and S-T2 using S as an anchor. Such a T1–T2 alignment should be consistent with the T1 to T2 alignment generated by a structure alignment program. Otherwise, there may be errors in S-T1 and S-T2 alignments since template-template structure alignment usually is more accurate than sequence-template alignment. That is, we can use template-template structure alignment to improve sequence-template alignment by enforcing consistency among all pairwise alignments. See Peng and Xu[28] for the details of our realignment algorithm and here we only briefly summarize the major procedure.

Our method consists of two major components: selection of multiple templates and alignment of the target to multiple templates. Currently RaptorX uses a simple and conservative strategy to choose templates for a specific target, based upon the template ranking list generated by the predicted alignment quality (from high to low). We first exclude all the templates not among the top 20 from consideration. Second, a template with predicted alignment quality 10% less than the highest is also excluded. Finally, a template is removed if its TMscore with the first-ranked template is less than 0.65 or the highest predicted quality. By using only mutually similar templates, we can avoid introducing large inconsistency into our alignment algorithm and MODELLER.

To realign the target to multiple templates, we build an initial probabilistic alignment matrix (PAM) for a target-template pair using RaptorX single-template alignment algorithm. Each entry in a PAM is the (marginal) alignment probability of two residues, which can be calculated using the forward-backward algorithm[4]. One PAM encodes all possible alignments of two proteins by probability and thus, contains more information than a single deterministic alignment. We also generate a PAM, consisting of binary values, for any two templates using structure alignment programs TMalign[33] and/or Matt[34]. Then, we run our probabilistic-consistency transformation algorithm to iteratively adjust all the PAMs, to maximize the consistency among all PAMs and to improve the alignment between two proteins using the others. Finally, we generate a target-to-multiple-template alignment from the updated PAMs using progressive alignment and iterative refinement.

Building 3D models from alignments

We tested our methods through three different CASP9 servers. RaptorX-MSA and RaptorX-Boost use MODELLER and Skolnick's TASSERLite[35] to build 3D models from a given alignment, respectively. RaptorX-Boost performed worse than RaptorX-MSA in CASP9, maybe because we did not correctly use TASSER. RaptorX is a combination of RaptorX-MSA and RaptorX-Boost. When the target appears to be easy, RaptorX used the results from RaptorX-MSA, otherwise from RaptorX-Boost. When no reliable templates can be identified for a target, RaptorX used our in-house free modeling program[36] to generate five models (for only two targets). We also use this free modeling program to fold the unaligned two ends of a protein target. When doing so, for the middle region we fix only the C_α positions but not the other atoms. This is why our CASP9 models have very good C_α accuracy, but bad full-atom accuracy. This paper focuses only on RaptorX-MSA results and does not evaluate our free-modeling procedure since generally it does not help.

Summary of methods—For a specific target RaptorX first aligns it to each of the templates using the single-template alignment algorithm. Then RaptorX predicts the alignment quality and ranks all the templates by predicted quality descendingly. If the target is not suitable for multiple-template threading, RaptorX builds a 3D model for the target from the pairwise alignment with the highest predicted quality. Otherwise, RaptorX runs multiple-template threading for the target and builds a corresponding 3D model.

Results

Evaluation of single-template alignment accuracy

We evaluate our single-template alignment algorithm using 38 TBM (template-based modeling) CASP9 targets for which RaptorX submitted single-template models. To evaluate how much room is left for improvement in alignment, for a given target and its first-ranked template (according to the ranking by RaptorX), we use TMalign to generate their pairwise structure alignment, assuming that the native structure of the target is known. We compare the 3D models built by MODELLER from the TMalign alignments with the 3D models built from RaptorX single-template alignments, as shown in Figure 1. For many targets, RaptorX alignments are still significantly worse than TMalign alignments. This implies that there is still large improvement room for single-template alignment, especially for those targets without close templates.

Our single-template alignment algorithm failed to generate accurate alignments for a bunch of targets such as T0568 and T0603. The best template RaptorX identified for T0568 is 2p9rA and the 3D model built from RaptorX alignment to this template has TMscore of 0.354. However, the 3D model built from TMalign alignment to this template has TMscore 0.572. We can improve the RaptorX model by using two templates 2p9rA and 2p5nA, which yield a 3D model with TMscore 0.441.

The best template identified by RaptorX for T0603 is 3godaA and the 3D model built from RaptorX alignment has TMscore 0.633. However, the 3D model built from TMalign alignment has TMscore 0.700. If another two top templates 2yzsA and 3lfxF are combined with 3godA, RaptorX can generate a 3D model with TMscore 0.748. RaptorX did not submit multiple-template models for T0568 and T0603 because RaptorX misclassified them as single-template targets. Therefore, it is important to develop a better strategy to decide if a target is suitable for multiple-template modeling or not.

Evaluation of alignment quality prediction

As shown in the following sections, alignment quality prediction is a very important component for RaptorX. To evaluate how well we can predict the quality (TMscore) of a pairwise sequence-template alignment, we compare the real and predicted quality of the models built from the first-ranked templates for 111 targets, as shown in Figure 2. The predicted quality is highly correlated with the real quality (correlation coefficient = 0.85). However, for a good percentage of targets the absolute prediction error is still larger than 0.05. For some targets, the absolute prediction error is larger than 0.10 or even 0.20. We also observed that when the real model quality is less than 0.6, the predicted quality tends to slightly underestimate the real value; otherwise overestimate.

