Approximating Conserved Regions of Protein Structures

Yi Wei, Mingfu Shao, Jishuang Yang, Chao Wang, Shuai Cheng Li, Dongbo Bu*

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China
**Email: dbu@ict.ac.cn*

Motivation: We present in this paper an efficient algorithm to identify conserved regions from multiple protein structures. The Critical Assessment of Techniques for Protein Structure Prediction [1] experience suggests that for a given target sequence, threading methods usually generate several structures (called decoys or models) with conserved regions similar to the native structure, and identification of conserved regions can help improve structure prediction [16]. Thus, it is important to efficiently detect conserved regions without requirement of alignment information.

Results: Based on our previous work on approximating the bottleneck distance [11], we present in this paper an $O(m^2n^2 \log n)$ time algorithm to identify the maximum conserved regions from m decoys, where n is the number of amino acids per decoy.

To measure the quality of the identified conserved regions, we performed two experiments: the first one directly compare the identified conserved regions against native structures; the other experiment serves as an indirect measure of conserved regions; that is, it aims to investigate whether these conserved regions help improve protein structure prediction.

Experimental results demonstrate that for 16 out of 25 TBM (template-based modeling) targets in CASP7, our method can identify over 70% native-like regions and filter out over 90% of non-native-like regions, simultaneously. The algorithm also performs well for 10 out of 12 FM (free modeling) targets in CASP7, where we obtain more than half of native-like regions and filter out over 80% non-native-like regions.

Furthermore, we applied the identified conserved regions to improve fragment-assembly-based approaches to protein structure prediction. We observed that for 10 out of 12 FM targets in CASP7, our method shows higher accuracies than ROSETTA [22]. In particular, by identifying conserved regions, TM-score [37] are improved significantly from meaningless (< 0.4) to meaningful (> 0.4) on four targets. This experiment provides with an indirect evidence of the performance of our algorithm to identify conserved regions.

1. Introduction

The methods to attack protein structure prediction problem can be divided into three families: homology modeling [20, 2], threading [35, 24, 31], and *ab initio* methods [15, 22]. Homology modeling and threading methods usually generate accurate predictions if query-template sequence identity is over 30% [18]. In contrast, *ab initio* methods were designed for the case that significantly sequence similar templates are unavailable. The most successful *ab initio* strategy is fragment assembly [15, 22], where short structure fragments are used to build a native-like structure. Typically, an *ab initio* method consists of two steps: predicting local structures for each 9-mer sequence segment of the query sequence first, and then constructing a decoy by choosing building blocks from these local structure candidates.

Identifying conserved regions can help reduce the search space of the fragment-assembly-based *ab initio* methods. For example, ROSETTA, a typical fragment-assembly-based method, has a search space with a size of $O(25^n)$ since for each 9-mer sequence segment, 25 local structures are predicted as candidates (n is the sequence length). However, analysis suggested that the possible conformation space can be approximated by $O(1.6^n)$ [21, 12], implying the possibility to reduce ROSETTA's search space. In fact, after identifying conserved regions, ROSETTA's search space can be significantly narrowed down by sampling non-conserved regions [16, 33].

In addition, identifying conserved regions from a set of predicted decoys is also important to improve protein structure prediction. For example, it has been reported that though the decoys generated by *ab initio* method are globally inaccurate, the conserved regions shared by these decoys are usually similar to the corresponding regions in the native conformation [16]. Similar observation were obtained

*Corresponding author.

for decoys generated by threading methods. Specifically, taking consensus decoy from threading results was proved to be an efficient method to identify better decoys. For example, 3D-Jury [38] employs two scoring functions to measure the global quality of decoys; 3D-Shotgun [39] depends on pair-wise structural alignment to detect conserved regions.

It should be noted that most of previous studies focused on the identification of conserved regions with helps of alignment information. For example, Julie *et al.* defined *Core Blocks* based on secondary structure and residue conservation [9]; Yamada *et al.* proposed a method that utilizes alignments generated from distances of dihedral angles [32]; Tatusov *et al.* presented a method based on sequence alignment blocks [25]; and Krishnan *et al.* applied wavelet technique to identify conserved regions from multiple alignments [10]. All these alignment-based conserved regions were designed to categorize proteins into families. Contrastly, in this paper, the conserved region refers to the similar local structures shared by multiple decoys, and we aim to identify conserved regions without the requirement of alignment information. In addition, we aim to identify conserved local regions rather than consensus global structure as 3D-Jury did.

