Published in final edited form as:
J Bioinform Comput Biol. 2012 June ; 10(3): 1242003. doi:10.1142/S0219720012420036.

RECURSIVE PROTEIN MODELING: A DIVIDE AND CONQUER STRATEGY FOR PROTEIN STRUCTURE PREDICTION AND ITS CASE STUDY IN CASP9

JIANLIN CHENG[*,†,§], **JESSE EICKHOLT**[*,¶], **ZHENG WANG**[*,‡,||], and **XIN DENG**[*,**]

[*] Department of Computer Science, University of Missouri Columbia, MO 65211, USA

[†] Informatics Institute, University of Missouri Columbia, MO 65211, USA

[‡]Bond Life Science Center, University of Missouri Columbia, MO 65211, USA

Abstract

After decades of research, protein structure prediction remains a very challenging problem. In order to address the different levels of complexity of structural modeling, two types of modeling techniques — template-based modeling and template-free modeling — have been developed. Template-based modeling can often generate a moderate- to high-resolution model when a similar, homologous template structure is found for a query protein but fails if no template or only incorrect templates are found. Template-free modeling, such as fragment-based assembly, may generate models of moderate resolution for small proteins of low topological complexity. Seldom have the two techniques been integrated together to improve protein modeling. Here we develop a recursive protein modeling approach to selectively and collaboratively apply template-based and template-free modeling methods to model template-covered (i.e. certain) and template-free (i.e. uncertain) regions of a protein. A preliminary implementation of the approach was tested on a number of hard modeling cases during the 9th Critical Assessment of Techniques for Protein Structure Prediction (CASP9) and successfully improved the quality of modeling in most of these cases. Recursive modeling can signicantly reduce the complexity of protein structure modeling and integrate template-based and template-free modeling to improve the quality and efficiency of protein structure prediction.

Keywords

Protein structure prediction; recursive protein modeling; template-free modeling; template-based modeling; CASP

1. Introduction

Predicting protein tertiary structure from protein sequence is important for protein engineering, protein design and protein function analysis.[1] It is becoming more and more important in the post-genomic era as the vast majority of millions of protein sequences being generated by high-throughput next-generation sequencing projects do not have known

[§] Corresponding author chengji@missouri.edu.
[¶]jlec95@mail.mizzou.edu
[||]zwyw6@mail.mizzou.edu
[**]xd9d3@mail.mizzou.edu

structures.[2] Currently, there are more than 100 million protein sequences in GenBank,[3] whereas only about 70,000 of them have known structures in the Protein Data Bank (PDB).[4]

In order to address this challenge, two major types of protein structure modeling methods have been developed to model protein structure from sequence — template-based modeling and template-free modeling. Template-based modeling (e.g. comparative modeling or homology modeling) builds the structure of a query protein from the known structures of other proteins (i.e. templates), which are homologous to the query.[5–7] Template-free (e.g. *ab initio*) modeling folds the structure of a query protein from scratch without explicitly referring to specific structural templates.[8,9] Template-based methods work well if an appropriate template structure (e.g. a close homolog) can be found, but fails to produce an accurate structure if no template is available or only incorrect templates are used. At present, template-free modeling can generate low resolution models for small proteins with simple topologies. This is due to the difficulty of efficiently exploring the huge conformation space.

Although a variety of methods have been developed and tested for template-based and template-free modeling, only a few have integrated the two methodologies together to improve protein structure prediction. Initial efforts at combining both approaches were aimed at modeling relatively small local regions and applying *ab initio* methods to model loops[10–14] or N-/C- terminal tail regions of existing models.[15] Inspired by these initial attempts and the hierarchical protein folding process,[16,17] we designed a general, iterative, and recursive protein folding procedure to seamlessly integrate the complementary strengths of both template-based and template-free methods to effectively and efficiently predict the structures of any proteins. The approach can reduce the complexity of protein modeling by dividing the modeling problem into certain (i.e. template-based) and uncertain (i.e. template-free) regions. The regions are then modeled recursively and collaboratively using the appropriate techniques and the most useful information. The approach was implemented in our MULTICOM protein structure prediction system[18] that uses alternative alignments and multiple templates in conjunction with both the focused template-based model generation and the more exploratory template-free model generation in order to construct an ensemble of models for selection. The recursive modeling approach was blindly tested on a number of hard protein targets in the ninth Critical Assessment of Techniques for Protein Structure Prediction (CASP9) (http://predictioncenter.org/casp9/).[19] The approach successfully improved the accuracy of predicted models in a majority of cases. The experiment demonstrated that the recursive protein modeling approach can integrate template-based and template-free information together in a collaborative and reinforcing way to address a full spectrum of protein modeling problems.

It is worth noting that our recursive protein modeling approach is a kind of the Divide and Conquer protein modeling strategy of reducing modeling complexity that had been explored in protein structure prediction. Other related previous protein structure prediction work adopting Divide and Conquer strategies includes separating modeling of loops from regular structures,[11–14] assembling protein models from fixed-size fragments and super-secondary structures,[8,9] dividing modeling of a whole multi-domain protein into individual domains widely used in CASPs,[19–21] *ab initio* simulation of protein terminals,[15] and distinguishing modeling of easy parts of a protein from hard parts.[6,22] The main conceptual differences between our method and previous methods are the generality of our definition of certain and uncertain regions and the progressive expansion of certain regions through the collaborative inter-play between template-based modeling and template-free modeling. As the same modeling protocol is iteratively applied to the same query protein with shrinking uncertain regions, the process is kind of recursive. The conceptual differences and other substantial implementation differences with other Divide and Conquer strategies are described in greater details in Sec. 2.