Evaluation of template selection

RaptorX uses the predicted alignment quality to rank templates. To evaluate the ranking performance of the predicted alignment quality, we examine how well RaptorX can identify the best available template for a given target. In order to conduct a rigorous evaluation, we have to build a 3D model from each pairwise target-template alignment using MODELLER, which is time-consuming since there are >20,000 templates. Instead, for each target we build 3D models only from the top 10 templates (according to the ranking by RaptorX) and then examine ranking of the models built from the frist-ranked templates, using the models from the top 10 templates as reference.

As shown in Figure 3, for a large percentage of targets (even some hard targets), the models built from the first-ranked templates is or very close to the best out of the top 5 or 10 models. That is, the predicted alignment quality can rank the templates very well. However, RaptorX indeed failed to identify the best available templates for some hard targets, as shown in Figure 3(b). For example, RaptorX identified a very bad template for T0576, which leads to a 3D model with TMscore only ~0.2, although RaptorX generated a model with TMscore ~0.7 from the top 10 templates. The reason RaptorX failed is that T0576 has a sparse sequence profile and the traning data set for our quality prediction algorithm does not contain enough number of sequences with sparse sequence profile.

Multiple-template models vs. single-template models

RaptorX generated multiple-template models for 48 CASP9 targets, each of which has at least 2 good templates. There are also several other targets suitable for multiple-template modeling, but RaptorX only submitted single-template models for them. As shown in Figure 4, using multiple-template threading RaptorX can generate models for most of the 48 targets better than single-template models even if the best template is used. Here by “best” we mean the best out of the multiple templates used by RaptorX to build a multiple-template model. In particular, the accumulative TMscores of the first-template models, the best-template models and the multiple-template models are 34.042, 34.770 and 35.473, respectively. This result indicates that using multiple templates RaptorX can indeed improve modeling accuracy for most targets.

The improvement in modeling accuracy by multiple-template threading arises from three factors: inclusion of the best single templates, better alignment and larger coverage. Table 1 shows the modeling results on the 99 CASP8 and CASP9 targets with multiple good templates. The models built from the first-ranked single templates with alignments generated by single-template threading in total have TMscore 72.86 and GDT 6265.68, respectively. The models built from the best single templates with alignments generated by single-template threading in total have TMscore 74.06 and GDT 6381.54, respectively. Therefore, by using the best single templates, we can gain ~1.20 and ~116 for TMscore and GDT, respectively. When alignments generated by multiple-template threading are used, the models built from the best single templates have TMscore 74.85 and GDT 6465.12, respectively. That is, the TMscore and GDT improvements from better alignment are ~0.79 and ~84.0, respectively. Table 1 also shows that the models built from multiple templates (i.e., best single templates plus others) in total have TMscore 75.68 and GDT 6585.70, respectively. This indicates that the TMscore and GDT improvements from larger coverage are 0.83 and 120.0, respectively. If only the 48 CASP9 targets are evaluated, the improvement from inclusion of the best single templates is larger than that from better alignment, which in turn is larger than that from larger coverage.

RaptorX failed on one target T0589, for which RaptorX identified four mutually similar templates 3lc0A, 1httD, 1z7mD and 1wu7A (their pairwise TMscore > 0.8). These four templates yield 4 single-template alignments with predicted quality ~0.8, but the best template 1wu7A was not ranked first. By combing these four templates, RaptorX generated a 3D model with TMscore 0.752, significantly lower than the quality (TMscore 0.852) of the model built from 1wu7A. To examine what went wrong with our multiple-template method, we extract the pairwise T0589-1wu7A alignment from RaptorX’s multiple-template alignment and run MODELLER to generate a 3D model from this extracted pairwise alignment. Such a 3D model has TMscore 0.853, which indicates that RaptorX did not worsen the T0589-1wu7A alignment by doing multiple-template threading. The reason that MODELLER did not produce a good 3D model from the multiple-template alignment is more likely that MODELLER was misled by three not-so-good templates 3lc0A, 1httD and 1z7mD. The single-template models built from 3lc0A, 1httD and 1z7mD have TMscore 0.742, 0.638 and 0.602, respectively, much worse than the single-template model from 1wu7A. However, according to the predicted alignment quality 1wu7A is not significantly better than the other three templates, so RaptorX used 4 templates instead of only 1wu7A to build the model. To overcome this issue, we need a better alignment quality prediction algorithm to tell 1wu7A apart from the other three templates. Otherwise we may have to use a 3D model building tool that is more robust to bad templates than MODELLER.

Evaluation of multiple-template alignment accuracy

A natural question to ask is if there is still any improvement room for multiple-template threading method, especially for the hard targets? First we want to point out that template selection is still challenging for multiple-template threading. Different combinations of templates may lead to models with very different quality (e.g., T0589). Here we focus on the alignment aspect of multiple-template threading.

To examine the multiple-template alignment quality, we conduct an experiment using 48 TBM targets for which RaptorX submitted multiple-template models. For each target we use our in-house multiple structure alignment tool 3DCOMB (submitted for publication) to generate its alignment to the same set of templates used by RaptorX, using the native structures of all the proteins under consideration. Then we run MODELLER to produce a 3D model for the target from the 3DCOMB alignment. We compare the 3D models built from the 3DCOMB alignments with those generated by RaptorX multiple-template threading. The accumulative GDT of the 3DCOMB models are much higher than that of the RaptorX models. In particular, the 3DCOMB models for more than 20 targets on average have GDT at least 5 units better than their corresponding RaptorX models (see Figure 5). This result indicates that there is still large improvement room for multiple-template alignment.

Comparison with HHpred—In this subsection we compare RaptorX with the best profile-profile alignment method HHpred using two datasets: CASP9 data and a large PDB25 dataset. The PDB25 set consists of 6125 non-redundant protein chains generated by the PISCES server (http://dunbrack.fccc.edu/PISCES.php). Any two chains in this set share no more than 25% sequence identity. All the proteins in this PDB25 set are used as templates and 1000 of them are randomly chosen as our test targets. We run both RaptorX (single-template method) and HHpred to predict the 3D structure for each of the 1000 target proteins using the 6125 templates. Note that when predicting structure for one target protein, we remove itself from the template list.

