Applying Undertaker Cost Functions to Model Quality Assessment

John Archie and Kevin Karplus

September 5, 2008

Abstract

Undertaker is a program designed to help predict protein structure using alignments to proteins of known structure and fragment assembly. The program generates conformations and uses cost functions to select the best structures from among the generated conformations. This paper describes the use of Undertaker's cost functions for model quality assessment (MQA). We achieve an accuracy that is similar to other methods, without using consensus-based techniques. Adding consensus-based features further improves our approach substantially. We report several correlation measures, including a new weighted version of Kendall's τ (τ_3) and show MQA results superior to previously published results on all correlation measures when using only models with no missing atoms.

Introduction

Given the multitude of available methods for structure prediction and the increasing availability of computing resources, researchers can generate many models for a protein of interest with relative ease. As a result, the ability to select the best model from among multiple predictions has become an increasingly important problem.

The SAM web server uses the SAM hidden Markov model package to generate alignments and fragments. The server then runs undertaker to assemble alignments and fragments into initial 3D models. Finally, undertaker uses a genetic algorithm to generate structures, selecting the best structures at each iteration via a combined cost function [1]. However, undertaker also has the ability to read arbitrary models and output scores for the combined cost function. The combined cost function consists of the weighted sum of individual cost functions. We aim to assign weights to the individual cost functions such that the combined cost function can be applied to model quality assessment instead of structure prediction. All of Undertaker's cost functions use either evolutionary or physics-like terms. Consequently, we do not rely upon any consensus-based methods for the evaluation of model quality as do other high performing methods [2, 3, 4, 5].

Methods

Our approach was to define a measure of correlation for predicted quality versus actual quality that captures many of the properties desirable for model quality assessment and to optimize weights of the 73 individual Undertaker cost functions, maximizing correlation. This approach differs from the typical approach of directly predicting a model quality measure (such as GDT_TS [10] or GDT_HA [11]) using neural networks [4, 6], support vector regression [2], or a similar technique.

Weighted Kendall's τ

Kendall's τ is a nonparametric statistic to measure a monotonic relationship between two variables, with an easily interpreted meaning for model quality assessment: τ is directly related to the probability that, given a random pair of structures, the structure with the better cost is the better model ($\tau = 2p - 1$, where p is the probability).

We present a method of weighting Kendall's τ that is a special case of a weighting described elsewhere [7] but can still be computed in $\mathrm{O}(n \log n)$ [8]. If models are ranked by cost (with the predicted best model having a rank of 0), models with lower rank can be weighted more with

$$\mathrm{W}_{\alpha,i} = e^{-\alpha i/(n-1)} \tag{1}$$

where i is the rank of the decoy by cost, α is an arbitrary weighting parameter, and n is the total number of models. The definition of weighted τ or τ_α is then,

$$\tau_\alpha = 2\frac{\sum_i \mathrm{W}_{\alpha,i} \sum_{j \neq i} \mathrm{C}_{i,j}}{\sum_i \mathrm{W}_{\alpha,i}(n-1)} - 1 \tag{2}$$

where $C_{i,j}$ is 1 if the model with better cost (i or j) is actually superior, 0 if the model with better cost is inferior, and 0.5 if the models are tied in either cost or quality. If α is zero, this measure is equivalent to Kendall's τ, and as α approaches infinity, weighted τ becomes one less than twice the fraction of models of lower quality than the lowest-cost model.

Optimization

The following optimization techniques are designed to assign weights for the different components of Undertaker's cost function. Both techniques described below require a rebalancing step: If two sets of cost function components ($\mathbb{A}$ and $\mathbb{B}$) are to be combined, it is necessary to find a weighting parameter ($0 \leq p \leq 1$) specifying the proportion of the weight given to components in partition $\mathbb{A}$, where the remaining weight ($1-p$) is given to partition $\mathbb{B}$. To select p such that the average correlation is maximized, Brent's method [9] works well for τ_α and is used here. This step is denoted in the following algorithms as Rebalance($\mathbb{A}, \mathbb{B}$). After this optimal weight has been found, each cost function component in set $\mathbb{A}$ is scaled by p; each component in set $\mathbb{B}$, $1-p$.

