

# LOMETS: A local meta-threading-server for protein structure prediction

Sitao Wu and Yang Zhang\*

Center for Bioinformatics and Department of Molecular Bioscience, University of Kansas, 2030 Becker Dr, Lawrence, KS 6604, USA

Received December 22, 2006; Revised April 4, 2007; Accepted April 5, 2007

## ABSTRACT

We developed LOMETS, a local threading meta-server, for quick and automated predictions of protein tertiary structures and spatial constraints. Nine state-of-the-art threading programs are installed and run in a local computer cluster, which ensure the quick generation of initial threading alignments compared with traditional remote-server-based meta-servers. Consensus models are generated from the top predictions of the component-threading servers, which are at least 7% more accurate than the best individual servers based on TM-score at a *t*-test significance level of 0.1%. Moreover, side-chain and C-alpha ( $C_\alpha$ ) contacts of 42 and 61% accuracy respectively, as well as long- and short-range distant maps, are automatically constructed from the threading alignments. These data can be easily used as constraints to guide the *ab initio* procedures such as TASSER for further protein tertiary structure modeling. The LOMETS server is freely available to the academic community at <http://zhang.bioinformatics.ku.edu/LOMETS>.

## INTRODUCTION

The meta-server technique represents one of the major progresses in the field of protein tertiary structure prediction during recent years (1–4). It generates 3D structure predictions by taking the consensus models from a variety of individual (mainly threading/fold-recognition) servers. Various benchmarking and blind test experiments demonstrate that the consensus meta-server predictions outperform the best individual threading server (5,6).

There are, however, several drawbacks in the current meta-servers. First, all the meta-servers, including 3D-Jury (2) and GeneSilico (4), take the initial threading inputs from remote computer servers installed in other laboratories. Because of the differences in the available computer resources among different laboratories,

it is difficult to quickly collect all the threading results from the individual servers, which influences its usefulness in the large-scale protein structure prediction (7,8). Especially, some remote individual servers can be occasionally shut down or become not available. In the 3D-Jury meta-server, for example, there was only one server from FFAS03 (9) that was available during the CASP7 season. The absence of sufficient initial threading inputs will influence the performance of the final meta-server results.

The second drawback of the current meta-servers is the instability of the algorithms of the remote servers. To achieve the best performance, the meta-servers need to balance various cutoff parameters for the selection and combination of the final models. This requires careful tuning and training of the meta-server algorithms based on all the individual servers. However, the inconsistent updating and modifications of the remote individual servers make the development of a steady and robust meta-server algorithm difficult.

In this work, we developed a new meta-threading-server, LOMETS, where all nine individual threading servers are installed locally. This will allow us to control and tune our meta-server algorithms in a consistent manner, and make the users able to obtain the comprehensive predictions of all servers quickly. In addition to the construction of the best possible 3D models, the LOMETS server also provides the  $C_\alpha$  and side-chain contact and distance map predictions, combined from all threading alignments. These constraints can be used to guide the structure construction procedures such as MODELLER (10), ROSETTA (11) and TASSER (12) for generating protein tertiary models.

## METHODS

### Component threading programs in LOMETS

LOMETS server takes predictions from nine different servers that represent a diverse set of state-of-the-art threading algorithms, i.e. FUGUE (13), HHSEARCH (14), PROSPECT2 (15), SAM-T02 (16), SPARKS2 (17), SP3 (18), PAINT, PPA-I and PPA-II. The first six

\*To whom correspondence should be addressed. Tel: +1 785 864 1948; Fax: +1 785 864 5558; Email: [yzhang@ku.edu](mailto:yzhang@ku.edu)

programs were copied from other laboratories and the last three developed in our own lab. All the nine servers are installed and run in our local computer cluster with template libraries updated every week. The algorithms were selected to cover different threading methods. Here, we give a brief introduction of the methods.

**FUGUE.** FUGUE is developed at the Blindell Lab (13). It aligns target sequence profile against template structural profile collected from HOMSTRAD (19). Dynamic programming algorithm (20) is used to find the best sequence–structure match.

**PROSPECT2.** PROSPECT2 (15) is developed at the Xu Lab, which uses a score function including residue mutations, secondary structure propensity, solvent accessibility and pairwise contact potential. A divide-and-conquer searching approach (15) is exploited to generate the global optimization of alignments.

**SPARKS2 and SP3.** Both methods have been developed at the Zhou lab (17,18). In SPARKS2 (17), the authors exploit a sequence profile–profile alignment combined with a single-body knowledge-based statistical potential; in SP3 (18), they use a residue depth-dependent structure profile to replace the single-body potential in the SPARKS2. Both methods use dynamic programming for the sequence–structure alignment search.

**SAM-T02.** SAM-T02 (16) is developed at the Karplus lab, which starts from the PSI-Blast sequence database search (21). Based on the PSI-Blast multiple sequence alignment, a hidden Markov model (HMM) will be constructed in an iterative way, which is then exploited to search through the whole template library by the Viterbi algorithm (22).

**HHSEARCH.** HHSEARCH (14) is developed at the Soding lab, which aligns the profile HMM of target with the profile HMM of templates by maximizing the log-sum-of-odds score.