The subject of this paper is to design an efficient method to identify conserved regions without requirement of alignment information. Specifically, we first formulated conserved regions identification problem as an optimization problem, i.e., MAXIMUM CONSERVED REGION problem, and then proposed an approximation algorithm to solve it.

To measure the quality of conserved regions, we performed two experiments: the first experiment directly compares the identified conserved regions against native structures; the second experiment serves as an indirect measure of the identified conserved regions by investigating whether these conserved regions help improve protein structure prediction.

Experimental results demonstrate that the identified conserved regions are similar to their counterpart regions in the native structure, and can help improve the quality of the final generated structures.

2. Methods

2.1. Problem Statement

Given m protein structure models (decoys) for a protein of n residues, each model can be represented as n ordered three dimensional (3D) points $P_i = (p_{i,1}, ..., p_{i,n})$, $1 \leq i \leq m$, $p_{i,j} \in \mathbb{R}^3$ (here we use Cα atom to represent a residue). Let us denote the decoy set as $P = \{P_1, ..., P_m\}$. A fragment of a structure P_i is a sequence of consecutive points, denoted as $P_i[a, b] = (p_{i,a}, p_{i,a+1}, ..., p_{i,b})$. Given two fragments $P_i[a, b]$ and $P_j[a, b]$, the distance between them is defined as:

$$d(P_i[a,b], P_j[a,b]) = \min_{T} \max_{a \leq k \leq b} ||p_{i,k} - T(p_{j,k})||$$

where $||\bullet||$ denotes the Euclidean distance, and T denotes a rigid transformation that includes a rotation along with a translation. Here, we adopt *bottleneck distance* [11] to measure the distance between two structural fragments. Briefly, the distance between two fragments is defined as the maximum distance between corresponding points under the optimal rigid transformation to superimpose them.

Given a decoy $P_i \in P$, the region $P_i[a, b]$ is conserved if it is shared by more than ωm decoys in the corresponding regions, where ω is a constant with $0 < \omega \leq 1$. Mathematically, given a distance threshold θ, $P_i[a, b]$ is conserved if we can find a subset $P' \subset P$, $|P'| >= \omega m$, such that $d(P_i[a, b], P_j[a, b]) \leq \theta, \forall P_j \in P'$.

We say an interval $[a, b]$ is a *conserved region* if there exists at least one decoy $P_i \in P$ such that $P_i[a, b]$ is conserved. Obviously, any sub-interval of a conserved region is still a conserved region. A conserved region is denoted as *maximal conserved region* if it cannot be extended any more. It is easy to see that maximal conserved regions may overlap but a maximal conserved region cannot contain any other maximal conserved regions. We use *maximum conserved regions* to refer to a series of conserved regions that the total length of these regions is maximized. For the sake of simplicity, we denote the region boundaries as $L([a, b]) = a$ and $R([a, b]) = b$.

The problem of identifying maximum conserved regions can be defined as follows:

MAXIMUM CONSERVED REGION PROBLEM.

Given distance threshold θ, consensus factor ω, length threshold λ, and an integer k, the goal is to identify k disjoint regions $[a_1, b_1]$, $[a_2, b_2]$ ... $[a_k, b_k]$ such that the total length of these regions is maximized, i.e.,

- $[a_l, b_l]$ is a conserved region shared by at least ωm models, $|b_l - a_l| \geq \lambda$, $1 \leq l \leq k$
- $\sum_{l=1}^{k}(b_l - a_l + 1)$ is maximized.

In general, the distance threshold θ is small, say, $\theta \leq 2\mathring{A}$, while length threshold λ is a constant, say, $\lambda \geq 9$. Therefore, some structures might not contain any conserved regions. We call these structures random decoys.

It should be noted that our problem is different from both the LCP (Largest Common Point) problem [6] and the LWPS (Largest Well-Predicted Subset) problem [19, 11]. Both LCP problem and LWPS problem apply the same transformation between two point sets while we use different transformations for different region. In addition, the conserved regions are required to be consecutive.