The average correlation of a set of components, $\mathbb{C}$, with GDT_TS [10] or another quality measure is denoted as Cor($\mathbb{C}$).

Systematic optimization and greedy optimization (Figures 1 and 2) both use the same starting point for cost function weights. To give each cost function component equal initial influence, each weight in the set of components is assigned an initial weight of the inverse of the pooled standard deviation across all targets. Pooled standard deviation is defined by

$$\sigma_{\text{pooled}} = \sqrt{\frac{\sum_{t \in \mathbb{T}}(n_t - 1)\sigma_t^2}{\sum_{t \in \mathbb{T}}(n_t - 1)}} \tag{3}$$

where $\mathbb{T}$ is the set of targets, n_t is the number of structures for target t, and σ_t is the standard deviation of the cost function among models of target t.

```
repeat
    for $c \in \mathbb{C}$ do
        $\mathbb{A} \Leftarrow \{c\}$
        $\mathbb{B} \Leftarrow \mathbb{C} - \{c\}$
        Rebalance($\mathbb{A}, \mathbb{B}$)
        $\mathbb{C}_{\text{prev}} \Leftarrow \mathbb{C}$
        Update $\mathbb{C}$ with weights of $\mathbb{A}$ and $\mathbb{B}$
    end for
until Cor($\mathbb{C}$) − Cor($\mathbb{C}_{\text{prev}}$) $< T$
```

Figure 1: SystematicOpt($\mathbb{C}$). This algorithm tries to improve the average correlation by taking each component in turn and rebalancing it against the remaining components. T is the threshold for convergence. Initial weights for each cost function are the inverse of the pooled standard deviation of the corresponding cost function.

Real Cost

As an alternative to GDT_TS, we also use a different quality measure, real cost. Real cost is designed to avoid some of the undesirable characteristics of GDT_TS. Specifically, GDT_TS is imprecise when models are of very high or very low quality and is also insensitive to the correct prediction of H-bonds and side chain rotamers. The real cost measure is described in Table 3. Weights were assigned to the various components subjectively and by observing the relative scores of models to make sure that real cost was ranking models reasonably.

Weight	Term	Range	Description
50	real_hbond_u	[-1, 0]	negated proportion of H-bonds in real structure also present in decoy
50	real_hbond	[-1, 0]	like real_hbond_u; each H-bond weighted by its scaled likelihood
50	decoy_hbond_u	[-1, 0]	negated proportion of H-bonds in decoy also present in real structure
50	decoy_hbond	[-1, 0]	like decoy_hbond_u; each H-bond weighted by its scaled likelihood
10	real_NO_hbond_u	[-1, 0]	like real_hbond_u; only backbone H-bonds
10	real_NO_hbond	[-1, 0]	like real_hbond; only backbone H-bonds
10	decoy_NO_hbond_u	[-1, 0]	like decoy_hbond_u; only backbone H-bonds
10	decoy_NO_hbond	[-1, 0]	like decoy_hbond; only backbone H-bonds
200	clens	[0, 1]	a contact-based measure
35	log_rmsd	[-7, ∞]	log of the heavy atom RMSD
30	log_rmsd_ca	[-7, ∞]	log of C_α RMSD
1	GDT	[-100, 0]	negated GDT_TS score
1	smooth_GDT	[-100, 0]	like GDT_TS with many thresholds between 1/2 and 12 Å

Table 1: Real cost quality measure. Real cost is computed as a linear combination of terms in the table above. The likelihood of an H-bond is estimated by geometry using a method that is described in the supplementary materials. Including the proportion of H-bonds present in a real structure that are also present in the decoy and *vice versa* penalizes overprediction and underprediction. The RMSD terms are useful for evaluating high-accuracy models, and taking the log makes the terms less influential on poor models. The log_rmsd functions will not return values of under -7 to avoid large negative values when comparing a model to itself. The smooth GDT cost function is similar to GDT_TS, but uses a weighted average over a continuous range of distances (supplementary materials). Combining both GDT and RMSD measures is an attempt to avoid rewarding overprediction or underprediction.