**PPA-I.** PPA-I is a simple sequence Profile–Profile Alignment approach combined with secondary structure matches. The alignment score between the  $i$ th residue of the query sequence and the  $j$ th residue of the template structure is defined as

$$\text{Score}(i, j) = \sum_{k=1}^{20} P_{\text{query}}(i, k) L_{\text{template}}(j, k) + c_1 \delta(s_{\text{query}}(i), s_{\text{template}}(j)) + c_2, \quad 1$$

where  $P_{\text{query}}(i, k)$  is the frequency of the  $k$ th amino acid at the  $i$ th position of the query sequence when a PSI-BLAST search of the query sequence runs against a non-redundant sequence database (<ftp://ftp.ncbi.nih.gov/blast/db/nr.Z>) with an  $E$ -value cutoff of 0.001;  $L_{\text{template}}(j, k)$  is the log-odds profile of template sequence in the PSI-BLAST search;  $s_{\text{query}}(i)$  is the secondary structure prediction from PSIPRED (23) for the  $i$ th residue of the query sequence and  $s_{\text{template}}(j)$  the secondary structure assignment by DSSP (24) for the  $j$ th residue of the template;  $\delta(s_{\text{query}}(i), s_{\text{template}}(j))$  equals to 1 if  $s_{\text{query}}(i) = s_{\text{template}}(j)$  and 0 otherwise. The weight

factor  $c_1$  is an adjustable parameter for balancing the profile term and the secondary structure matches; the shift constant  $c_2$  is introduced to avoid the alignment of unrelated regions in the local alignment (18). The Needleman–Wunsch (20) dynamic programming algorithm is used to find the best match between query and template sequences. A position-dependent gap penalty in the dynamic programming is employed: no gap is allowed inside the secondary structure regions; gap opening ( $g_o$ ) and gap extension ( $g_e$ ) penalties apply to other regions; ending gap-penalty is neglected. The four parameters [i.e.  $c_1$ ,  $c_2$ , in Equation (1), and  $g_o$ ,  $g_e$  of gap penalties in dynamic programming] are decided by trial and error on the ProSup benchmark (25).

**PPA-II.** PPA-II is also a profile–profile alignment algorithm. The only difference from PPA-I is that the sequence profiles in PPA-II are collected from SAM-T99 sequence alignments (26). Here, we do not use SAM-T02 because we found that PPA-II with SAM-T99 sequence profile generates slightly better alignments as judged by average TM-score. During the construction of sequence profiles, Henikoff weights (27) are used for re-weighting the redundant sequences.

**PAINT.** PAINT is a PAirwise-Interaction-based Threading algorithm similar to RAPTOR (28). There are five terms in PAINT’s energy function which account for environment fitness, residue mutation, secondary structure match, pair-wise interactions and gap penalty. A detailed description of the energy terms and the PAINT algorithm can be found in the Supplementary Data. Since the sequence–structure alignment is defined by the integer coefficients ( $x$ ’s) of the energy function, the goal of the PAINT threading is to identify the set of integer coefficients which maximize the total alignment score of Equation (S1). Under the constraint of Equations (S2–S5),  $x$ ’s can be solved by the established integer programming programs of GLPK (<http://www.gnu.org/software/glpk>). Since the computation of integer programming is time-consuming for big proteins, we take only a subset of template proteins which consist of top 10 templates from each of other eight threading servers. The average CPU time for the alignment of the 80 template proteins is around 5 min. There are two main differences of PAINT and RAPTOR algorithm (28). For the identification of possible alignment positions, only the alignment positions with top 40% energy score are considered for the purpose of reducing the chance of missing possible alignment positions. Second, rather than using SVM in RAPTOR, we have used a simple scaled score of  $E/L_{\text{ali}}$  for the ranking of alignments, where  $E$  is the energy score and  $L_{\text{ali}}$  is the number of aligned residues after alignment.

#### Threading model selection

Models in LOMETS are selected from individual servers purely based on consensus, i.e. the structure similarity of

the considered model with other threading alignments. For the best performance, 30 models are taken from the top predictions of the nine servers sequentially from PPA-I, SP3, PPA-II, SPARKS, PROSPECT, FUGUE, HHSEARCH, PAINT and SAM-T02, where the order of the servers are based on their performance on independent test runs. The 30 models are taken as following: First, select the first model of PPA-I and then the first model from SP3. This procedure proceeds until all the first models from nine servers are collected. Then, all the second models from nine servers are collected in the same order. The collection process proceeds and stops until 30 models have been reached. During the collection, the templates of very short alignments, i.e. the number of aligned residues is less than a quarter of the query sequence length, are neglected. The consensus score of each (*i*th) of the 30 models is calculated by the average TM-score (29):

$$\langle \text{TM-score}_i \rangle = \frac{1}{29} \sum_{j=1}^{29} \text{TM-score}_{ij}. \quad 2$$

We note that, when running the TM-score program with model<sub>*i*</sub> and model<sub>*j*</sub>, the TM-score is by default normalized by the length ( $L_j^{\text{Thr}}$ ) of the second model (i.e. model<sub>*j*</sub>). But in Equation (2) TM-score<sub>*ij*</sub> should be uniformly normalized by the query sequence length ( $L$ ). To do this, one can first run the TM-score program with an option of ‘-d  $d_0$ ’ with  $d_0 = \sqrt[3]{L} - 15 - 1.8$  to obtain TM-score<sub>*ij*</sub>( $L_j^{\text{thr}}$ ). The normalized TM-score<sub>*ij*</sub> can be then obtained by  $L_j^{\text{thr}} \text{TM-score}_{ij}(L_j^{\text{thr}})/L$ . Here, purpose of the option ‘-d  $d_0$ ’ in the TM-score program is to assign the new-defined length scale of  $d_0$  to the Levitt–Gerstein score (29).