Fig. 6 shows the GDT-TS of the 3D models generated by both RaptorX and HHpred for all the CASP9 targets. The GDT-TS scores are directly taken from Zhang's CASP9 assessment website (http://zhanglab.ccmb.med.umich.edu/casp9/). Each target is evaluated by the whole chain instead of by domains so that we can exclude impact of different domain parsing methods. As shown in Fig. 6, RaptorX is obviously better than HHpred for hard targets with GDT-TS less than 30. For easy targets, HHpred is slightly better than RaptorX.

Figures 7 and 8 show the quality of the first-ranked 3D models generated by RaptorX and HHpred. On average RaptorX obtains a TMscore of 0.534 while HHpred 0.516 and their difference is significant (P-value=1.77E-09). Figures 7 and 8 show that RaptorX outperforms HHpred on targets across almost the whole TMscore range, which is different from what is displayed in Fig. 6. The discrepancy is due to two reasons. One is that all the templates in the PDB25 set are not so close to the target while some CASP9 targets have very close templates. RaptorX tends to perform better when close templates are not available. The other is that HHpred in CASP9 uses a better method to build a 3D model from an alignment and RaptorX in CASP9 also uses multiple-template threading while Figures 7 and 8 are based upon only single-template models generated by MODELLER. As shown in Fig. 8, the numbers of targets for which RaptorX generates models with TMscore at least 0.05 and 0.10 better than HHpred are 255 and 133, respectively. In contrast, the numbers of targets for which HHpred generates models with TMscore at least 0.05 and 0.10 better than RaptorX are 133 and 67, respectively. Fig. 7 shows that both RaptorX and HHpred did very badly on a small number of targets. For example, for about 8 targets, HHpred generates 3D models with TMscore around 0.6 while RaptorX obtains TMscore between 0.2 and 0.4. Similarly, for about 15 targets, RaptorX generates 3D models with TMscore around 0.6 while HHpred only obtains TMscore between 0.2 and 0.4. We will do further study on these targets and hopefully can fix them.

Issues with building 3D models from alignments

Even if we can generate a pretty good alignment, sometimes it is very challenging to build a high-quality 3D model from the alignment. We identified three targets for which RaptorX generated very good alignments, but failed to obtain good 3D models by running MODELLER. One possible reason may be that we did not use MODELLER in the best way.

1. T0566. The best available template for this target is 1usvF or 1usuB. RaptorX was able to identify the best template 1usvF and produced a 3D model with TMscore 0.759, which is not ranked well among all the server models. The structure alignment generated by TMalign for T0566 and 1usvF has TMscore 0.740. The 3D model built by MODELLER from the TMalign alignment has TMscore 0.751, even slightly worse than the RaptorX model. This may imply that the RaptorX alignment is good, but MODELLER failed to build a good 3D model from it.

2. T0586. RaptorX identified a good template 3by6A and generated a model with TMscore 0.7390, which is better than the model (TMscore 0.699) built by MODELLER from TMalign alignment. However, many other groups generate much better models than RaptorX. In particular, chunk-TASSER[37] submitted the best server model using only 3by6A as the template.

3. T0580. RaptorX identified several good templates 1e2bA, 3czcA, 1tvmA and 2fewA. The predicted quality of the alignment between T0580 and 1e2bA is much higher than others (TMscore 0.768). The pairwise structural similarity between these templates is relatively low (TMscore ~0.6). Therefore, RaptorX used only 1e2bA to build a single-template model with TMscore 0.728. By using multiple templates, RaptorX can generate a model with TMscore 0.815, which is still worse than the model (TMscore 0.901) built by BAKER group using only 1iibB (same as 1e2bA) as a template.

We are still investigating these three targets and hopefully identify plausible reasons for our failures.

Conclusions and Discussions

We have presented a new protein modeling program RaptorX to replace our previous program RAPTOR. RaptorX has much better alignment accuracy than RAPTOR because of several unique features: 1) RaptorX takes into consideration the correlation among protein features; 2) RaptorX deals well with proteins with sparse sequence profile by leveraging structure information; and 3) RaptorX employs a novel multiple-template threading algorithm to exploit template structure similarity to improve alignment accuracy.

Although RaptorX significantly excels RAPTOR in alignment accuracy, our analysis of the CASP9 results indicates that there is still large improvement room for both single-template and multiple-template threading, in both template selection and sequence-template alignment. We may improve alignment of proteins with sparse sequence profile by somehow artificially enriching profile. For example, we can use protein design programs to generate hypothetical sequences for a given protein (or family). These sequences then can be combined with natural sequences to build an enriched sequence profile[38,39]. The issue with this method is that it usually takes a long time to generate sequences using a protein design program. Very recently, simulated evolution, a more efficient method, is also proposed to enrich sequence profile and seems to help[40]. Maybe we can incorporate these methods into our current probabilistic alignment model to further improve alignment accuracy.

RaptorX demonstrates that when sequence profile is sparse, we can use structure information to improve alignment accuracy[3]. Due to the enlargement of the NR database, many proteins now have a very dense sequence profile. Using a very dense profile may worsen the alignment accuracy due to lose of specificity, as reported in[41]. We need to investigate a strategy to generate a profile best for alignment accuracy.