2. Methods

2.1. A general recursive modeling procedure

In the recursive protein modeling procedure, a query protein is first searched against a template protein library using a sequence or profile alignment method.[23–28] A query-template sequence alignment will be generated if some seemly homologous/analogous templates or template fragments are found. The sequence of the query protein is then initially decomposed into certain and uncertain regions based on its alignment with the significant homologous template hits. Certain regions correspond to portions of the query sequence which align well with any one of significant homologous templates (e.g. low PSI-BLAST e-value < 0.001)[23]; and uncertain regions are the long query regions (e.g. ≥20 residues) that are not covered by a template or aligned with low confidence. The short unaligned regions in the query sequence are not considered as uncertain regions here. Instead, they are treated as loops in the certain regions to be handled by template-based modeling. Therefore, the uncertain regions in the decomposition usually correspond to one or more domains or a large portion of a domain composed of different kinds of secondary structures rather than a single loop, which distinguishes our approach from traditional protein loop modeling. Similarly, a certain region may correspond to any component (a part of a domain, a domain, or multiple domains) of a protein. After the decomposition, the conformations of the certain regions are generated by template-based modeling using the alignments and the corresponding template structures while leaving the uncertain regions alone. It is worth noting that in a complicated situation, one query may have multiple disjoint certain regions covered by one template or multiple templates. In practice, this situation does not pose any difficulties as such regions can be handled altogether by current template-based modeling tools. While keeping the conformation of the certain regions which usually form the core of the structure fixed or rigid, template-free modeling methods are applied to sample the conformations of uncertain regions. This template-free sampling is different from an independent, free sampling of uncertain regions because of the influence of the certain regions (e.g. core) is taken into account in both conformation sampling and energy assessment. The core-restrained sampling can often improve the effectiveness and efficiency of template free sampling by dragging the “wild,” free conformation toward the core region through the presence of the conformation of the certain core region and its energy. This method is particularly effective for sampling partial uncertain regions of a single protein domain taking into account of the influence of its certain/fixed regions. It can also facilitate docking whole uncertain domains with whole certain domains together by taking into account interactions between them when modeling multi-domain proteins, which may work better than traditional approaches splitting multi-domain proteins into separate chunks, modeling them independently and then almost randomly assemble them together. However, since the latter approaches are generally faster at modeling individual domains that can largely fold rather independently, in order to speed up sampling in practice, sometime it is necessary to model domain-level uncertain regions in a large multi-domain protein using template-free modeling separately. In this situation, the conformations of different regions simulated by either template-based or template-free modeling will then be used as templates by an extra template-based modeling to combine them into one full-length model.

After a round of sampling, the quality of each certain and uncertain region is assessed using global/local protein model quality assessment methods.[29–34] The conformations of certain regions and some well-modeled uncertain regions are combined into larger certain regions, leaving a smaller set of uncertain regions. The same modeling process is applied to model the newly defined certain and uncertain regions by using the conformations generated in the last iteration for the certain regions as new templates. After each round of modeling, the new certain region becomes larger than before because it includes both the previous certain region and all or a part of the previous uncertain regions that has been improved to an acceptable quality. Since the larger certain region is used as templates by template-based modeling in the next round of modeling, their conformation will be kept (almost) completely fixed, leaving a smaller uncertain region for template-free modeling to explore. The process continues until no uncertain regions remain or the quality of the entire query protein is acceptable according to model quality assessment. The entire procedure is described in Fig. 1.

It is worth pointing out that the term "region" here may refer to any level of protein structure, such as a part of a domain, an entire domain, or even multiple domains. It is different from the *ab initio* loop modeling that is exclusively used to build a loop joining two parts of protein structure. Here, conceptually the recursive modeling procedure aims to build a protein structure from smaller components in a bottom-up, hierarchical way. On one hand, it somewhat conceptually mimics or is in accordance with the physical, hierarchical protein folding process, where local regions fold firstly and then interact to fold into larger protein conformations,[16,17] although each decomposed region may not actually correspond to a physical folding unit. On the other hand, the procedure is in accordance with the "divide and conquer strategy" widely used in computer science, where a complicated problem is divided into smaller, easier to solve problems, and the solutions to the smaller problems are combined recursively in order to solve the larger problem. In the protein modeling context, the procedure improves template-based modeling by better packing of long unaligned regions (e.g. loops, tails, and small domains) and enhances template-free modeling by utilizing the template core as restraints. The protocol can not only integrate template-base and template-free modeling seamlessly and collaboratively, but also improve the quality and speed of protein modeling.