Finally, the models are ranked based on  $\langle \text{TM-score}_i \rangle$ , i.e. the models with higher average TM-score to other models are ranked higher.

### Spatial constraints

For each protein, threading models are categorized as ‘good’ or ‘bad’ depending on whether the inherent Z-score (the energy in standard deviation units relative to mean) of the alignment is above or below a threshold Z-score<sub>cut</sub>. The threshold cutoff is determined by the minimization of the false positive (high Z-score but with low TM-score) and false negative rate (low Z-score but with high TM-score) of each threading program based on an independent benchmark set of 1489 non-redundant proteins (12). For PPA-I, SP3, PPA-II, SPARKS2, PROSPECT2, FUGUE, HHSEARCH, PAINT and SAM-T02, the Z-score<sub>cut</sub> are 8.2, 8.0, 7.0, 8.8, 4.0, 6.0, 11.0, 0.5 and 9.5, respectively. If the total number of ‘good’ models is more than nine (i.e. on average at least one ‘good’ model from each server), the target is defined as an ‘Easy’ target; if there is no ‘good’ model at all in all the servers, the target is a ‘Hard’ target; otherwise, it is a ‘Medium’ target. For Easy/Medium/Hard targets,  $N$  (=20/30/50) highest confident models are selected from the servers for the next constraint construction. The ‘good’ models and then the ‘bad’ models are taken in

a sequential server order as mentioned above until  $N$  models are selected. The logic for the decision of  $N$  is the following: for ‘Easy’ targets where we have good templates, about top two (good) templates on average are taken from each program while including more templates with bad quality will bring more noises for the good templates. For the ‘Medium’ and ‘Hard’ targets where we do not have good templates and constraints overall, we will take more templates to enhance the consensus information because there are usually some partially correct substructures even in the low rank templates which may be identified by the consensus selections.

There are four types of spatial constraints that are collected from the  $N$  selected threading alignments:

**Side-chain contacts.** A pair of side-chains is considered as contact if the distance between the centers of mass in the aligned templates is below an amino acid specific cutoff:

$$\text{cut}(A, B) = d(A, B) + 2.5\Delta(A, B) \quad 3$$

Here  $d(A, B)$  was obtained by calculating the average distance of side-chain centers of mass of the contacted residues  $A$  and  $B$  with at least one pair of heavy atoms in  $A$  and  $B < 4.5 \text{ \AA}$  in 6379 non-homologous PDB structures.  $\Delta(A, B)$  is the SD of  $d(A, B)$ . The data of  $d(A, B)$  and  $\Delta(A, B)$  can be seen at our website [http://zhang.bioinformatics.ku.edu/LOMETS/sidechain\\_contact.txt](http://zhang.bioinformatics.ku.edu/LOMETS/sidechain_contact.txt). In the side-chain contact file of LOMETS server, we list the identities of all the contacts with contact order  $\geq 5$ , as well as the confidence score that is defined as the number of occurrences of the contacts divided by the total number of templates that have both residues aligned.

**C<sub>α</sub> contacts.** The C<sub>α</sub>-contact file lists the identity of all predicted C<sub>α</sub> pairs in contact with contact order  $\geq 5$  and the confidence score. A pair of C<sub>α</sub>s is considered as contact if the distance of C<sub>α</sub> atoms is below 6 Å.

**Long-range C<sub>α</sub> distance map.** This file contains the C<sub>α</sub>-distances between  $i$  and  $i+j*10$  residues ( $i=1, \dots, L$ ;  $j=1, 2, \dots$ ), which are collected from the top four templates.

**Short-range C<sub>α</sub> distance constraints.** This file contains the average C<sub>α</sub>-distances of  $i$  and  $i+j$  residues ( $i=1, \dots, L$ ;  $j=2, \dots, 6$ ), taken from all  $N$  templates. It includes only local structure information and can be used for guiding the protein-like secondary structure construction.

## RESULTS

For the testing of the LOMETS server, we select 620 non-homologous proteins (<25% sequence identity with lengths from 50 to 600) from PDBSELECT (2006 March) (30). A list of the 620 benchmark proteins and the threading results of all nine programs are available at <http://zhang.bioinformatics.ku.edu/LOMETS/benchmark.html>.