There is still improvement room for multiple-template alignment. As shown in Figure 5, there is still a gap between RaptorX multiple-template alignments and 3DCOMB structure alignments even for some targets with good templates, which implies that multiple-template alignment may be improved further. Selection of templates for multiple-template threading is also quite challenging and needs more investigation. Our current strategy requires that the TMscore between two templates is at least 0.65. Can we break this barrier down to a smaller value, say 0.4? In particular, when the target does not have a very good template, (e.g., the highest predicted alignment quality is less than 0.5), can we use two templates with TMscore 0.4~0.65 to improve alignment accuracy? Finally, when the first-ranked template of a target is much better than other templates, can we still use structure information in other not-so-good templates to help with the alignment of the first-ranked template? In addition to align a single target sequence to multiple templates, can we also improve alignment accuracy by aligning multiple sequences (homologous to the target) to multiple templates?

A more accurate method is also needed for alignment quality prediction, which is critical for template ranking and selection of templates for multiple-template threading. As shown before, because of bad quality prediction, we used a wrong combination of 4 templates to build a multiple-template model for T0589, which generates a 3D model significantly worse than the best-single-template model. RaptorX also failed badly in picking up the best template for an easy target T0576 due to bad quality prediction.

In summary, our CASP9 results indicate that there is still large improvement room for RaptorX and template-based modeling. New methods are needed for the three major aspects of template-based modeling: sequence-template alignment, alignment quality prediction and template ranking.

Acknowledgments

This work is financially supported by the National Institutes of Health grant R01GM089753 (to JX) and the National Science Foundation grant DBI-0960390 (to JX). The authors are grateful to Dr. Tobin Sosnick and anonymous reviewers for helpful suggestions and proofreading the manuscript. The authors are also grateful to the computational resources support from UChicago Teraport, TeraGrid, SHARCNet and Open Science Grid.

References

1. Xu J, Li M, Lin G, Kim D, Xu Y. Protein threading by linear programming. Pac Symp Biocomput. 2003:264–275. [PubMed: 12603034]

2. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. Journal of Bioinformatics and Computational Biology. 2003; 1(1):95–117. [PubMed: 15290783]

3. Peng J, Xu J. Low-homology protein threading. Bioinformatics. 2010; 26(12):i294–300. [PubMed: 20529920]

4. Peng J, Xu J. Boosting Protein Threading Accuracy. Lect Notes Comput Sci. 2009; 5541:31. [PubMed: 20169009]

5. Zhou H, Zhou Y. SPARKS 2 and SP3 servers in CASP6. Proteins. 2005; 61 (Suppl 7):152–156. [PubMed: 16187357]

6. Wu S, Zhang Y. MUSTER: Improving protein sequence profile-profile alignments by using multiple sources of structure information. Proteins. 2008; 72(2):547–556. [PubMed: 18247410]

7. Soding J. Protein homology detection by HMM-HMM comparison. Bioinformatics. 2005; 21(7): 951–960. [PubMed: 15531603]

8. Shi J, Blundell TL, Mizuguchi K. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J Mol Biol. 2001; 310(1):243–257. [PubMed: 11419950]

9. Ginalski K, Grotthuss Mv, Grishin NV, Rychlewski L. Detecting distant homology with Meta-BASIC. Nucleic Acids Research. 2004; (32):W576–W581. [PubMed: 15215454]

10. Han SJ, Lee BC, Yu ST, Jeong CS, Lee S, Kim D. Fold recognition by combining profile-profile alignment and support vector machine. Bioinformatics. 2005; 21(11):2667–2673. [PubMed: 15769835]

11. Jaroszewski L, Rychlewski L, Li ZW, Li WZ, Godzik A. FFAS03: a server for profile-profile sequence alignments. Nucleic Acids Research. 2005; 33:W284–W288. [PubMed: 15980471]

12. Jones DT. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. Journal of Molecular Biology. 1999; 287(4):797–815. [PubMed: 10191147]

13. Kelley LA, MacCallum RM, Sternberg MJE. Enhanced genome annotation using structural profiles in the program 3D-PSSM. Journal of Molecular Biology. 2000; 299(2):499–520. [PubMed: 10860755]

14. McGuffin LJ, Jones DT. Improvement of the GenTHREADER method for genomic fold recognition. Bioinformatics. 2003; 19(7):874–881. [PubMed: 12724298]

15. Prlic A, Domingues FS, Sippl MJ. Structure-derived substitution matrices for alignment of distantly related sequences. Protein Engineering. 2000; 13(8):545–550. [PubMed: 10964983]

16. Qiu J, Elber R. 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):881–891.

17. Sadreyev RI, Grishin NV. Quality of alignment comparison by COMPASS improves with inclusion of diverse confident homologs. Bioinformatics. 2004; 20(6):818–828. [PubMed: 14751996]

18. Skolnick J, Kihara D. Defrosting the frozen approximation: PROSPECTOR -A new approach to threading. Proteins: Structure, Function and Genetics. 2001; 42(3):319–331.

19. Zhang W, Liu S, Zhou Y. SP5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model. PLoS ONE. 2008; 3(6)

20. Zhou HY, Zhou YQ. 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):1005–1013.

21. Fernandez-Fuentes N, Madrid-Aliste CJ, Rai BK, Fajardo JE, Fiser A. M4T: a comparative protein structure modeling server. Nucleic Acids Res. 2007; 35(Web Server issue):W363–368. [PubMed: 17517764]

22. Cheng JL. A multi-template combination algorithm for protein comparative modeling. Bmc Structural Biology. 2008; 8

23. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000; 302(1):205–217. [PubMed: 10964570]

24. Edgar RC. MUSCLE:multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004; 32(5):1792–1797. [PubMed: 15034147]

25. Caffrey CR, Placha L, Barinka C, Hradilek M, Dostal J, Sajid M, McKerrow JH, Majer P, Konvalinka J, Vondrasek J. 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):339–349. [PubMed: 15899696]

26. O'Sullivan O, Suhre K, Abergel C, Higgins DG, Notredame C. 3DCoffee: combining protein sequences and structures within multiple sequence alignments. J Mol Biol. 2004; 340(2):385–395. [PubMed: 15201059]

27. Pei JM, Kim BH, Grishin NV. PROMALS3D: a tool for multiple protein sequence and structure alignments. NucleicAcids Research. 2008; 36(7):2295–2300.