2.2. Identifying Conserved Regions

The intuition of our algorithm is as follows: We first identify all maximal conserved regions by using an approximation algorithm to calculate the bottleneck distance. Due to the property of maximal conserved region, we can obtain a conserved region by simply extracting a sub-interval from any maximal conserved region, which save efforts in the subsequent steps. Next we use dynamic programming technique to calculate the maximum length of conserved regions in time $O(m^2n^2 \log n)$. In addition, we also put forward some pruning rules to speed up in practice.

It has been proved that the bottleneck distance between two structures can be computed in polynomial time $O(n^7)$ [11]. However, this algorithm is too slow for the application of identifying conserved regions. In our previous work [11], we have proposed an approximation algorithm to tackle this problem. Briefly speaking, we first proved the existence of the approximation transformation, and then proposed a construction approach by using the techniques of discretization and enumeration. Since the focus of this paper is to identify conserved regions, the proofs of the existence and construction of a rigid transformation are omitted. Please refer to [11] for strict proofs.

2.2.1. *Approximating the Maximal Conserved Regions*

In this subsection, we propose a method to calculate the maximal conserved regions by using the efficient bottleneck distance approximation technique. Suppose the decoy set is P, $|P| = m$, and $P_i \in P$ is the center structure. We aim to compute the distance between $P_i[a, b]$ and the corresponding regions of the rest decoys to decide whether $P_i[a, b]$ is conserved or not. The above-mentioned approximation technique enable us to judge whether a region $[a, b]$ is conserved or not in time $O(m|b-a|)$ when a center structure is given. Thus, by trying each structure as the center structure, we can determine whether region $[a, b]$ is conserved or not in time $O(m^2|b - a|)$.

As mentioned in the previous section, maximal conserved regions may overlap but no maximal conserved region contains any other maximal conserved regions. Therefore, there are at most $O(n)$ maximal conserved regions since there is at most one maximal conserved region starting at i, $1 \leq i < n$. Thus, the remaining difficulty is to determine the right endpoints of the maximal conserved regions.

We used a binary search to identify the end of the maximal conserved region for each i, $1 \leq i < n$. In total, we need $O(n \log n)$ searches to obtain the set of maximal conserved regions. Therefore, the total time to identify all the maximal conserved regions is $O(m^2n^2 \log n)$.

2.2.2. *Dynamic Programming Algorithm to Compute Maximum Conserved Regions*

We assume $\lambda = 1$ in this subsection. Our approach can be easily extended to the cases when $\lambda \neq 1$.

We use dynamic programming technique to find the maximum length of disjoint conserved regions. The dynamic programming table has two dimensions, denoted as $M[i, l]$, $1 \leq i \leq n$, $0 \leq l \leq k$. The entry $M[i, l]$ records the maximum length of no more than l conserved regions contained in the interval $[1, i]$, $1 \leq l \leq k$. Let $\mathcal{I}$ denote the set of maximal

conserved regions. Then the table can be built by the following recursive formulation.

$$M[i,l] = \max_{\substack{i,j \in I, I \in \mathcal{I} \\ i-j>\lambda}} \begin{cases} M[j-1, l-1] + |i-j+1| \\ M[i, l-1] \\ M[i-1, l] \end{cases} \quad (1)$$

Here, $M[n,k]$ consists of kn entries, and each entry can be computed in $O(n^2)$ time. Therefore, this dynamic programming algorithm runs in $O(kn^3)$ time. We can further improve the algorithm based on the following observation.

Claim 2.1. *For the maximum conserved regions problem, there exists an optimal solution $I_1, I_2, \cdots, I_k$, and for each $I_i, 1 \leq i \leq k$, we have a maximal conserved region $I_i' \in \mathcal{I}$ such that $R(I_i) = R(I_i')$.*

Proof.

The proof is by contradiction. Suppose in an optimal solution S there is a conserved region I_i such that I_i is neither a maximal conserved region nor shares the right endpoint with any maximal conserved region. I_i should be covered by at least one maximal conserved region since it is itself not a maximal conserved region.

We break the proof into two cases.

Case 1: $R(I_i) = L(I_{i+1}) - 1$

In this case, we can extend I_i by increasing $R(I_i)$ and shrink I_{i+1} by increasing $L(I_{i+1})$ simultaneously. This process is repeated until $R(I_i)$ is equal to the right endpoint of a maximal conserved region. It is obvious that both I_i and I_{i+1} are still conserved regions in this process, and the new maximum conserved regions length does not change.