### Threading alignment and consensus selections

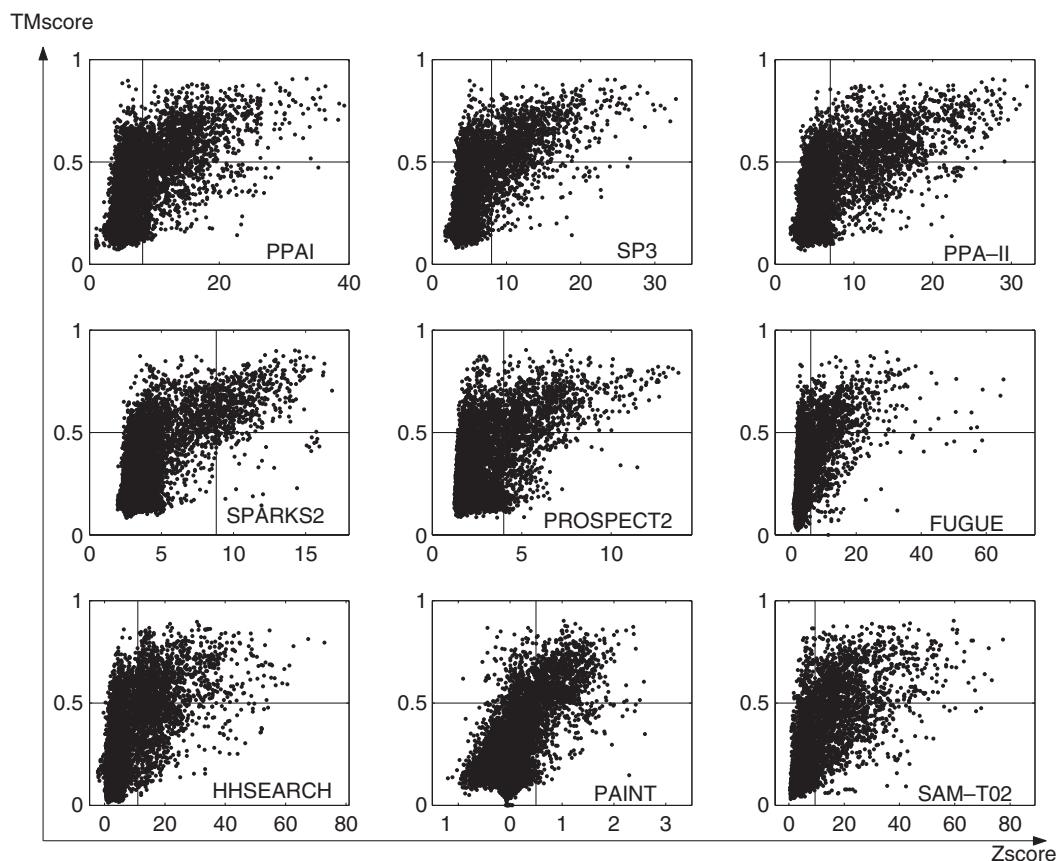
In Figure 1, we present the threading results of the nine individual servers on the 620 benchmark proteins, where all homologous templates with sequence identity to targets  $>30\%$  have been removed from the template library. Since all servers run locally, we could obtain the threading results quickly and the average CPU time for one target is less than 20 min in our computer cluster when we run them at nine nodes in parallel. There is an obvious correlation between the TM-score and the Z-score of each server. We also show the Z-score cutoff in each plot. If we use  $\text{TM-score} \geq 0.5$  (or  $<0.5$ ) to define a correct (or wrong) threading model, the false negative and false positive rates of the Z-score cutoffs are: 0.0444 and 0.0622 (for PPA-I), 0.0515 and 0.0282 (for SP3), 0.0359 and 0.0597 (for PPA-II), 0.0829 and 0.0045 (for SPARKS2), 0.0602 and 0.0376 (for PROSPECT2), 0.0183 and 0.0447 (for FUGUE), 0.0339 and 0.0733 (for HHSEARCH), 0.0219 and 0.0193 (for PAINT) and 0.0154 and 0.0831 (for SAM-T02).

In Table 1, we list the average TM-score, RMSD, and alignment coverage of all threading programs on the 620 proteins. Below the values, we also list the average TM-score and RMSD of the full-length models built by MODELLER v8.2 (10), where external constraints from LOMETS are incorporated. We found that MODELLER

generates slightly better results when using the LOMETS spatial constraints than running MODELLER by default. Based on the average TM-score, the improvement is  $\sim 0.8\%$ . Except for the external constraint file, MODELLER has an option to include multiple templates where MODELLER extracts constraints from multiple templates by itself. By trial and error, we found that for ‘Easy’ targets the MODELLER program using up to five consensus templates ( $0.75 < \text{TM-score} < 1.0$ ) as input works the best. For ‘Medium’ and ‘Hard’ targets, the structures of top templates are usually divergent and only one template is exploited here. These full-length models built by MODELLER are also provided at the LOMETS server.

Since the lengths of MODELLER models are longer than those of threading alignments, the average TM-scores of the full-length MODELLER models are relatively larger than the threading alignments although the topology of the core regions are unchanged. The increment of TM-score ranges from 2.9% (PPA-I) to 5.6% (PAINT) depending on the threading alignment coverage. In general, the smaller the threading alignment coverage is, the bigger increment the TM-score of MODELLER models has, because more residues have been added in the full-length models.