28. Peng J, Xu J. a multiple-template approach to protein threading. PROTEINS: Structure, Function and Bioinformatics. 2011

29. Wang, Z.; Zhao, F.; Peng, J.; xu, J. Protein 8-class secondary structure prediction using Conditional Neural Fields. Hong Kong: IEEE; 2010. p. 109-114.

30. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins-Structure Function and Bioinformatics. 2004; 57(4):702–710.

31. Xu J. Fold recognition by predicted alignment accuracy. IEEE/ACM Trans Comput Biol Bioinform. 2005; 2(2):157–165. [PubMed: 17044180]

32. Tan YH, Huang H, Kihara D. Statistical potential-based amino acid similarity matrices for aligning distantly related protein sequences. Proteins-Structure Function and Bioinformatics. 2006; 64(3): 587–600.

33. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. 2005; 33(7):2302–2309. [PubMed: 15849316]

34. Menke M, Berger B, Cowen L. Matt: local flexibility aids protein multiple structure alignment. PLoS Comput Biol. 2008; 4(1):e10. [PubMed: 18193941]

35. Skolnick J, Pandit SB, Zhang Y. TASSER-Lite: An automated tool for protein comparative modeling. Biophys J. 2006; 91(11):4180–4190. [PubMed: 16963505]

36. Xu JB, Zhao F, Peng JA. Fragment-free approach to protein folding using conditional neural fields. Bioinformatics. 2010; 26(12):i310–i317. [PubMed: 20529922]

37. Zhou HY, Skolnick J. Ab initio protein structure prediction using Chunk-TASSER. Biophys J. 2007; 93(5):1510–1518. [PubMed: 17496016]

38. Busch MSA, Sedano A, Simonson T. Computational Protein Design: Validation and Possible Relevance as a Tool for Homology Searching and Fold Recognition. PLoS ONE. 2010; 5(5)

39. Simonson T, Busch MSA, Mignon D. Computational protein design as a tool for fold recognition. Proteins-Structure Function and Bioinformatics. 2009; 77(1):139–158.

40. Kumar A, Cowen L. Augmented training of hidden Markov models to recognize remote homologs via simulated evolution. Bioinformatics. 2009; 25(13):1602–1608. [PubMed: 19389731]

41. Chubb D, Jefferys BR, Sternberg MJ, Kelley LA. Sequencing delivers diminishing returns for homology detection: implications for mapping the protein universe. Bioinformatics. 2010; 26(21): 2664–2671. [PubMed: 20843957]

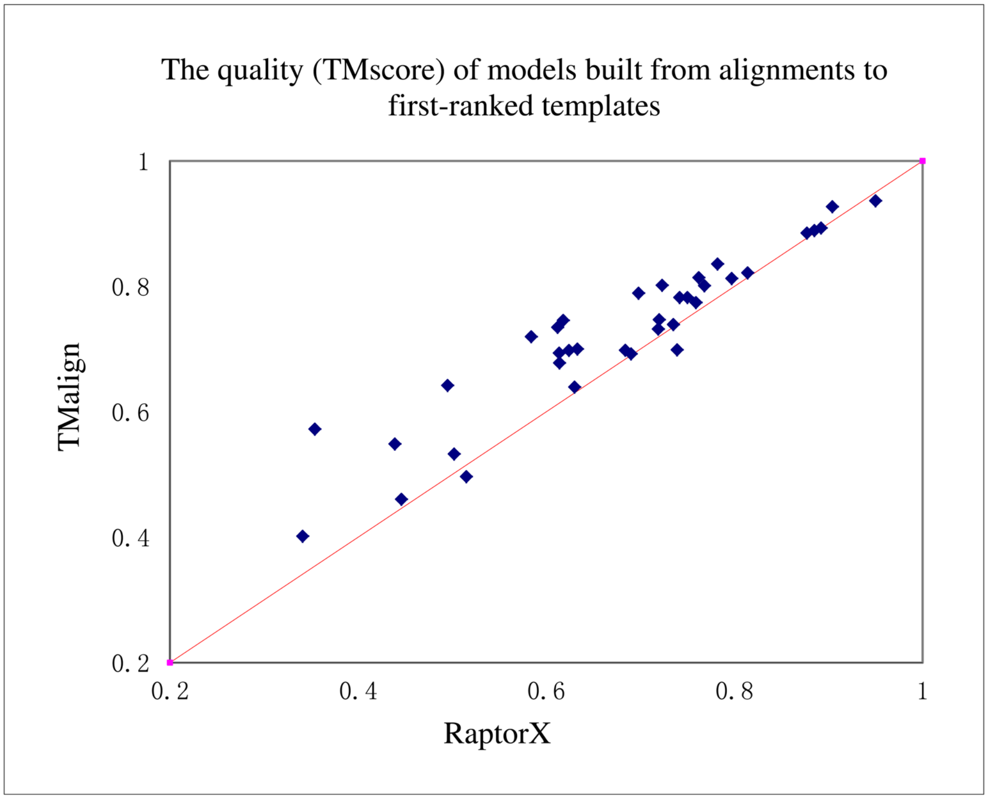

Figure 1.