Case 2: $R(I_i) < L(I_{i+1}) - 1$

In this case, we can extend I_i by increasing $R(I_i)$ until $R(I_i)$ is equal to the right endpoint of a maximal conserved region. We can also set $L(I_{i+1})$ to be $R(I_i) + 1$ if necessary. It is not difficult to see that the new I_i is still conserved, and the conserved regions length will increase. This contradicts the assumption of maximum conserved regions. Thus our claim holds. □

This claim implies that for each entry in equation 1, we just need to check only maximal conserved regions rather than all possible intervals. Therefore, the running time of the dynamic programming algorithm can be improved from $O(kn^3)$ to $O(kn^2)$, and the whole conserved regions identification algorithm runs in $O(m^2n^2 \log n + kn^2)$ time.

2.2.3. *Speed-up Technique*

An $O(m^2n^2 \log n + kn^2)$ time algorithm is still too slow for practical applications; therefore, we propose the following pruning rules to speed up:

- Before identifying maximal conserved regions, we evaluate all the regions of length λ first, and then decide whether it is necessary to search for longer intervals or not. The reason is that if $[a,b]$ is a conserved region, all of its sub-intervals are still conserved; otherwise, none of its super-intervals is a conserved region.
- We used RMSD (Root Mean Square Distance) [3] to filter out the non-conserved regions based on the fact that $RMSD(A,B) \leq d(A,B)$ [11]. We can further skip some RMSD calculations by using the triangular inequality of RMSD.

  Generally speaking, the calculation of RMSD is faster than that of bottleneck distance; hence these pruning rules can make our algorithm much faster in practice.

3. Results and Discussions

To evaluate our algorithm, we conducted two experiments: The first experiment aims to directly measure the quality of identified conserved regions, i.e., the similarity between the identified conserved regions and the corresponding parts of the native structure; the second experiment serves as an indirect measure; that is, it aims to investigate whether or not the identification of conserved regions can help improve protein structure prediction from threading results.

3.1. Evaluation of Conserved Region Quality

Due to the dependency on alignment information of the previous studies [9, 25, 10, 32], it is unfair to compare our method with these methods directly.

We either cannot make direct comparison with 3D-Jury since 3D-Jury reports a consensus global structure rather than conserved regions. Here, we performed comparision of conserved regions against native structure directly. In particular, we calculate the distance between these regions and the corresponding regions in the native structure.

Notice that it is also unfair to use the same distance criteria for conserved regions with variable lengths. To overcome this difficulty, we adopted the following criteria: we cut the conserved regions into fragments with k residues, and investigated the quality of these k-mer fragments. A fragment is called native-like if its distance from the corresponding native region is less than a threshold θ. This provides a fair way to compare the quality of conserved regions with variable length. In our experiments, we focused on 10-mer fragments, and set the threshold θ to be $2\mathring{A}$.

We selected 37 hard CASP7 targets with less than 200 amino acids. The targets are listed in Table 1 and 1. Specifically, among these 37 targets, 25 are considered as template based modeling targets (TBM), and 12 are considered as free modeling (FM) targets. For each target, we first run two kinds of threading methods, HHpred [24] and SP3 [35], and obtained 10 decoys with each method. Next we used our method to identify conserved regions from these models, cut these conserved regions into k-mer fragments, and calculated the distance between k-mer fragments and their corresponding native regions.

The performance of conserved fragments for the 25 TBM targets is shown in Table 1. In this table, the sensitivity column describes how many native-like regions we can detect and the specificity column describes how many non-native-like regions we can filter out. This table suggests that for 16 out of the 25 template based targets, about 70% of native-like regions are identified as conserved regions, and over 90% of non-native-like regions are filtered out. In general, for all of these targets, we can filter out over 80% regions that are significantly different from the native structures.

We also notice that for 4 targets, i.e., $T0306$, $T0342$, $T0358$ and $T0383$, the sensitivities are below 36%. By investigating the input decoys carefully, we identified two reasons: First, the TMscore of these decoys are less than 0.2, which means a low accuracy for these decoys. Generally speaking, a decoy with a TMscore less than 0.17 can be treated as a random prediction [37]. It is difficult to identify conserved regions from these nearly random predictions. Second, the effective of our methods will decrease significantly if few templates are correct and the majority have same errors.