One major conceptual difference between our method and the other Divide and Conquer approach — TASSER[6] and I-TASSER[22] is that our method reduces uncertain region and enlarges certain region from iteration to iteration, while TASSER and I-TASSER use the distance constraints extracted from the models generated in the last iteration as whole to guide modeling in the next iteration without gradually keeping a larger and larger region fixed from iteration to iteration. So our method is a recursively progressive approach i.e. the newly fixed region always contains the previously fixed region and likely more in order to achieve the higher simulation efficiency, while I-TASSER may simulate the same regions again and again with different constraints from iteration to iteration. Apart from TASSER and I-TASSER, our method synergistically models certain and uncertain regions using both template-based and template-free modeling alternately in order to shrink uncertain regions and expand certain regions gradually to reach a locally, approximately optimal solution. During each iteration, template-base and template-free modeling influence/improve each other through the progressive expansion and growth of certain regions. That is, from iteration to iteration, template-based modeling creates a largely fixed conformation of enlarged certain regions to facilitate template-free modeling of shrunk uncertain regions, whereas template-free modeling continuously reduces the size of uncertain regions by turning some or the whole uncertain regions into certain regions. There are also substantial implementation differences between our method and I-TASSER. For example, our method uses fragment assembly to model uncertain regions, but I-TASSER uses a 3D grid simulation approach for hard regions.

2.2. A specific preliminary implementation

The general recursive modeling process can be implemented in a number of ways depending on the specific tools and methods selected for each step. For the recursive modeling in our MULTICOM protein modeling system, we used *buildali.pl* (i.e. *buildali.pl query.fasta*) in the HHSearch 1.5[35] calling PSI-BLAST to search a query protein sequence against the

NCBI Non-Redundant protein sequence database to collect homologous sequences. The *e*-value parameter (-*e*) of PSI-BLAST was set to 0.001. The query sequence together with its homologous sequences was used by *hhmake* of a profile-profile alignment tool HHSearch[35] to construct a Hidden Markov Model (HMM) as profile. We used *hhsearch* of the HHSearch[35] to search the query profile against our pre-built in-house template profile library. The profile–profile search returned a number of templates ranked by their significances (*e*-values) and generated alignments between the query protein and each template. The *e*-value threshold for selecting significant templates was set to 0.001. The alignment between the query and the most significant template was selected to decompose the query sequence into certain and uncertain regions. Generally, a long unaligned region of the query (e.g. >= 20 residues) was considered an uncertain region (see the regions of T0547 circled by red and green rectangle in Fig. 2 as an example).

We used the simplest protocol of Modeller 9v7[36] — *automodel* — to perform template-based modeling for certain regions with default parameter settings taking the query-template alignments generated by HHSearch and the template structures as input. The *automodel* protocol constructed structural conformations for aligned residues in certain regions and also automatically built loops for unaligned residues if they existed. If certain regions were covered by multiple significant templates, all of them were used by *automodel* to generate models according to their alignments with the same certain regions. Ten models were generated and the model with the minimum Modeller energy was chosen as the model of the certain regions. Modeller also generated conformations — most likely long extended chains — for uncertain regions, which were further folded by template-free modeling as follows.

To model unaligned/uncertain regions, we used a modified Rosetta 3.1 modeling protocol to do template-free (i.e. fragment assembly) modeling.[37] The modeling procedure for these regions was carried out in four steps. First, a query specific fragment library for the sequence was created using the *make_fragments* script included in the Rosetta software suite (command: make_fragments.pl -nojufo -noprof -nosam -xx aa -id 9999 *query.fasta*). This process identified and created 3- and 9-residue long fragments of the protein chain stored in two library files (e.g. *aa999909_05.200_v1_3*, *aa999903_05.200_v1_3*). Next, the template-based models were idealized using the idealize program from the Rosetta Modeling Suite (command: *idealize.linuxgccrelease -in:file:s query.pdb*). This ensured that bond lengths and angles contained in the models passed in from Modeller were compatible with the modeling protocols of Rosetta. A new idealized model file (*query.idealized.pdb*) was generated. Then a modified version of the AbinitioRelax protocol was called to perform fragment assembly on uncertain regions of the idealized model. For example, one command may look like “AbinitioRelax.linuxgccrelease -in:file:native query.idealized.pdb-database $ROSETTA_PATH/rosetta_database/ -in:file:frag9 aa999909_05.200_v1_3-in:file:frag3aa999903_05.200_v1_3-out:pdb -run:use_time_as-seed -out:nstruct 100 -abinitio:refine_range “*1–25*” *-abinitio:increase_cycles* “*0.3*” *-abinitio:start_native true*”, which aims to do fragment assembly on residues 1–25 of the model and produce 100 new refined models. The primary purpose of this protocol is to randomly replace portions (e.g. uncertain regions) of a protein model with fragment selected from the fragment library. After a replacement was made, the resulting structure was evaluated by Rosetta’s scoring function; and if the new conformation was more favorable, it was kept and otherwise rejected.[9] This process was repeated a number of times and then the final structure was refined using a full-atom force field which is part of the *AbinitioRelax* protocol. As shown in the command above, we modified the protocol (i.e. its C++ code) such that we could specify where fragment replacements could be applied in terms of range of residues. For example, if the first 30 residues of a query are uncertain, the range (1–30) can be passed into the modified *AbinitioRelax* to do fragment assembly on the first 30 or so residues without changing much of other certain regions. One caveat is that the actual region changed by

Rosetta can be 8 (or 16) residues larger than the input range because Rosetta can do a 9-mer fragment replacement starting from either end of the input range. Although conformation changes were limited to specified uncertain regions, the whole conformation including certain regions was used to calculate an energy change induced by a fragment replacement in the uncertain regions. Thus, this approach allowed for the unaligned/uncertain regions to be modeled in such a way that when fragments were replaced and scored, they were influenced by the certain/fixed regions. Finally, after the models had been updated by the *AbinitioRelax* protocol, they were compared to their form prior to modification by fragment based assembly. This last check was needed to ensure that only unaligned/uncertain regions were modified. It was required as occasionally the idealization or relaxation of a model caused changes outside of the specified unaligned/uncertain regions.