This figure compares RaptorX alignments with TMalign alignments for 38 CASP9 targets. A point above the diagonal line indicates that the RaptorX alignment is worse than the corresponding TMalign alignment.

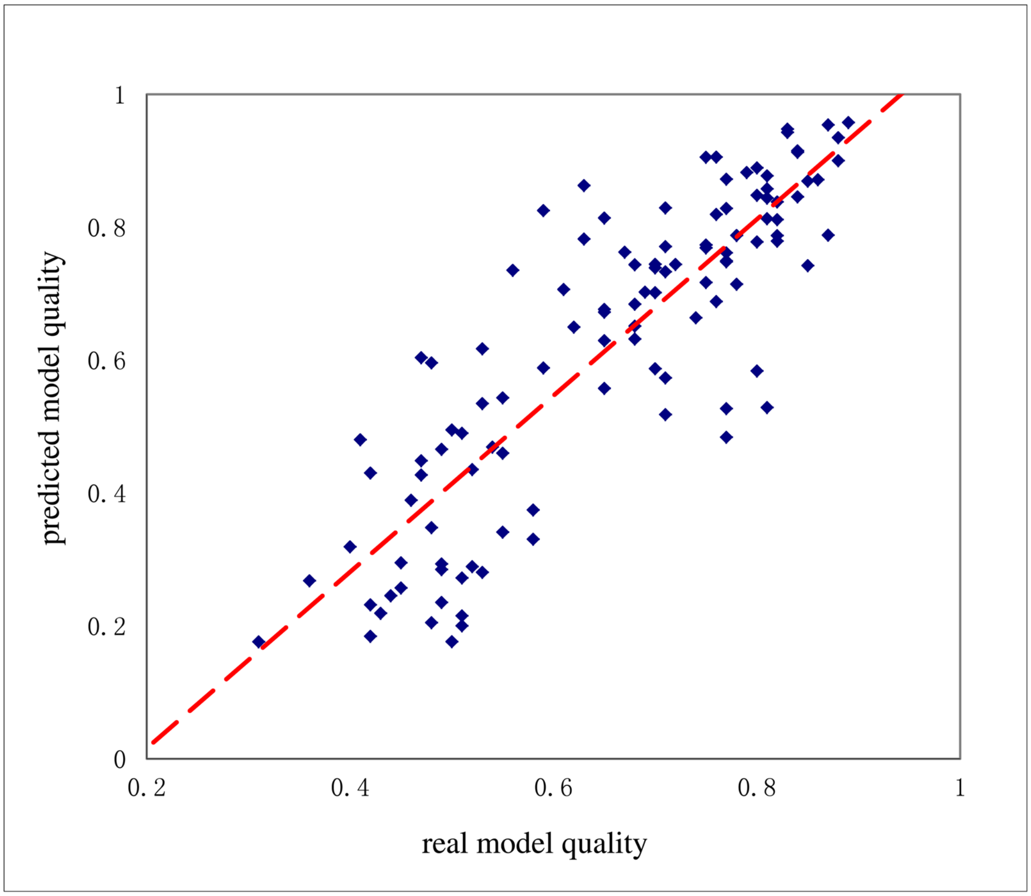

Figure 2.

This figure compares the predicted and real quality of the models built from the first-ranked templates. A point above the diagnoal line indicates that for a specific model its predicted quality overestimates the real value.

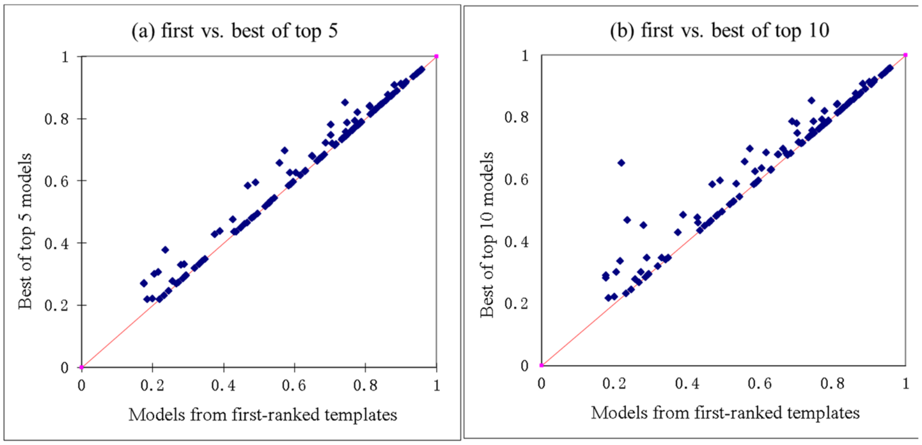

Figure 3.

This figure illustrates the quality of the models built from the first-ranked templates and those from the best out of the top 5 or top 10 templates. A point on or close to the diagnoal line indicates that the first-ranked model is (or very close to) the best out of the top 5 or 10 models.