In summary, the experimental results demonstrate that when the native-like regions occupy more than 20% positions of the input decoys, we can identify most of the native-like regions correctly and exclude most of the non-native-like regions for TBM targets.

We also obtained similar observations on the 12 free-modeling targets. As shown in Table 1, more than half native-like regions can be detected for 10 out of 12 targets, and over 80% of non-native-like regions can be excluded. These observations suggest that our method can help distinguish native-like regions from non-native-like regions.

3.2. Applying Conserved Regions to Improve Protein Structure Prediction

An alternative and indirect way to evaluate our algorithm is to investigate whether or not the identified conserved regions help improve protein structure prediction. Specifically, if our algorithm can filter out non-native-like regions, then protein structure prediction can be improved by applying the conserved regions technique iteratively. In brief, we identify conserved regions from decoys first, then we can make use of these partial "good" sub-structures by running ROSETTA with these sub-structures unchanged. In other words, we focus on the non-native regions only and thus narrow down the search space. By employing iteration strategy, we can obtain more and more good partial "good" sub-structures until convergence. Therefore, the quality of the convergence decoys is a good indirect measure of our algorithm.

The iterative idea was initiated at [12], where the distribution of torsion angles can be estimated more and more accurately as the iteration proceeds. Here, we attempt to employ this framework to evaluate our conserved regions algorithm.

Table 1. Performance of Conserved Fragments for 25 CASP7 Template-based-modeling(TBM) Targets and 12 Free-modeling(FM) Targets. TP: the number of true positive; FN: the number of false negative; FP: the number of false positive; TN: the number of true negative; SEN: Sensitivity; SPE: Specificity.

Type	Target	TP	FN	FP	TN	SEN	SPE
TBM	T0283	294	107	80	1044	0.73	0.92
	T0306	19	43	32	721	0.30	0.95
	T0312	170	152	67	751	0.52	0.91
	T0322	596	198	79	350	0.75	0.81
	T0327	298	44	22	527	0.87	0.95
	T0331	316	111	47	525	0.74	0.91
	T0335	208	29	15	295	0.87	0.95
	T0342	58	127	14	958	0.31	0.98
	T0349	269	101	9	494	0.72	0.98
	T0351	85	33	35	445	0.72	0.92
	T0354	225	77	44	610	0.74	0.93
	T0357	246	194	184	1064	0.56	0.85
	T0358	62	108	68	544	0.36	0.88
	T0360	263	165	119	562	0.61	0.82
	T0362	1066	172	11	326	0.86	0.96
	T0363	222	187	62	618	0.54	0.90
	T0364	1029	178	32	288	0.85	0.90
	T0368	869	56	21	436	0.93	0.95
	T0369	499	167	81	919	0.74	0.91
	T0370	893	298	86	850	0.74	0.90
	T0373	902	257	48	436	0.77	0.90
	T0374	859	303	60	623	0.73	0.91
	T0380	469	178	37	438	0.72	0.92
	T0383	217	536	55	617	0.29	0.92
	T0385	1317	234	1	451	0.84	0.99
FM	T0287	164	88	69	452	0.65	0.86
	T0300	144	85	18	201	0.62	0.91
	T0304	36	30	43	830	0.54	0.95
	T0307	349	167	51	1172	0.67	0.95
	T0309	28	87	8	377	0.24	0.97
	T0314	65	51	58	882	0.56	0.93
	T0319	110	100	118	1278	0.52	0.91
	T0348	39	119	9	282	0.24	0.96
	T0350	160	135	39	1096	0.54	0.96
	T0353	127	112	17	778	0.53	0.97
	T0361	335	174	123	884	0.65	0.87
	T0382	149	95	58	679	0.61	0.92

The iteration idea is described in Figure 3.2. Specifically, the first step is to identify conserved regions from 10 decoys generated by HHpred [24] and 10 decoys by SP3 [35]. In this step, we determined the maximum length of conserved regions from these decoys. Then we re-generated 1,000 decoys by running ROSETTA with the conserved regions fixed. Next we applied our identification method again to detect conserved regions from the newly generated 1,000 decoys. Thus this "identification-and-prediction" cycle was repeated iteratively until the generated decoys converge. In practice, this iteration process generally ends in less than three rounds of iterations. Finally we adopted a clustering method to choose one decoy from the decoys generated in the final round.