By specifying and limiting fragment replacements to uncertain regions (and possibly some short extensions into adjacent certain regions for the purpose of smoothly stitching the boundaries), the fragment assembly of the other regions can be influenced by the rigid regions because their conformations are considered during the assembly of fragments for uncertain regions. For instance, fragment replacements in uncertain regions that are energetically favored by certain regions are more likely to be accepted. Generally, *AbinitioRelax* was executed to generate 100 or more models with almost the exactly same conformation for the certain regions and likely different conformations for the uncertain regions. One model was selected by either ModelEvaluator[34,38] or APOLLO.[30] ModelEvaluator is a single model quality assessment tool that uses the structural features extracted from the model (e.g. secondary structure, relative solvent accessibility, and contact map) to predict its structural similarity with native structure in terms of GDT-TS score.[39] ModelEvaluator only requires two parameters (a query sequence file and a model file) as input to predict the score of a model. APOLLO has a pairwise structure comparison-based model evaluation tool that superposes each model with all other models to calculate their structural similarity scores (e.g. GDT-TS score or Root Mean Squared Distance) and uses the average score as the predicted quality score of the model. Based on model superposition, APOLLO can also predict local quality of a model.[30] APOLLO only requires as input a directory of containing model files and a list of model file names to be evaluated. The model of the query protein can then be decomposed into larger certain regions (the previous certain regions before template-free modeling *plus* newly formed certain regions of good quality produced by template-free modeling) and shrunk, smaller uncertain regions, which may be subjected to the next round of template-based and template-free modeling if necessary i.e. the conformation of enlarged certain regions will be used as new template by template-based modeling and that of shrunk uncertain regions will be refined by template-free modeling. The process stops if the overall quality of the model is acceptable (e.g. above a quality score threshold) or there is no uncertain region. The threshold of acceptable global quality score of ModelEvaluator and APOLLO was set to 0.5 and 0.4, respectively.

In addition to the main process described above, a simplified modeling process was applied to uncertain regions that were sufficiently long (e.g. >45 residues) to be stand-alone domains within a large multi-domain protein for time efficiency. Considering that a domain-level uncertain region can fold rather independently, our MULTICOM system excised it together with a few residues extension at its ends off from the query sequence to do modeling, and also invoked the template-free modeling to construct models for it, and then composed the selected model of this region with the remaining conformations of other regions into one model using the template-based modeling tool Modeller. The composition process simply used the conformations of all the regions as templates for Modeller to generate a combined full-length model. The main reason of applying this simplified modeling to uncertain domains was to speed up the template-free modeling process in practice as the current template-free modeling method (e.g. Rosetta) was very slow on large proteins (e.g. > 300 residues) with multiple domains. Folding a single smaller domain alone with the template-free modeling can be much faster and thus can explore larger conformation space in limited time than running a full template-free modeling in the context of the whole conformation of a large protein. Furthermore, in most cases, the template-based modeling can readily compose the conformations of certain and uncertain multiple domains into one cohesive model. However, the relative orientations between some domains whose models were not constructed in the same step might not be assembled very accurately by the less exploratory template-based composition process. The problem was often alleviated because large certain domains modeled together tend to have largely correct domain orientations to be used as scaffold to restrict the orientation of uncertain template-free domains (see case 1 in Sec. 3 for an example). The problem might be further addressed by orienting domain models generated in this simplified protocol according to a smaller number of models generated by the slower domain-level template-free modeling in the context of the conformation of the whole protein. Moreover, as more and more computing power is available and template-free modeling becomes faster, large-scale running of domain-level template-free modeling with the whole conformation of a large multi-domain protein will become more practical, likely leading to the better domain assembly. These options will be extensively explored in the future as we continue to improve the implementation of the protocol.

3. Results

The recursive modeling approach was implemented within our MULTICOM system as four automated protein structure prediction servers (i.e. MULTICOM-CLUSTER, MULTICOM-REFINE, MULTICOM-NOVEL, and MULTICOM-CONSTRUCT), which mainly differed in model ranking and combination.[40] The MULTICOM system was blindly tested during the 9th Critical Assessment of Techniques for Protein Structure Prediction (CASP9), 2010.[19] It showed its promise by improving the quality of protein modeling in a majority of hard cases where both template-based and template-free modeling could be applied. Here we discuss how the recursive modeling improved structure prediction in three typical situations.

Case 1: Recursive modeling enhances the modeling of large, complicated, multi-domain proteins

Instead of improving the uncertain regions of a single domain, here, recursive modeling can synergistically model several template-based and template-free domains entangled together. The decomposition of a query protein into multiple regions can help solve the complicated domain architecture involving discontinuous segments and domain insertions.