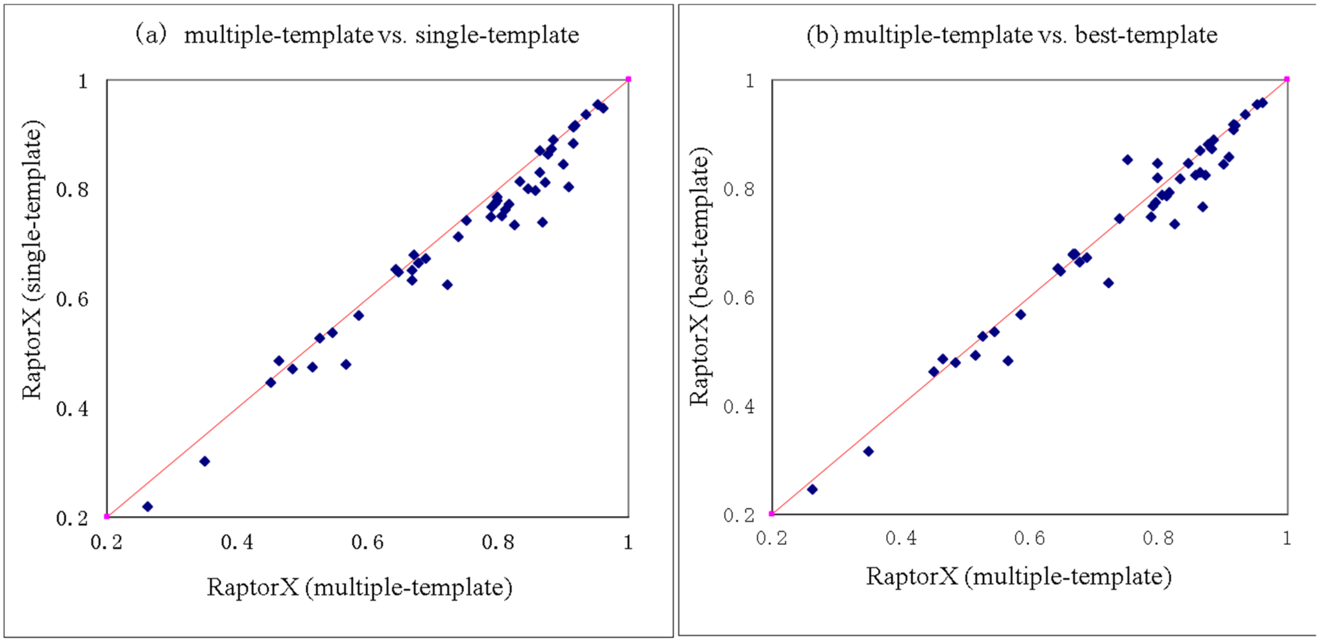

Figure 4.
This figure illustrates the advantage of multiple-template models over single-template models: (a) multiple-template models vs. first-ranked template models; (b) multiple-template models vs. the best single-template models. A point under the diagonal line indicates that the multiple-template model is better.

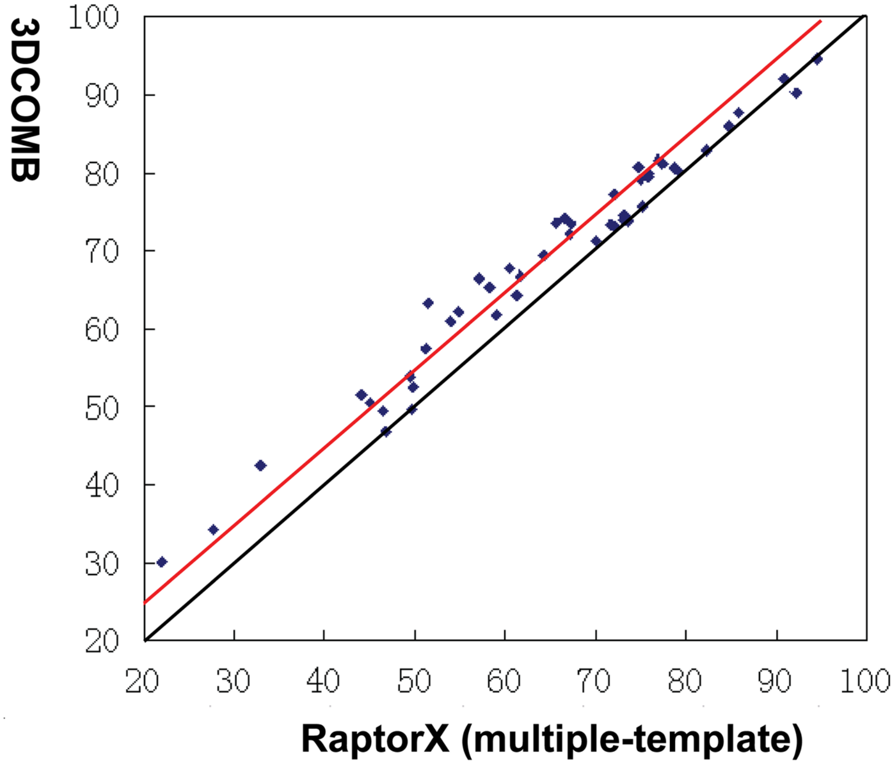

Figure 5.

This figure illustrates the quality of the models built from RaptorX multiple-template alignments and 3DCOMB alignments. A point above the red line indicates that the 3DCOMB model is at least 5 GDT units better than the RaptorX model.