Undertaker Cost Functions

A comprehensive description of all 73 Undertaker cost functions used here is beyond the scope of this paper. However, the most useful cost functions turn out to be the alignment-based constraint sets described in the companion paper to this one [12].

Neural-net predicted local structure alphabet cost functions were also extremely useful. The alphabet predictions were generated with the program predict-2nd [1, 13]. An amino acid sequence and multiple alignment including that sequence is given as input to predict-2nd, and for each position in the input sequence, predict-2nd will output an estimated probability vector over each letter in a local structure alphabet. Undertaker can read these predictions as well as models to be evaluated. Undertaker uses each model to generate a corresponding sequence over a local structure alphabet. Most of the cost functions that incorporate neural net predictions compute a cost for each model with

$$\text{cost} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\mathrm{P}_{\mathrm{nn}}(l_i)}{\mathrm{P}_{\mathrm{bg}}(l_i)} \qquad (4)$$

where N is the number of amino acids in the sequence, l_i is the letter in a local structure alphabet corresponding to residue i, $\mathrm{P}_{\mathrm{nn}}(l_i)$ is the

```
$\mathbb{P}_{\text{next}} = \{c\}$ such that $\text{Cor}(c \in \mathbb{C})$ is maximized
repeat
  $\mathbb{P} \Leftarrow \mathbb{P}_{\text{next}}$
  $\mathbb{C} \Leftarrow \mathbb{C} - \mathbb{P}$
  for $c \in \mathbb{C}$ do
    $\mathbb{A} \Leftarrow \{c\}$
    $\mathbb{B} \Leftarrow \mathbb{P}$
    $\text{Rebalance}(\mathbb{A}, \mathbb{B})$
    if $\text{Cor}(\mathbb{A} \cup \mathbb{B}) > \text{Cor}(\mathbb{P}_{\text{next}})$ then
      $\mathbb{P}_{\text{next}} \Leftarrow \mathbb{A} \cup \mathbb{B}$
    end if
  end for
  $\text{SystematicOpt}(\mathbb{P}_{\text{next}})$
until $\text{Cor}(\mathbb{P}_{\text{next}}) - \text{Cor}(\mathbb{P}) < T$
$\mathbb{C} \Leftarrow \mathbb{P}$
```

Figure 2: GreedyOpt($\mathbb{C}$). This algorithm seeks only to include those cost functions which are useful for model quality assessment, while ignoring the others. The result will be simpler if only a few cost function need to be included. The greedy optimization method seeks to identify individual components in $\mathbb{C}$ that can improve average correlation and to add them to a pool of useful components, $\mathbb{P}$, only if they can improve average correlation. Initial weights for each cost function are the inverse of the pooled standard deviation of the corresponding cost function.

neural-net predicted probability of letter l_i occurring at position i in the amino acid sequence, and $P_{\text{bg}}(l_i)$ is the background probability of l_i as estimated from a thinned version of the PDB.

Data Set and Evaluation

Our data set includes the 91 evaluated CASP7 targets with structures in the PDB. We only use this subset of targets because GDT_TS and real cost are computed locally, and computing these measures is not possible if there is no publicly available structure. We have tested our greedy optimization method with five-fold cross validation. Our model quality predictions for each CASP7 target were determined by a method trained on 4/5 of the data set that did not include the target for which quality predictions were being made.