The CASP9 target T0547 is a good example illustrating this case. This protein has a very complicated domain architecture composed of four domains as illustrated in Fig. 2(a). The first template-based domain has three discontinuous segments interrupted by two inserted domains — one template-based domain (i.e. domain 2) and one template-free domain (i.e. domain 3). The third fragment of domain 1 is joined by the fourth template-free domain. Traditional template-based modeling alone will fail on the two template-free domains, and template-free modeling alone simply cannot handle such a large protein with such complicated domain architecture. However the region decomposition approach used by recursive modeling can successfully identify the two template-based domains and template-free domains and compose them together. Figure 2 shows that two disjoint fragments of T0547 were aligned with one template 1TWI (chain A), which was considered a certain region. The entire aligned region was modeled by MULTICOM-REFINE based on the structure of template 1TWI using template-based modeling, which was better than modeling the two disjoint parts separately using a traditional domain-cutting strategy. The latter would not be able to model the three disjoint fragments of the first domain. Thus the "region" concept used in recursive modeling is a broader modeling-oriented concept, which may correspond to a part of a domain, one domain or even multiple domains and can even span discontinuous sequence fragments. The two unaligned/uncertain regions were modeled by MULTICOM-REFINE with the template-free method. Then the three components were composed into one model using the template-based modeling again (Fig. 3). In this case, the large template-based component consisting of two domains served as a scaffold to dock two *ab initio* domains with other domains. This is more effective than traditional approaches which would randomly assemble these domains without considering their interactions at all.

It turned out that all four domains generated by MULTICOM-REFINE's recursive modeling procedure were ranked among the high-quality server models in CASP9 (Fig. 3). This example clearly demonstrates that the recursive modeling protocol can effectively decompose a large protein to reduce modeling complexity, resulting in better modeling quality. In addition to this example, we found that recursive modeling could also improve modeling on other targets composed of multiple template-based and template-free domains (e.g. T0543, T0571).

Case 2: Recursive modeling improves ab initio modeling by starting from a very weak, largely incorrect template that contains a few fragments close to the native structure

For some very hard targets, only a number of highly uncertain templates can be found; and these templates may only have partially correct template conformations (e.g. just one or two secondary structure elements). In this case, a template-free extension from the partially correct core secondary element(s) may still improve the quality of modeling. Target T0616 (107 residue long) was a TBM/FM example, in which some analogous templates existed but were not likely be found or used by any server predictor. Our server MULTI-COM-REFINE found a partial template covering the last ~80 residues (Fig. 4), which at most had part of a helix matching the native structure. The recursive modeling method initially built a model for the aligned regions from the template [Fig. 5(a)], and then extended the unaligned N-terminal region using template-free modeling [Fig. 5(b)]. As shown in Fig. 5, starting from the partial central helix, the template-free modeling on the first 31 residues (i.e. unaligned 21 residues plus a short extension) was able to extend the partially correct region to a structure closer to the native structure. The model was the best CASP9 server model submitted for this target.

Case 3: The recursive modeling procedure improves template-based modeling by fixing uncertain terminal regions

In this case, a large portion of a query protein can be aligned confidently to one or more partial templates, while leaving some parts of the query unaligned (e.g. front/back tails, partially unfolded internal helices/strands/loops). The recursive modeling models template-based regions firstly and then uses them as additional restraints for template-free modeling to improve the modeling of unaligned/uncertain regions iteratively. The CASP9 target T0539 is a good example, where the whole target except for the ~20 N-terminal residues can be aligned to a few templates (Fig. 6).

Using the conformation core generated from the template information as restraints, the recursive modeling method in the MULTICOM-CONSTRUCT server correctly reconstructed the loop-helix-loop structure of the uncertain front region and its interaction with the core as shown in Fig. 7. The GDT-TS score[39] was increased by 14% from 0.64 to 0.73. There were quite a few other similar CASP9 targets (e.g. T0568, T0574, T0592, T0593, T0596, T0597, T0632, and T0636) whose uncertain regions could be improved by the recursive modeling procedure. However, the improvement may not always be reflected in the GDT-TS scores according to CASP9 assessment because in CASP some uncertain regions are often removed before assessment. Overall, according to our assessment, recursive modeling generally improves the quality of modeling in this situation.

The three typical cases above demonstrate that recursive modeling can readily integrate template-based and template-free modeling to improve tertiary structure prediction of both single- and multi-domain proteins. It can also be easily implemented using existing or slightly modified alignment and model generation tools. However, it is worth pointing out that recursive modeling may not realize its best potential if region decomposition deviates too far away from the true boundaries between certain and uncertain regions. For instance, modeling one half of an uncertain region (e.g. template-free/*ab initio* domain) using template-free modeling and the other half by an incorrect template usually leads to a poor prediction as evidenced by our predictions for target T0534. In this situation, the GDT-TS score of the template-free region is often low (e.g. ~0.2). Nevertheless, the alignment-based region decomposition is generally robust against some residue shifts. A slightly more conservative region decomposition approach, that is to say only classifying very confident regions into certain regions in the beginning, seems to work better.