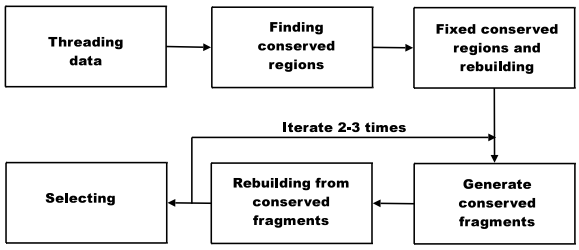

Fig. 1. Iteration Strategy to Improve Protein

We applied this iteration strategy to make blind structure predictions for the 37 CASP7 targets. Experimental results illustrate that compared with ROSETTA, our method helps generate more accurate models for 10 out of 12 FM targets. Due to the space limit, only four of these FM targets, $T0300$, $T0348$, $T0350$ and $T0361$, are shown in Figure 3. Take $T0300$ as an example. We observe that after two rounds of iterations, our method yields 68.1% good decoys, and this ratio increases to 98.7% after three rounds of iterations. In contrast, ROSETTA generates only 28.9% good decoys. Here, a decoy is good if its TMscore is over 0.4 [37]. In addition, the selected decoy is also better than that reported by ROSETTA.

We observe similar improvements for TBM targets. Specifically, in the first step, the conserved regions cover over 60% positions. The decoys are improved step by step and the final decoys are better than the threading results for most targets. Take $T0283$ as a concrete example. All current threading methods fail to find any significantly similar templates for this target. In contrast, after three iterations of our method (See Figure 5), the TMscore is improved to be over 0.4. This result suggests a clear advantage of our method relative to threading methods.

In the experiments, we also find that the decoys cannot be further improved after more itera-

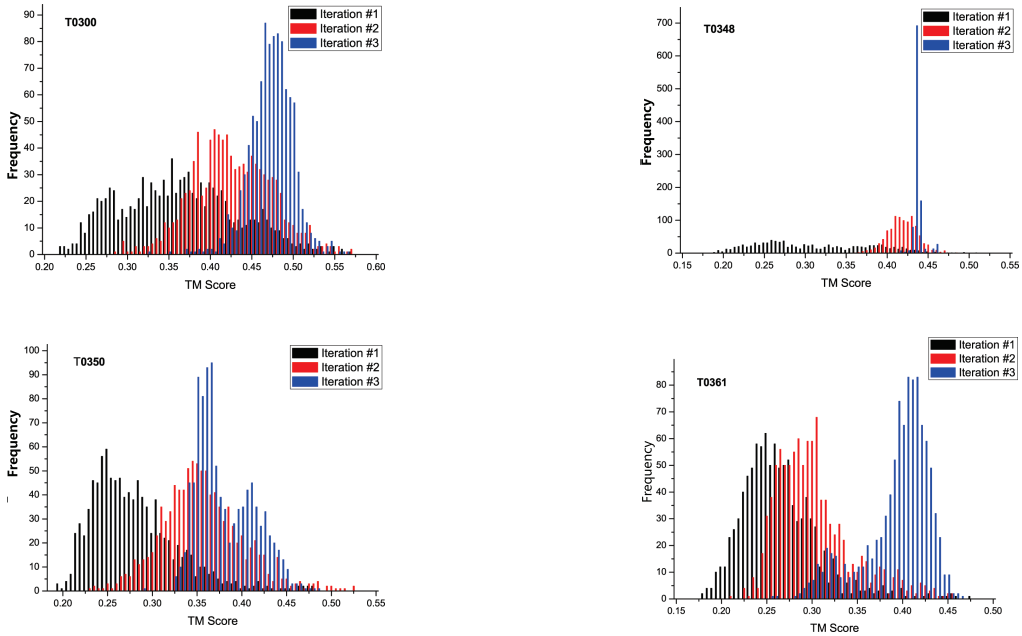

Fig. 3. The performance of Iteration Strategy for T0300,T0348,T0350,T0361. X-axis denotes TMscore, and Y-axis denotes the number of models having this score. In each iteration we generate 1000 decoys.

tion steps. One possible reason is that although the conserved regions are native-like, there still exist significant differences for the loop regions. The low accurency of loop regions are confirmed by a careful examination of ROSETTA's fragments.