The training data set for our method included all CASP7 predictions as well as the same predictions after they had been optimized with SCWRL 3.0 [14]. The testing data set included CASP7 server predictions only so that we could compare our performance to that of other methods using the same data set.

In each case we trained our method on data using τ_3 against GDT_TS or real cost with a threshold of 0.0001. The choice of τ_3 was subjective, but the measure provides a balance between general correlation and assigning more weight to the predicted best models. Setting $\alpha = 3$ gives just over half of the weight to the top quarter of models. Both τ_3 and the convergence threshold were chosen before performing cross-validation.

Results from Other Groups

To compare our results to those of other groups, we downloaded the CASP7 quality assessment data from the CASP7 website. In the case of the TASSER group [3], which was disqualified for not providing numerical scores in their submissions, we converted the order of models listed into ranks. Data for the Qiu group were downloaded from the online supplement for that work [2].

Results & Discussion

The optimized Undertaker combined cost function performs comparably to other existing state-of-the art methods for model quality assessment. (Table 2(a)). We perform competitively on all measures of correlation and do well at selecting the best model as shown by the average GDT_TS, although the difference between our group and the TASSER group is not statistically significant (one-tailed Wilcoxon signed rank test with paired data; P=0.782).

One of the interesting results in CASP7 was the excellent models produced by ZhangServer [15, 16]. The C_α trace was typically excellent when compared to other methods but less attention was given to the placement of other backbone atoms and side chain rotamers. As a result, the models look somewhat unique, and to ensure that our method was not just learning to recognize ZhangServer models, we evaluated on a set with those models removed. Even after removing the ZhangServer models from the testing data set, our performance is similar (Table 1 in supplement) and our method continues to select good models.

One simple method for obtaining a consensus score is to take a model and determine the TM-score [17] between that model and the first model submitted by each CASP server. (CASP structure prediction groups are allowed to submit five models for each target. The first model is the one that the group thinks is most likely to be correct.) The median TM-score is a powerful consensus signal [2]. By including the median TM-score as if it were another Undertaker cost function, we were able to substantially improve all measures reported in Table 2; for all reported measures of correlation the difference was statistically significant ($P \ll 0.0001$; one-tailed Wilcoxon signed rank test with paired data). The quality of the best selected model also improved, but the difference was not statistically significant in all cases ($P > 0.05$).

Furthermore, Undertaker is designed to optimize complete models, and many of the cost functions were not designed to handle missing atoms well. As an unintended result, some cost functions reward models for omitting atoms. If we train and evaluate only on complete models (Table 2(b)), performance relative to other methods improves, especially in regard to Pearson's r.

Additionally, the model quality measure used—GDT_TS or real cost—matters when looking at the cost functions chosen (Tables 3 and 4). Optimization was done on full models to ensure that cost functions which did not handle missing atoms well could be included. For real cost, the n_notor H-bond alphabet (Figure 9 in supplement) component was assigned a fairly large weight. The n_notor alphabet was also consistently given a high weight when doing five-fold cross validation, and was never selected when optimizing for GDT (supplementary materials). To a lesser degree, the Φ and Ψ torsion angles, as described by the Bystroff alphabet [18] were also selected when optimizing for real cost.

The most useful cost functions included alignment-based constraints [12] and neural-net predicted alphabets for burial [19], alpha torsion angle [20], and (for real cost) the n_notor H-bond alphabet (see supplementary materials).

Conclusions

The goal of most model quality assessment methods is to select the best model from a group of structures. In these terms, our method is competitive with the other methods examined. Nonetheless, among the best-performing quality assessment methods compared here, ours is the only one to exclusively use a nonconsensus approach, and consequently we do not require the generation of many structures in order to perform model quality assessment.

Consensus-based measures do help when you have a large fraction of good models in your set to evaluate, as in the CASP7 set we evaluated on. Combining our cost functions with the median TM-score improved results substantially relative to using undertaker cost functions alone.