4. Conclusion

We have described a general recursive protein modeling approach which can effectively integrate template-based and template-free modeling to improve protein modeling quality as demonstrated by its successful performance in the CASP9 experiments. This approach can often decompose a large, complicated modeling problem into several smaller and simpler modeling problems, which can be more readily addressed by synergistically integrating template-based and template-free modeling. Furthermore, the solutions to the smaller problems can be composed together to solve a larger, more complex modeling problem. In general, this “divide and conquer” strategy can improve both the quality and speed of protein structure modeling. According to this strategy, it is not necessary to divide protein modeling into two distinct approaches; instead, it can be viewed as a full spectrum of modeling based on an arbitrary percentage of template-based or template-free modeling. In the future, we plan to improve the modeling process by designing more robust methods for the detection of certain and uncertain regions based on sequence alignments or the local quality of a model. We also plan to implement more effective ways (e.g. conformation ensemble-like approaches[41]) to use template information to guide template-free modeling or to use template-free modeling to extend template-based regions. We expect to test the new methods in both the rolling CASP experiment (CASP ROLL) and the upcoming 10th CASP experiment (CASP10).

Acknowledgments

The work was partially supported by an NIH grant R01GM093123 to JC, a NLM fellowship to JE, and the Shumaker fellowship to ZW.

Biography

Jianlin Cheng is an Assistant Professor of Bioinformatics in the Computer Science Department at the University of Missouri, Columbia. He earned his Ph.D. in Computer Science at the University of California, Irvine, in 2006. His current research is focused on bioinformatics, computational biology, machine learning and data mining.

Jesse Eickholt received his B.S. degrees in Mathematics and Computer Science from the University of Missouri in 2001 and a Masters in Applied Mathematics in 2010. He is currently pursuing his Ph.D. in Computer Science at the University of Missouri. His research interests include bioinformatics, machine learning and deep learning.

Zheng Wang is a Ph.D. candidate in the Computer Science Department at University of Missouri, USA (2008–present). He obtained his Master of Computer Science degree at University of New Brunswick, Canada in 2007 and a Bachelor degree in Management Information System from Shandong University of Finance and Economics, China, in 2004. His research interests include protein structure prediction, protein function prediction, protein model quality assessment, cancer epigenomics, genome annotation, and species phylogeny inference.

Xin Deng is a Ph.D. candidate in the Computer Science Department at the University of Missouri, Columbia, since 2008. Her current research is focused on computational biology and machine learning, such as developing computational methods in multiple sequence alignment, fold recognition based on similarity network, and protein structure prediction.

References

1. Baker D, Sali A. Protein structure prediction and structural genomics. Science. 2001; 294:93–96. [PubMed: 11588250]
2. Fox BG, Goulding C, Malkowski MG, Stewart L, Deacon A. Structural genomics: From genes to structures with valuable materials and many questions in between. Nat Methods. 2008; 5:129–132. [PubMed: 18235432]
3. Benson D, Boguski M, Lipman D, Ostell J, Ouellette B, Rapp B, Wheeler D. GenBank. Nucleic Acids Res. 1999; 27:12–17. [PubMed: 9847132]
4. Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov I, Bourne P. The protein data bank. Nucleic Acids Res. 2000; 28:235–242. [PubMed: 10592235]
5. Cheng J. A multi-template combination algorithm for protein comparative modeling. BMC Struct Biol. 2008; 8:18. [PubMed: 18366648]
6. Zhang Y, Skolnick J. Automated structure prediction of weakly homologous proteins on a genomic scale. Proc Natl Acad Sci. 2004; 101:7594–7599. [PubMed: 15126668]
7. Sali, A.; Blundell, T. Comparative protein modelling by satisfaction of spatial restraints. In: Bohr, H.; Brunak, S., editors. Protein Structure by Distance Analysis. IOS Press; Amsterdam: 1994. p. 64-86.
8. Jones D, McGuffin L. Assembling novel protein folds from super-secondary structural fragments. Proteins: Struct Function Bioinformatics. 2003; 53:480–485.
9. Simons K, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol. 1997; 268:209–225. [PubMed: 9149153]
10. DePristo MA, de Bakker PIW, Lovell SC, Blundell TL. *Ab initio* construction of polypeptide fragments: Efficient generation of accurate, representative ensembles. Proteins. 2002; 51:41–55. [PubMed: 12596262]
11. Coutsias EA, Seok C, Jacobson MP, Dill K. A kinematic view of loop closure. J Comput Chem. 2004; 25:510–528. [PubMed: 14735570]
12. Canutescu AA, Dunbrack RL Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci. 2003; 12:963–972. [PubMed: 12717019]
13. Liu P, Zhu F, Rassokhin DN. Agrafiotis DK, A self-organizing algorithm for modeling protein loops. PLOS Comput Biol. 2009; 5:e1000478. [PubMed: 19696883]
14. Lee J, Lee D, Park H, Coutsias EA, Seok C. Protein loop modeling by using fragment assembly and analytical loop closure. Proteins. 2010; 78(16):3428–3436. [PubMed: 20872556]
15. Yang Y, Zhou Y. *Ab initio* folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions. Protein Sci. 2008; 17:1212–1219. [PubMed: 18469178]
16. Boczko E, Brooks 3rd C. First-principles calculation of the folding free energy of a threehelix bundle protein. Science. 1995; 269:393–396. [PubMed: 7618103]