Although both MODELLER (10) and I-TASSER (31) make use of consensus restraints from templates in their



**Figure 1.** TM-score of threading alignments of nine component servers on 620 non-homologous proteins versus the Z-score, where Z-score is defined as the deviation of the inherent raw score from mean divided by the SD. The vertical line in each box indicates a Z-score cutoff to distinguish ‘bad’ and ‘good’ predictions.

structure modeling, the structure improvement of I-TASSER models on the templates is much higher. Based on the recent CASP7 experiment, the average TM-score of the models generated by I-TASSER ('Zhang-Server') is 16.9% higher than that of the best template (32). There may be two factors contributing to the difference. First, the I-TASSER force field includes a variety of knowledge-based, protein-sequence specific/nonspecific potentials obtained from variant resources (31), which has been optimized using structure decoys. Second, the conformational space of MODELLER is searched using a conjugated gradient algorithm, which is a local minimization method. The advantage of the conjugated gradient method is the quick convergence to the local minimum of an object function. But if the external restraint is different from the initial templates, the method does not guarantee the optimal satisfaction of all the constraints. In contrast, the conformational space in I-TASSER is searched by the parallel Monte Carlo sampling method (33), the goal of which is to identify the lowest free-energy state by global search. But the Monte Carlo simulation of I-TASSER takes much longer CPU time than MODELLER does. Since the major purpose of the LOMETS server is to provide a quick collection of the alignments and restraints from multiple local threading servers, we do not include the I-TASSER simulation here. A publicly available server of the I-TASSER algorithm is provided separately at our website: <http://zhang.bioinformatics.ku.edu/I-TASSER>.

At the bottom of Table 1, we show the result of LOMETS consensus selections. The average TM-score of the first model in LOMETS is 0.4287, ~7% better than the best individual server (PPA-I). This difference is statistically significant, which is at a 0.1% significance level based on the *t*-test. The TM-score of the best in top five models of the LOMETS selection is shown at Column 3, which also outperforms the best individual server. The higher TM-score of the LOMETS models demonstrates a better balance of RMSD (Columns 4 and 5) and alignment coverage (Columns 6 and 7) in comparison with that of the individual servers.

As a control, we also downloaded the PCONS5, the newest version of PCONS meta-server selection program by Wallner and Elofsson (34), which combines consensus analysis (by LGscore), structural evaluation and inherent score of threading servers. The PCONS5 selection result is listed in the last row of Table 1. The selection of PCONS5 is also better (~3%) than the best individual server but not better than LOMETS. This result seems to indicate that the consensus analysis, which is the only fact adopted in LOMETS by TM-score analysis, is the most robust factor of meta-server selections.

### Spatial constraint predictions

The effect of spatial constraints on the protein structure modeling is a tradeoff of the prediction accuracy (Acc) and the prediction coverage (Cov) (35). For the quantitative evaluation of the  $C_\alpha$  and side-chain contact predictions, we define

$$\text{Acc} = \frac{N_{\text{corr}}}{N_{\text{pred}}}, \quad \text{Cov} = \frac{N_{\text{pred}}}{L} \quad 4$$

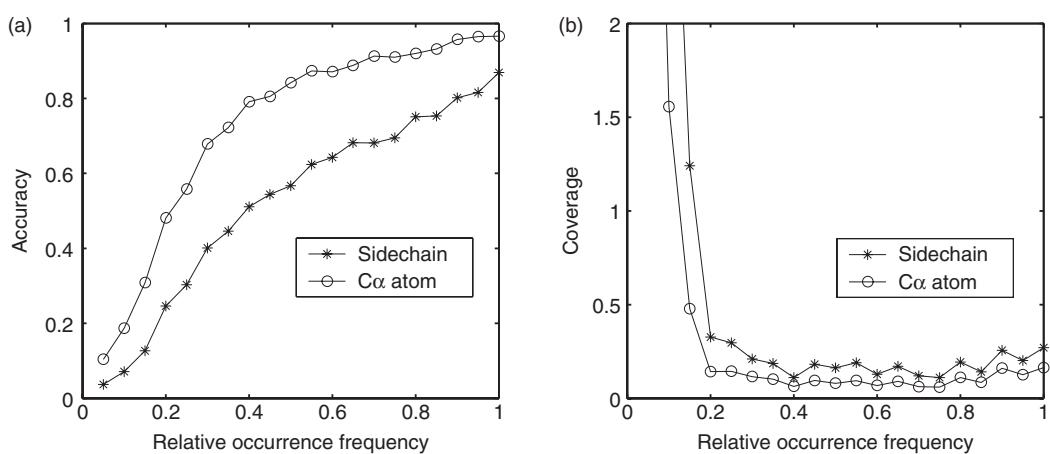
where  $N_{\text{corr}}$  is the number of correctly predicted contacts that are true contacts in native structures based on the same distance cutoff of Equation (3),  $N_{\text{pred}}$  is the number of total predicted contacts and  $L$  is the length of target sequence.

In Figure 2a, we show the accuracy of predicted contacts versus relative occurrence frequencies with which the contacts occurred in the models for the 620 testing proteins. Here relative occurrence frequency for a contact is defined as  $N_0/N$ , where  $N_0$  is the number of templates having the contact and  $N$  (=20/30/50) the total number of the selected threading templates. It is worth noting that accuracy in Figure 2a is non-cumulative, i.e. the accuracy at frequency  $f$  is an average accuracy calculated in  $[f - 0.05, f + 0.05]$ . As expected, the more often the contacts occur, the more accurate the contacts are, which indicate that the occurrence frequency can be considered as a confidence score for the contact prediction.