Although we did maintain a clean division between CASP7 targets used for training and testing, other groups mentioned here did not have the benefit of training on any CASP7 data. We look forward to CASP8 results for a truly fair comparison of methods.

Finally, we have presented tools to better judge model quality assessment measures. While average GDT_TS of the predicted best model shows how well methods are doing at selecting the best model, it does not reveal the full picture. (What if the predicted best model was not present? Would the assessment method still have chosen a good model?) GDT_TS of the predicted best model tends to be more variable than measures of correlation because it relies mainly on one model, and it is also not useful for comparing relative performance among targets of different difficulty. The weighted τ_α provides a balance between simply taking the GDT_TS of the

(a)

Group	$\bar{r}$	$\bar{\rho}$	$\overline{\text{GDT}}$	$\bar{\tau}_0$	$\bar{\tau}_3$
under+TM	**0.90**	**0.85**	**61.2**	**0.70**	**0.68**
under	0.76	0.76	*60.5*	0.59	*0.60*
TASSER	0.64	0.70	60.3	0.55	0.53
Qiu	*0.85*	0.75	60.2	0.59	0.55
Pcons	0.82	0.75	58.6	0.56	0.52
LEE	0.82	*0.78*	58.1	*0.64*	0.59
ModFOLD	0.66	0.55	55.9	0.40	0.37

(b)

Group	$\bar{r}$	$\bar{\rho}$	$\overline{\text{GDT}}$	$\bar{\tau}_0$	$\bar{\tau}_3$
under+TM	**0.90**	**0.84**	**61.8**	**0.69**	**0.66**
under	*0.86*	*0.78*	*61.0*	*0.62*	*0.59*
Qiu	0.85	0.74	60.5	0.58	0.55
TASSER	0.63	0.69	60.4	0.54	0.52
LEE	0.80	0.72	58.4	0.58	0.53
Pcons	0.85	0.74	58.0	0.56	0.51
ModFOLD	0.70	0.62	57.0	0.46	0.44

(c)

Group	$\bar{r}$	$\bar{\rho}$	$\overline{\text{RC}}$	$\bar{\tau}_0$	$\bar{\tau}_3$
under+TM	**0.93**	**0.88**	**47.5**	**0.73**	**0.70**
under	*0.91*	*0.84*	*38.5*	*0.68*	*0.66*
Qiu	0.86	0.76	34.1	0.60	0.58
TASSER	0.70	0.74	30.7	0.58	0.56
LEE	0.80	0.72	14.5	0.57	0.52
Pcons	0.84	0.75	5.7	0.56	0.51
ModFOLD	0.76	0.69	14.9	0.52	0.50

Table 2: Performance of different model quality assessment methods. "under" denotes the Undertaker cost functions; "under+TM," the Undertaker cost functions with the median TM-score consensus term; and Qiu, data from a scoring function including the median TM-score consensus term, an atom-pairwise distance potential, and Rosetta terms [2]. TASSER, LEE, Pcons, and ModFOLD indicate CASP7 groups 125, 556, 634, and 704. The correlation measures are against negative GDT_TS (a,b) or real cost (c) and are averaged over 91 CASP7 targets. The metrics are Pearson's r, Spearman's ρ, average quality of predicted best model (GDT denotes GDT_TS; RC, real cost), Kendall's τ, and τ_3. Evaluation and training was done using five-fold cross-validation on all models and GDT_TS (a), complete models and GDT_TS (b); and complete models and real cost (c). Tables are sorted by the average quality of the best model. The largest value in each column is presented in bold; the second largest, italics.