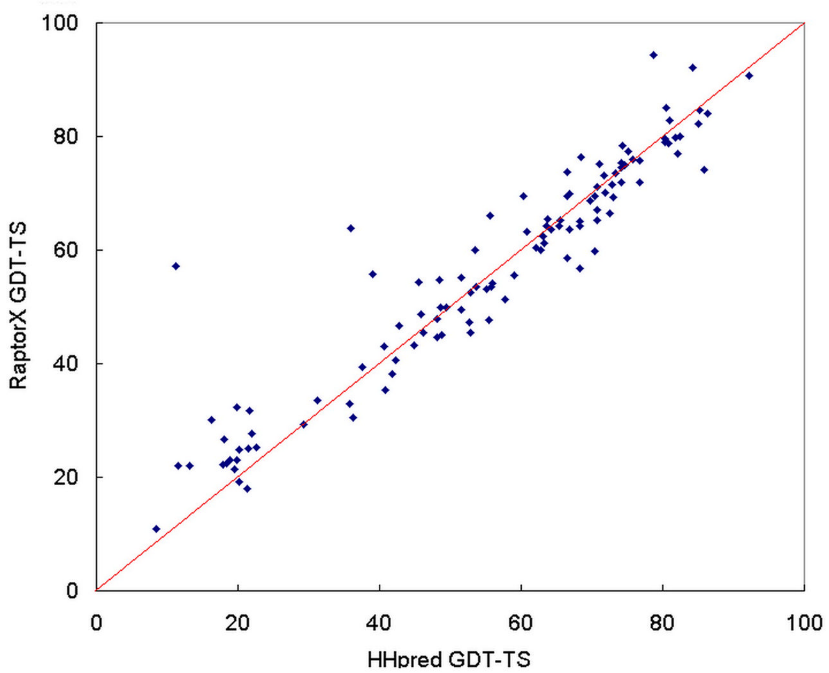

Figure 6.

This figure illustrates the quality of the models built by RaptorX and HHpred for the CASP9 targets. Each point represents one target. A point above the diagnoal line indicates that RaptorX generated a better 3D model than HHpred.

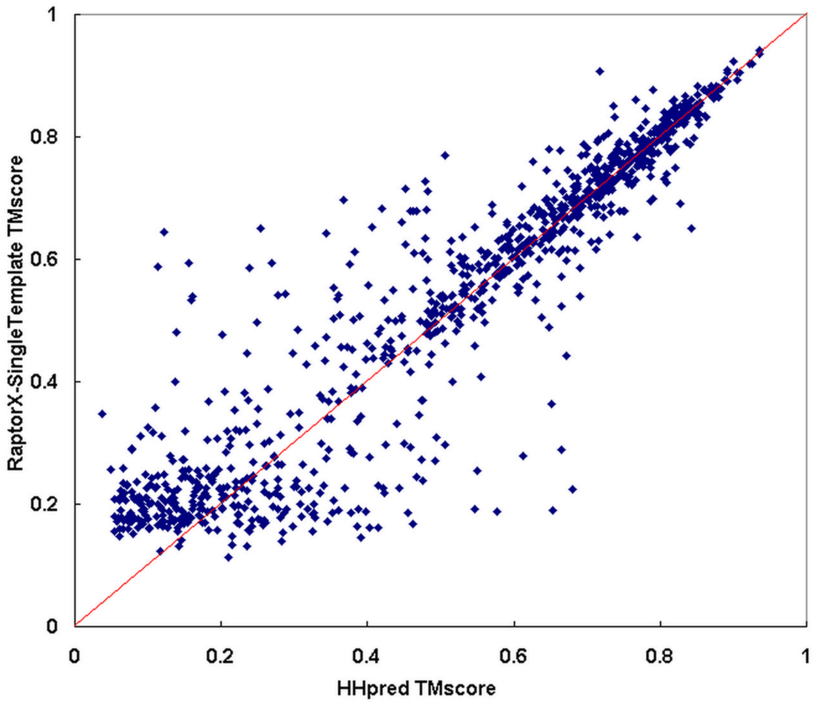

Figure 7.
This figure illustrates the quality of the models built by RaptorX and HHpred for 1000 proteins. Both methods use only single templates to generate alignments and build models (with MODELLER). The template set consists of more than 6000 non-redundant protein chains generated by the PISCES server while the target set includes 1000 proteins randomly-chosen from these 6000 proteins. Each point represents one target. A point above the diagnoal line indicates that RaptorX generated a better 3D model than HHpred.

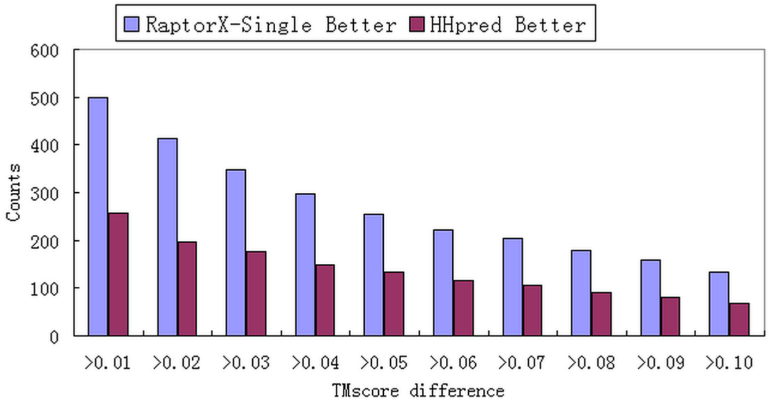

Figure 8.

This figure illustrates the TMscore difference distribution of the models generated by RaptorX and HHpred for 1000 proteins. For both methods we generated alignments and builded models (with MODELLER) using only single templates. The template set consists of more than 6000 non-redundant protein chains generated by the PISCES server while the target set includes 1000 proteins randomly-chosen from these 6000 proteins. The blue columns indicate that the numbers of targets for which RaptorX is better while the red columns indicate that HHpred is better.

Table 1

Quality (TMscore and GDT) of models generated for CASP targets from different alignments and templates.

Method	**48 CASP9 targets**		**99 CASP8&CASP9 targets**	
	TMscore	**GDT**	**TMscore**	**GDT**
Multi-template	35.4730	3068.50	75.686	6585.70
First-single-template [(a)]	34.0422	2909.37	72.8633	6265.68
Best-single-template [(a)]	34.7702	2980.76	74.0657	6381.54
Best-single-template [(b)]	35.2572	3046.95	74.8564	6465.12

[(a)] Alignment is generated by single-template threading;

[(b)] alignment is generated by multiple-template threading.