The CASP7 results demonstrate that relative to TBM targets, FM targets are generally more difficult to predict [1]. Our experimental results regarding conserved regions confirm this observation: For most TBM targets, the conserved regions cover more than 60% positions, while for FM targets, the ratio is just around 30%. This also implies a large improvement space of prediction methods for FM targets.

In summary, the experimental results lead to a conclusion that the iteration strategy can greatly reduce the search space since more and more regions become native-like as the iteration proceeds. In contrast, the commonly-used Monte Carlo method keeps an unchanged search space in the search process. Generally speaking, a smaller search space usually implies a higher possibility to obtain the native-like structure, showing an advantage of our method in theory.

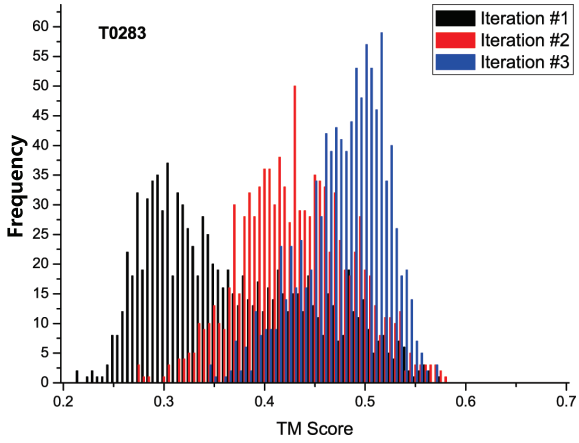

Fig. 5. The performance of Iteration Strategy for target T0283. In each iteration we generate 1000 decoys. X-axis denotes TMscore, and Y-axis denotes the number of models having this score.

4. Conclusion

In this paper, we present an algorithm to detect the maximum conserved regions from m decoys in $O(m^2n^2 \log n)$ time. Experimental results demonstrate that our method can significantly distinguish native-like regions from noisy regions.

Applying conserved regions in domain parsing and other topics will be in our future works.