Cost Function	Pooled SD	Description
align_constraint	3.854	selected alignment predicted constraints
pred_nb11_back	1.045	neural net predicted burial, near-backbone-11 alphabet
pred_alpha_back	0.843	neural net predicted alpha torsion angle
noncontacts_bonus	0.791	alignment predicted noncontacts
near_backbone	0.191	propensity predicted burial, near-backbone-11 definition
dry5	0.172	propensity predicted burial, dry-5 definition
rejected_bonus	0.168	rejected alignment predicted constraints
pred_o_sep_back	0.149	predicted H-bond sequence separation for O
contact	0.127	average number of contacts (centroids of the backbone and sidechain within 8 Å) per residue
pred_n_sep_back	0.117	predicted H-bond sequence separation for N
ehl2_constraint	0.105	secondary structure constraints
is_align	0.028	detects missing backbone atoms or chainbreaks

Table 3: Selected cost functions when optimizing against GDT_TS. Cost function weights were assigned to maximize τ_3 between the total cost and GDT_TS on models with no missing heavy atoms. Weights are reported as the pooled standard deviation of the cost function times the weight to avoid problems with the arbitrary scaling of individual cost functions. See the supplementary materials for more details of the cost functions.

best selected model and using a rank-based correlation metric which is relatively insensitive to outliers.

It is also possible to generalize Pearson's r and Spearman's ρ to include weighted observations—something not discussed here. Pearson's r has been criticized because it is sensitive to outliers and relies on linearity assumptions that are commonly untrue of quality assessment data [6]. For untied data, Kendall's τ may have slightly more power than Spearman's ρ, depending on certain assumptions regarding the underlying distribution and nature of outliers [21], but both measures are fairly robust to outliers and usually produce similar results. However, as argued in the companion paper to this one [12], ties should be penalized for quality assessment, and defining $\mathrm{C}_{i,j} = 0.5$ for ties in Equation 2 provides an elegant way to do just that. Furthermore, Kendall's τ is a very intuitive measure: For model quality assessment, τ is simply related to the probability that, given a random pair of structures, the structure with the better model quality score is actually the better model. Our definition of τ_α can be thought of in terms of a weighted probability.

We have shown that the selection of cost functions chosen by our greedy algorithm is sensitive to the choice of quality metric used. If we want to judge a method for predicting model quality, we need a measure of actual model quality that includes all of the features that are associated with good models, and while still not a perfect measure, real cost seems to capture more of these features than GDT_TS.

Finally, real cost has a drawback which, while not relevant for the current work, should be considered before adopting the measure for a different purpose. When there is a very large number of missing residues in a model (*e.g.* when only an α-helix is predicted), the log-RMSD and decoy H-bond terms can dominate and produce a score that is better than one might expect. In a CASP-like environment, where models with large number of missing residues need to be evaluated, an additional penalty term would need to be introduced to address this issue.

Cost Function	Pooled SD	Description
align_constraint	1.778	selected alignment predicted constraints
pred_nb11_back	1.217	neural net predicted burial, near-backbone-11 alphabet
pred_n_notor_back	0.892	neural net predicted H-bond properties, including NOtor torsion angle
pred_alpha_back	0.837	neural net predicted alpha torsion angle
align_bonus	0.755	selected alignment predicted constraints
noncontacts_bonus	0.728	alignment predicted noncontacts
rejected_bonus	0.263	rejected alignment predicted constraints
pred_bys_back	0.251	neural net predicted bystroff alphabet
sidechain	0.202	the negative log-probability of observing the sidechain and backbone conformation
dry5	0.140	propensity predicted burial, dry-5 definition
near_backbone	0.124	propensity predicted burial, near-backbone-11 definition
contact	0.121	average number of contacts (centroids of the backbone and sidechain within 8 Å) per residue
alpha_prev	0.078	propensity predicted alpha angle of the previous residue
hbond_count	0.073	the number of H-bonds normalized by chain length
pred_n_sep_back	0.047	predicted H-bond sequence separation for N
is_align	0.042	detects missing backbone atoms or chainbreaks

Table 4: Selected cost functions when optimizing on real cost. Cost function weights were assigned to maximize τ_3 between the total cost and real cost on models with no missing heavy atoms. Weights are reported as the pooled standard deviation of the cost function times the weight to avoid problems with the arbitrary scaling of individual cost functions. See the supplementary materials for more details of the cost functions.