17. Dill K. Dominant forces in protein folding. Biochemistry. 1990; 29:7133–7155. [PubMed: 2207096]
18. Wang Z, Eickholt J, Cheng J. MULTICOM: A multi-level combination approach to protein structure prediction and its assessments in CASP8. Bioinformatics. 2010; 26:882–888. [PubMed: 20150411]
19. Moult J, Fidelis K, Kryshtafovych A, Tramontano A. Critical assessment of methods of protein structure prediction (CASP) — round IX. Proteins. 2011; 79(S10):1–5. [PubMed: 21997831]
20. Tress ML, Ezkurdia I, Richardson JS. Target domain definition and classification in CASP8. Proteins. 2009; 77(S9):10–17. [PubMed: 19603487]
21. Kinch LN, Shi S, Cheng H, Cong Q, Pei J, Mariani V, Schwede T, Grishin NV. CASP9 target classification. Proteins. 2011; 79(S10):21–36. [PubMed: 21997778]
22. Roy A, Kucukural A, Zhang Y. I-TASSER: A unified platform for automated protein structure and function prediction. Nat Protocols. 2010; 5:725–738.
23. Altschul S, et al. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res. 1997; 25:3389–3402. [PubMed: 9254694]
24. Karplus K, et al. Hidden Markov models for detecting remote protein homologies. Bioinformatics. 1998; 14:846–846. [PubMed: 9927713]
25. Eddy S. Profile hidden Markov models. Bioinformatics. 1998; 14:755–763. [PubMed: 9918945]
26. Madera M, Gough J. A comparison of profile hidden markov model procedures for remote homology detection. Nucleic Acids Res. 2002; 30:4321–4328. [PubMed: 12364612]
27. Sadreyev R, Grishin N. COMPASS: A tool for comparison of multiple protein alignments with assessment of statistical significance. J Mol Biol. 2003; 326:317–336. [PubMed: 12547212]
28. Söding J. Protein homology detection by HMM–HMM comparison. Bioinformatics. 2005; 21:951–960. [PubMed: 15531603]
29. Benkert P, Tosatto S, Schomburg D. QMEAN: A comprehensive scoring function for model quality assessment. Proteins. 2008; 71:261–277. [PubMed: 17932912]
30. Cheng J, Wang Z, Tegge A, Eickholt J. Prediction of global and local quality of CASP8 models by MULTICOM series. Proteins. 2009; 77:181–184. [PubMed: 19544564]
31. Lundstrom J, Rychlewski L, Bujnicki J, Elofsson A. Pcons: A neural-network-based consensus predictor that improves fold recognition. Protein Sci. 2001; 10:2354–2362. [PubMed: 11604541]
32. McGuffin L. Prediction of global and local model quality in CASP8 using the ModFOLD server. Proteins: Structure, Function, and Bioinformatics. 2009; 77:185–190.
33. Pettitt C, McGuffin L, Jones D. Improving sequence-based fold recognition by using 3D model quality assessment. Bioinformatics. 2005; 21:3509–3515. [PubMed: 15955780]
34. Wang Z, Tegge A, Cheng J. Evaluating the absolute quality of a single protein model using structural features and support vector machines. Proteins. 2008; 75:638–647. [PubMed: 19004001]
35. Soding J, Biegert A, Lupas A. The HHpred interactive server for protein homology detection and structure prediction. Nucleic Acids Res. 2005; 33:W244–W248. [PubMed: 15980461]
36. Fiser A, Sali A. Modeller: Generation and refinement of homology-based protein structure models. Methods Enzymol. 2003; 374:461–491. [PubMed: 14696385]
37. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, et al. ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol. 2011; 487:545–474. [PubMed: 21187238]
38. Wang Z, Eickholt J, Cheng J. APOLLO: A quality assessment service for single and multiple protein models. Bioinformatics. 2011; 27(12):1715–1716. [PubMed: 21546397]
39. Zemla A. LGA: A method for finding 3D similarities in protein structures. Nucleic Acids Res. 2003; 31:3370–3374. [PubMed: 12824330]
40. Cheng J, Wang Z, Eickholt J. Integrated prediction of protein tertiary structure by MULTICOM predictors. CASP9 Proc. 2009; 9:173–175.
41. Eickholt J, Wang Z, Cheng J. A conformation ensemble approach to protein contact prediction. BMC Struct Biol. 2011; 11:38. [PubMed: 21989082]