**Table 1.** Summary of component-threading programs and the meta-server selections

Threading servers or meta-servers	TM-score of threading alignments (MODELLER models)		RMSD ( $\text{\AA}$ ) of aligned residues (MODELLER models)		Coverage <sup>a</sup> of threading alignments	
	First model	Best in top five models	First model	Best in top five models	First model	Best in top five models
PPA-I	0.4001 (0.4117)	0.4389 (0.4531)	10.11 (16.66)	9.13 (14.02)	0.831	0.846
SP3	0.3991 (0.4138)	0.4391 (0.4551)	10.50 (13.86)	9.62 (12.83)	0.858	0.867
PPA-II	0.3900 (0.4076)	0.4306 (0.4512)	10.72 (14.89)	9.40 (13.02)	0.837	0.847
SPARKS2	0.3855 (0.3973)	0.4283 (0.4441)	11.62 (13.60)	10.03 (12.23)	0.895	0.893
PROSPECT2	0.3793 (0.3914)	0.4245 (0.4384)	12.19 (13.01)	10.68 (12.02)	0.903	0.903
FUGUE	0.3580 (0.3721)	0.4038 (0.4173)	10.78 (19.26)	10.30 (15.82)	0.827	0.872
HHSEARCH	0.3635 (0.3827)	0.4016 (0.4224)	6.92 (22.38)	6.44 (19.04)	0.607	0.643
PAINT	0.3558 (0.3758)	0.4045 (0.4210)	10.35 (15.74)	9.86 (14.21)	0.735	0.786
SAM-T0 2	0.3402 (0.3575)	0.3798 (0.3971)	10.19 (21.75)	9.83 (17.53)	0.721	0.777
LOMETS	0.4287 (0.4434)	0.4481 (0.4669)	10.18 (10.99)	9.49 (10.61)	0.890	0.882
PCONS5	0.4117 (0.4272)	0.4434 (0.4628)	10.03 (15.39)	9.14 (13.67)	0.840	0.852

<sup>a</sup>Coverage = length of aligned residues/length of target sequence.



**Figure 2.** (a) Average accuracy of predicted C<sub>α</sub> and side-chain contacts versus the relative occurrence frequency of the contacts in the LOMETS threading templates. (b) Coverage of the predicted contacts versus the relative occurrence frequency. For each frequency value ( $f$ ), the data is calculated as an average within the bin of  $[f - 0.05, f + 0.05]$ .

**Table 2.** Summary of constraint predictions by LOMETS and the threading programs (with a relative occurrence frequency  $\geq 0.18$  for contact predictions)

Threading servers or meta-servers	Acc <sup>a</sup> <sub>sc</sub>	Cov <sup>b</sup> <sub>sc</sub> 2	Acc <sup>c</sup> <sub>C<sub>α</sub></sub> 3	Cov <sup>d</sup> <sub>C<sub>α</sub></sub>	Dif <sup>e</sup> <sub>short</sub>	No <sup>f</sup> <sub>short</sub>	Dif <sup>g</sup> <sub>long</sub>	No <sup>h</sup> <sub>long</sub>
PPA-I	0.249	1.655	0.431	0.696	1.178	600.5	3.732	1159.5
SP3	0.239	1.713	0.405	0.712	1.220	612.3	3.817	1196.6
PPA-II	0.253	1.527	0.410	0.661	1.216	591.3	3.894	1173.0
SPARKS2	0.223	1.654	0.375	0.659	1.356	629.4	3.804	1203.8
PROSPECT2	0.236	1.599	0.411	0.653	1.219	631.7	3.591	1198.0
FUGUE	0.221	1.185	0.379	0.438	1.586	625.3	3.649	1175.7
HHSEARCH	0.359	0.842	0.528	0.404	1.024	357.1	4.111	743.7
PAINT	0.248	1.174	0.372	0.529	1.267	527.3	4.138	1103.2
SAM-T02	0.227	1.164	0.350	0.534	1.597	520.0	3.923	1019.4
LOMETS	0.421	0.910	0.607	0.405	1.186	632.7	3.455	1193.0

<sup>a</sup>ACC<sub>sc</sub>: Average accuracy for side-chain center of mass contact predictions.

<sup>b</sup>Cov<sub>sc</sub>: Average coverage for side-chain center of mass contact predictions.

<sup>c</sup>Acc<sub>C<sub>α</sub></sub>: Average accuracy for C<sub>α</sub> atom contact predictions.

<sup>d</sup>Cov<sub>C<sub>α</sub></sub>: Average coverage for C<sub>α</sub> atom contact predictions.

<sup>e</sup>Dif<sub>short</sub>: Average difference (Å) between native and predicted short-range C<sub>α</sub>-distances.

<sup>f</sup>No<sub>short</sub>: Average number of predicted short-range C<sub>α</sub>-distances.

<sup>g</sup>Dif<sub>long</sub>: Average difference (Å) between native and the best predicted long-range C<sub>α</sub>-distances.

<sup>h</sup>No<sub>long</sub>: Average number of the best predicted long-range C<sub>α</sub>-distances.