Acknowledgments

We would like to acknowledge all of the people who have worked on Undertaker. Specifically, Martin Paluszewski created the alignment-based constraints; Rachel Karchin worked extensively with burial and backbone alphabets; Grant Thiltgen created the predicted H-bond alphabets and cost functions; and George Shackelford implemented the neural net predicted constraints.

This research was supported by NIH grants R01GM068570 and T32GM070386.

References

[1] Karplus K, Karchin R, Draper J, Casper J, Mandel-Gutfreund Y, Diekhans, M, Hughey R. Combining local-structure, fold-recognition, and new-fold methods for protein structure prediction. Proteins 2003; 53:491–496.

[2] Qiu J, Sheffler W, Baker D, Noble WS. Ranking predicted protein structures with support vector regression. Proteins 2008; 71:1175–1182.

[3] Zhou H, Skolnick, J. Protein model quality assessment prediction by combining fragment comparisons and a consensus C(alpha) contact potential. Proteins 2007; 71:1211–1218.

[4] Wallner B, Elofsson A. Prediction of global and local model quality in CASP7 using Pcons and ProQ. Proteins 2007; 69:184–193.

[5] Cozzetto D, Kryshtafovych A, Ceriani M, and Tramontano A. Assessment of predictions in the model quality assessment category. Proteins 2007; 69:175–183.

[6] McGuffin LJ. Benchmarking consensus model quality assessment for protein fold recognition. BMC Bioinformatics 2007; 8:345.

[7] Shieh GS. A weighted Kendall's tau statistic. Stat Probab Lett 1998; 39:17–24.

[8] Knight WR. A computer method for calculating Kendall's tau with ungrouped data. J Am Stat Assoc 1966; 61:436–439.

[9] Press WH, Teukolsky SA, Vetterling WT. 1992. Numerical Recipes in C. 2nd ed. New York: Cambridge Univ Pr. p 402–405.

[10] Zemla A. LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Res 2003; 61:3370–3374.

[11] Kryshtafovych A, Prlic A, Dmytriv Z, Daniluk P, Milostan M, Eyrich V, Hubbard T, Fidelis K. New tools and expanded data analysis capabilities at the protein structure prediction center. Proteins 2007; 69(S8):68–82.

[12] Paluszewski M, Karplus K. Model Quality Assessment using Distance Constraints from Alignments. Proteins 2008. Accepted.

[13] Katzman S, Barrett C, Thiltgen G, Karchin R. Predict-2nd: a tool for generalized local structure prediction. Bioinformatics 2008. Accepted.

[14] Canutescu AA, Shelenkov AA, Dunbrack RL. A graph-theory algorithm for rapid protein side-chain prediction. Protein Sci 2003; 12:2001–2014.

[15] Battey JND, Kopp J, Bordoli L, Read RJ, Clarke ND, Schwede T. Automated server predictions in CASP7. Proteins 2007; 69(S8):68–82.

[16] Zhang Y. Template-based modeling and free modeling by I-TASSER in CASP7. Proteins 2007; 69(Suppl 8):108–17.

[17] Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins 2004; 57:702-710.

[18] Bystroff C, Vesteinn T, Baker D. HMM-STR: a Hidden Markov Model for Local Sequence-Structure Correlations in Proteins. J Mol Biol 2000;301:173–190.

[19] Karchin R, Cline M, Karplus, K. Evaluation of local structure alphabets based on residue burial. Proteins 2004; 55:508–518.

[20] Karchin R, Cline M, Mandel-Gutfreund Y, Kevin, K. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins 2003; 51:504–514.

[21] Gideon RA and Hollister RA. A rank correlation coefficient resistant to outliers. JASA 1987; 82:656–666.