References

1. 7th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction. *http://predictioncenter.org/casp7/ Casp7.html*.
2. Altschul, S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25**, 3389-3402.
3. Arun, K.S. *et al.* (1987) Least-squares Fitting of Two 3-d Point Sets. *IEEE Trans. Pattern Anal. Mach Intell*, **9**, 698-700.
4. Bonneau, R. *et al.* (2001) Rosetta in CASP4: Progress in Ab Initio Protein Structure Prediction. *Proteins: Structure, Function, and Genetics*, **Suppl**, 119-126.
5. Bradley, P. *et al.* (2003) Rosetta Predictions in CASP5: Successes, Failures, and Prospects for Complete Automation. *Proteins*, **53**, 457-468.
6. Christoph Ambuhl *et al.* (2000) Computing Largest Common Point Sets under Approximate Congruence. *ESA '00: Proceedings of the 8th Annual European Symposium on Algorithms*, Springer-Verlag, 52-63.
7. Felts, A.K. *et al.* (2002) Distinguishing Native Conformations of Proteins from Decoys with An Effective Free Energy Estimator Based on the OPLS All-atom Force Field and the Surface Generalized born Solvent Model. *Proteins*, **48**, 404-422.
8. Holmes, J.B. and Tsai, J. (2004) Some Fundamental Aspects of Building Protein Structures from Fragment Libraries. *Protein Science*, **13**, 1636-1650.
9. Julie D. Thompson *et al.* (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**, 87-88.
10. Krishnan, A. *et al.* (2004) Rapid detection of conserved regions in protein sequences using wavelets. *In Silico Biology*, **4**, 133-148.
11. Li, S.C. *et al.* (2008) Finding Largest Well-predicted Subset of Protein Structure Models. *19th Annual Symposium on Combinatorial Pattern Matching*, **5029**, 44-55.
12. Li, S.C. *et al.* (2008) FALCON: A New Position-specific HMM for Protein Structure Prediction. *Protein Science*, **17**, 1925-1934.
13. Li, S.C. *et al.*, (2008) Designing Succinct Structural Alphabets. *Bioinformatics*, doi:10.1093/ bioinformatics/ btn165
14. Lundstrm, J. *et al.* (2001) Pcons: A neural-network-based consensus predictor that improves fold recognition. *Protein Science*, **10**, 2354-2362.
15. Pauling, L. and Corey, R.B. (1951) The Pleated Sheet, a New Layer Configuration of Polypeptide Chains. *PNAS*, **37**, 251-256.
16. Qian, B. *et al.* (2007) High-resolution Structure Prediction and the Crystallographic Phase Problem. *Nature*, **450**, 259-264.
17. Rohl, C.A. *et al.* (2004) Protein Structure Prediction Using Rosetta. *Methods Enzymol*, **383**, 66-93.
18. Rost, Burkhard. (1999) Twilight zone of protein sequence alignments. *Protein Eng.*, **12**, 85-94.
19. Siew, N. *et al.* (2000) MaxSub: An Automated Measure for the Assessment of Protein Structure Prediction Quality. *Bioinformatics*, **16**, 776-785.
20. Schwede, T. *et al.* (2003) SWISS-MODEL: An Automated Protein Homology-modeling Server. *Nucleic Acids Research*, **31**, 3381-3385.
21. Sims, G.E. and Kim, S.H. (2006) A Method for Evaluating the Structural Quality of Protein Models by Using Higher-order phi-psi Pairs Scoring. *Proceeding of the National Academy of Sciences*, **103**, 4428-4432.
22. Simons, K.T. *et al.* (1997) Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions. *J. Mol. Biol.*, **268**, 209-225.
23. Samarjit Chakraborty and Somenath Biswas. (1999) Approximation Algorithms for 3-D Commom Substructure Identification in Drug and Protein Molecules. *WADS*, 253-264
24. Soding, J. (2005) Protein Homology Detection by HMM-HMM Comparison. *Bioinformatics*, **21**, 951-960.
25. Tatusov, RL *et al.* (1994) Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc Natl Acad Sci U S A*, **91**, 12091-12095.
26. Vicky Choi and Navin Goyal. (2004) A Combinatorial Shape Matching Algorithm for Rigid Protein Docking. *CPM*, 285-296.
27. Vicky Choi and Navin Goyal. (2005) An Efficient Approximation Algorithm for Point Pattern Matching Under Noise. *http://www.citebase.org/ abstract?id=oai:arXiv.org:cs/0506019*,
28. Vorobjev, Y.N. and Hermans, J. (2001) Free Energies of Protein decoys provide insight into determinants of protein stability. *Protein Science*, **10**, 2498-2506.
29. Wallner, B. and Elofsson, A. (2005) Identification of correct regions in protein models using structural, alignment and consensus information. *Protein Science*, **15**, 900-913.
30. Wallner, B. and Elofsson, Arne. (2003) Can correct protein models be identified? *Protein Science*, **12**, 1073-1086.
31. Xu, J. *et al.* (2003) Protein threading by linear programming. *Pacific Symposium on Biocomputing*, **8**, 264-275.
32. Yamada Shinsuke *et al.* (2006) Automatic extraction of conserved region from alignment based on protein structure. *Biophysics*, **46**, S265.
33. Zhang, Y. *et al.* (2005) TASSER: An Automated Method for the Prediction of Protein Tertiary Struc-

tures in CASP6. *Proteins*, **61**, 91-98.
34. Zhang, Y. (2007) Template-based Modeling and Free Modeling by I-TASSER in CASP7. *Proteins*, **69**, 108-117.
35. Zhou, H. and Zhou, Y. (2005) Fold Recognition by Combining Sequence Profiles Derived From Evolution and From Depth-Dependent Structural Alignment of Fragments. *Proteins*, **58**, 321-328.
36. Zemla A. (2003) LGA: A Method for Finding 3D Similaritise in Protein Structures. *Nucleic Acids Research*, **31**, 3370-3374.
37. Zhang, Y. and Skolnick, J. (2004) Scoring Function for Automated Assessment of Protein Structure Template Quality. *Proteins*, **57**, 702-710.
38. Ginalski, K. and Elofsson, A. and Fischer, D. and Rychlewski, L. (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19(8)**, 1015-1018.
39. Sasson, I. and Fischer, D. (2003) Modeling three-dimensional protein structures for CASP5 using the 3D-SHOTGUN meta-predictors. *Proteins: Structure, Function and Genetics*, **53** 389-394.