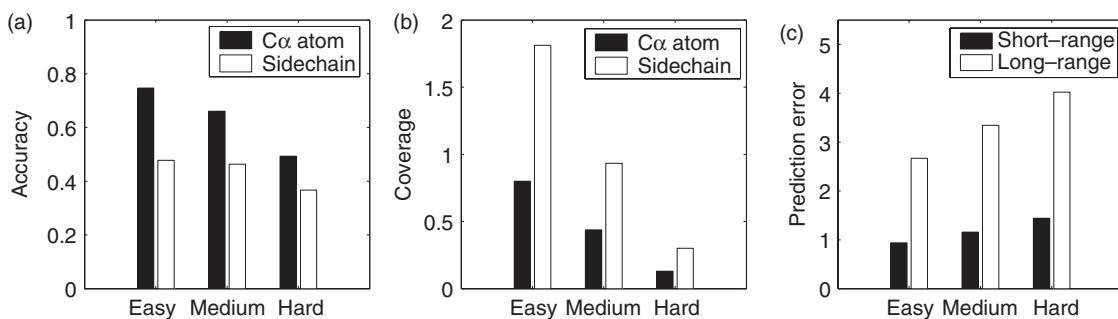
In Figure 2b, we show how the prediction coverage is reduced with increasing the relative occurrence frequency.

As demonstrated in our previous study (35), an accuracy of side-chain contact constraints of  $>22\%$  has a positive effect on *ab initio* protein structure modeling. This accuracy value corresponds to the occurrence frequency of  $\sim 0.18$  in Figure 2a.

In Table 2, we list a summary of contact predictions of C<sub>α</sub> and side-chains by LOMETS and its component threading programs with a confidence score  $\geq 0.18$  (Columns 2–5). Here the constraints of a single threading program are collected from the top ten templates. Obviously, the spatial constraints from consensus metaservers have much higher accuracy than those from individual threading programs.

In Column 6, we present the average differences between the native and the predicted distances for short-range C<sub>α</sub> distance maps of  $|i-j| < 7$ . Column 7 is the average

numbers of the predicted short-range distance constraints. For long-range C<sub>α</sub> distance map of  $|i-j| \geq 10$ , we generate up to four predictions for each pair of residues. The eighth column shows the average error of the best predicted long-range C<sub>α</sub> distance pairs and the ninth column gives the average numbers of the long-range distance constraints. Because of the differences in accuracy and coverage of threading alignments, the accuracy and number of distance constraints are different among the threading programs. For example, HHSEARCH has the highest accuracy of short-range distance constraints but the number of short-range distance constraints is the lowest because it has no alignment in many uncertain regions. For a balance of the accuracy and number of distance constraints, the consensus LOMETS has obviously the highest accuracy with a reasonable number of distance constraints on the distance maps.



**Figure 3.** The average result of spatial constraint predictions for ‘Easy’, ‘Medium’ and ‘Hard’ targets on 620 non-homologous proteins. (a) Accuracy of  $\text{C}_\alpha$  and side-chain center contact predictions. (b) Coverage of  $\text{C}_\alpha$  and side-chain center contact predictions. (c) Prediction error of short-range and the best long-range distance map.

The accuracy of the spatial constraints relies on the quality of the threading templates. In Figure 3, we plot the histogram of the prediction accuracy of constraints for 200 ‘Easy’, 120 ‘Medium’ and 300 ‘Hard’ proteins separately. Obviously, for ‘Easy’ targets, the templates have a better quality of alignments and the accuracy of constraints is higher than that of ‘Medium’ and ‘Hard’ targets (Figure 3a and c). Moreover, the number of aligned residues in ‘Easy’ targets is higher and the alignments by different servers are more consistent, which make the prediction coverage in ‘Easy’ targets also higher than that of ‘Medium’ and ‘Hard’ targets (Figure 3b). Here, because of the fixed small  $\text{C}_\alpha$  distance cutoff [ $<6\text{\AA}$  as used in TASSER modeling (12)], the coverage of  $\text{C}_\alpha$  contacts is lower than that of side-chain center contacts. The average accuracy of  $\text{C}_\alpha$  is also higher than that of side-chain centers which may be due to the fact that side-chain rotamers have more structure variations.

## SUMMARY

We have developed a quick and automated meta-server, LOMETS, for protein structure predictions. Different from other on-line meta-servers, all nine component-threading servers are installed and run in our local computer cluster. The local installation of the servers greatly speeds up the coherent generation of initial threading alignments, as well as facilitates the development of a robust and well-tuned meta-server algorithm. The consensus prediction taken from LOMETS servers is at least 7% more accurate than all the individual servers. The difference is also statistically meaningful with a *t*-test at 0.1% of significance level. The average CPU time for a medium size protein ( $\sim 200$  residues) is less than 20 min when the programs are run in parallel on nine nodes of our cluster.

In addition to the threading alignments, LOMETS also provides highly accurate contact and distance predictions for the query sequences. In our benchmark testing of 620 proteins, the average accuracy of side-chain center contacts is 0.42 with coverage of 91%; the average accuracy of  $\text{C}_\alpha$  contacts is 0.61 with coverage of 41%. The average errors of the best long- and short-range

distance map prediction are 3.5 and  $1.2\text{\AA}$ , respectively. These data can be easily used as constraints to guide the tertiary structure modeling procedures such as MODELLER (10), ROBETTA (11), TASSER (12,36).

Last but not the least, the template libraries of all nine servers are kept updated every week. We have managed to generate template files in our local computers for SAM-T02, PROSPECT2, SPARKS2, SP3, PPA-I, PPA-II and PAINT. The template library for FUGUE and HHSEARCH are automatically downloaded from the authors’ websites (i.e. <ftp://merlin.bioc.cam.ac.uk/pub/software/fugue/data> and [ftp://ftp.tuebingen.mpg.de/pub/protevo/HHsearch/databases/pdb70\\_\\*.hmm.tar.gz](ftp://ftp.tuebingen.mpg.de/pub/protevo/HHsearch/databases/pdb70_*.hmm.tar.gz)), which are also kept updated each week.

LOMETS will be open to add new and efficient threading programs when they become available.

## SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

## ACKNOWLEDGEMENTS

We want to thank Drs K. Karplus, K. Mizugushi, J. Soding, Y. Xu and Y. Zhou, for sending us the copies of their threading programs of SAM-T02, FUGUE, HHSEARCH, PROSPECT2, SPARKS2 and SP3. The project is partially supported by KU Start-up Fund 06194 and NGRF 2302003. Funding to pay the Open Access publication charges for this article was provided by KU Start-up Fund 06194.

*Conflict of interest statement.* None declared.

## REFERENCES

1. Lundstrom,J., Rychlewski,L., Bujnicki,J. and Elofsson,A. (2001) Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci.*, **10**, 2354–2362.
2. Ginalski,K., Elofsson,A., Fischer,D. and Rychlewski,L. (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19**, 1015–1018.
3. Fischer,D. (2003) 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, **51**, 434–441.
4. Kurowski,M.A. and Bujnicki,J.M. (2003) GeneSilico protein structure prediction meta-server. *Nucleic Acids Res.*, **31**, 3305–3307.

5. Fischer,D., Rychlewski,L., Dunbrack,R.L.Jr., Ortiz,A.R. and Elofsson,A. (2003) CAFASP3: the third critical assessment of fully automated structure prediction methods. *Proteins*, **53**(Suppl. 6), 503–516.
6. Rychlewski,L., Fischer,D. and Elofsson,A. (2003) LiveBench-6: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **53**(Suppl. 6), 542–547.
7. Skolnick,J., Fetrow,J.S. and Kolinski,A. (2000) Structural genomics and its importance for gene function analysis. *Nat. Biotechnol.*, **18**, 283–287.
8. Baker,D. and Sali,A. (2001) Protein structure prediction and structural genomics. *Science*, **294**, 93–96.
9. Jaroszewski,L., Rychlewski,L., Li,Z., Li,W. and Godzik,A. (2005) FFAS03: a server for profile–profile sequence alignments. *Nucleic Acids Res.*, **33**, W284–W288.
10. Sali,A. and Blundell,T.L. (1993) Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, **234**, 779–815.
11. Simons,K.T., Kooperberg,C., Huang,E. and Baker,D. (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.
12. Zhang,Y. and Skolnick,J. (2004) Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl Acad. Sci. USA*, **101**, 7594–7599.
13. Shi,J., Blundell,T.L. and Mizuguchi,K. (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.*, **310**, 243–257.
14. Soding,J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics (Oxford, England)*, **21**, 951–960.
15. Xu,Y. and Xu,D. (2000) Protein threading using PROSPECT: design and evaluation. *Proteins*, **40**, 343–354.
16. Karplus,K., Karchin,R., Draper,J., Casper,J., Mandel-Gutfreund,Y., Diekhans,M. and Hughey,R. (2003) Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. *Proteins*, **53**(Suppl. 6), 491–496.
17. Zhou,H. and Zhou,Y. (2004) Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins*, **55**, 1005–1013.
18. 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.
19. Mizuguchi,K., Deane,C.M., Blundell,T.L. and Overington,J.P. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.*, **7**, 2469–2471.
20. Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
21. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
22. Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, **77**, 257–286.
23. Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.
24. Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.
25. Domingues,F.S., Lackner,P., Andreeva,A. and Sippl,M.J. (2000) Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. *J. Mol. Biol.*, **297**, 1003–1013.
26. Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.
27. Henikoff,S. and Henikoff,J.G. (1994) Position-based sequence weights. *J. Mol. Biol.*, **243**, 574–578.
28. Xu,J., Li,M., Kim,D. and Xu,Y. (2003) RAPTOR: optimal protein threading by linear programming. *J. Bioinform. Comput. Biol.*, **1**, 95–117.
29. Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.
30. Hobohm,U. and Sander,C. (1994) Enlarged representative set of protein structures. *Protein Sci.*, **3**, 522–524.
31. Wu,S., Skolnick,J. and Zhang,Y. (2007) Ab initio modeling of small proteins by iterative TASSER simulations. *BMC Biology*, In press.
32. Zhang,Y. (2007) Template-based modeling and free modeling by I-TASSER in CASP7. *Proteins*, In press.
33. Zhang,Y., Kihara,D. and Skolnick,J. (2002) Local energy landscape flattening: Parallel hyperbolic Monte Carlo sampling of protein folding. *Proteins*, **48**, 192–201.
34. Wallner,B. and Elofsson,A. (2005) Pcons5: combining consensus, structural evaluation and fold recognition scores. *Bioinformatics*, **21**, 4248–4254.
35. Zhang,Y., Kolinski,A. and Skolnick,J. (2003) TOUCHSTONE II: A new approach to ab initio protein structure prediction. *Biophys. J.*, **85**, 1145–1164.
36. Zhang,Y. and Skolnick,J. (2004) Tertiary structure predictions on a comprehensive benchmark of medium to large size proteins. *Biophys. J.*, **87**, 2647–2655.