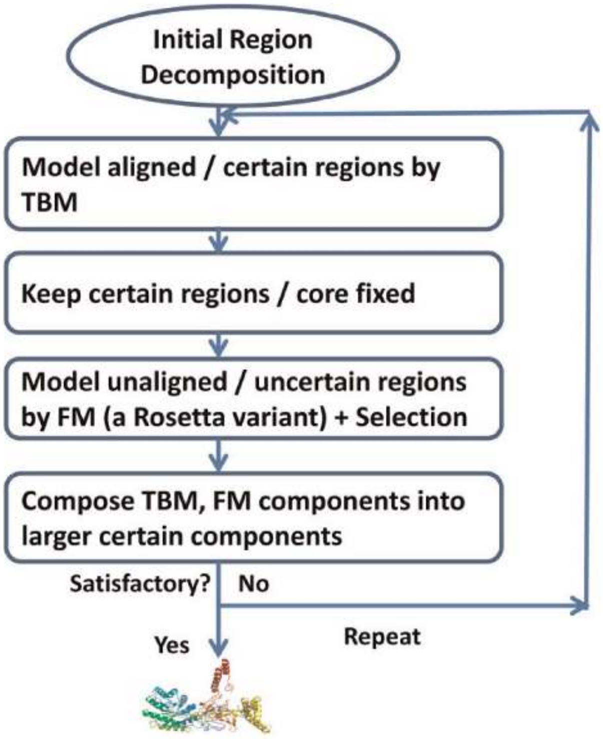
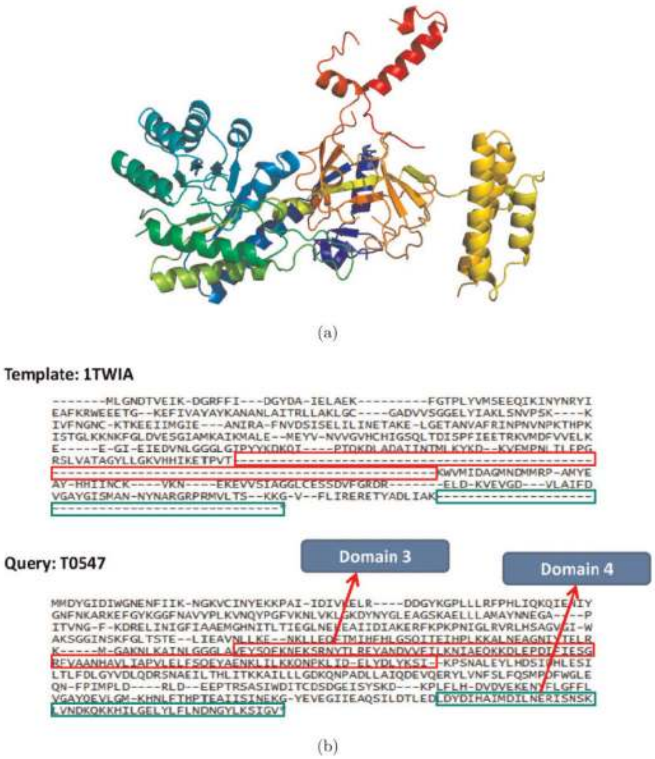

Initial Region Decomposition
Model aligned / certain regions by TBM
Keep certain regions / core fixed
Model unaligned / uncertain regions by FM (a Rosetta variant) + Selection
Compose TBM, FM components into larger certain components
Satisfactory?
No
Repeat
Yes

Fig. 1.

The flowchart of the recursive protein modeling procedure. TBM and FM denote template-based modeling and template-free modeling, respectively.



Template: 1TWIA

```
-------MLGNDTVEIK-DGRFFI---DGYDA-IELAEK----------FGTPLYVMSEEQIKINYNRYI
EAFKRWEEETG--KEFIVAYAYKANANLAITRLLAKLGC----GADVVSGGELYIAKLSNVPSK----K
IVFNGNC-KTKEEIIMGIE---ANIRA-FNVDSISELILINETAKE-LGETANVAFRINPNVNPKTHPK
ISTGLKKNKFGLDVESGIAMKAIKMALE--MEYV-NVVGVHCHIGSQLTDISPFIEETRKVMDFVVELK
E-----E-GI-EIEDVNLGGGLGIPYYKDKQI----PTDKDLADAIINTMLKYKD--KVEMPNLILEPG
RSLVATAGYLLGKVHHIKETPVT-----------------------------------------------
--------------------------------------------------KWVMIDAGMNDMMRP-AMYE
AY-HHIINCK----VKN----EKEVVSIAGGLCESSDVFGRDR--------ELD-KVEVGD---VLAIFD
VGAYGISMAN-NYNARGRPRMVLTS--KKG-V--FLIRERETYADLIAK---------------------
-----------------------------
```

Query: T0547

```
MMDYGIDIWGNENFIIK-NGKVCINYEKKPAI-IDIVKELR----DDGYKGPLLLRFPHLIQKQIENIY
GNFNKARKEFGYKGGFNAVYPLKVNQYPGFVKNLVKLGKDYNYGLEAGSKAELLLAMAYNNEGA----P
ITVNG-F-KDRELINIGFIAAEMGHNITLTIEGLNEKEAIIDIAKERFKPKPNIGLRVRLHSAGVGI-W
AKSGGINSKFGLTSTE--LIEAVNLLKE--NKLLEQETMIHEHLGSQITEIHPLKKALNEAGNIYTELR
K-----M-GAKNLKAINLGGGLAVEYSQEKNEKSRNYTLREYANDVVEILKNIAEQKKDLEPDIFIESG
REVAANHAVLIAPVLELFSQEYAENKLILKKQNPKLID-ELYDLYKSI--KPSNALEYLHDSIQHLESI
LTLFDLGYVDLQDRSNAEILTHLITKKAILLLGDKQNPADLLAIQDEVQERYLVNFSLFQSMPDFWGLE
QN-FPIMPLD----RLD--EEPTRSASIWDITCDSDGEISYSKD---KPLFLH-DVDVEKENYFLGFFL
VGAYQEVLGM-KHNLFTHPTEAIISINEKG-YEVEGIIEAQSILDTLEDLDYDIHAIMDILNERISNSK
LVNDKQKKHILGELYLFLNDNGYLKSIGV
```

Fig. 2.

Domain architecture of CASP target T0547. (a) The experimental structure of T0547; (b) The region decomposition of target T0547 based on its sequence alignment with a template. T0457 was aligned with template 1TWI (chain A) by HHSearch. Red and green rectangles delineate the unaligned regions, which correspond to template-free domains 3 and 4 of T0547. 1TWI covers domains 1 and 2 of T0547.

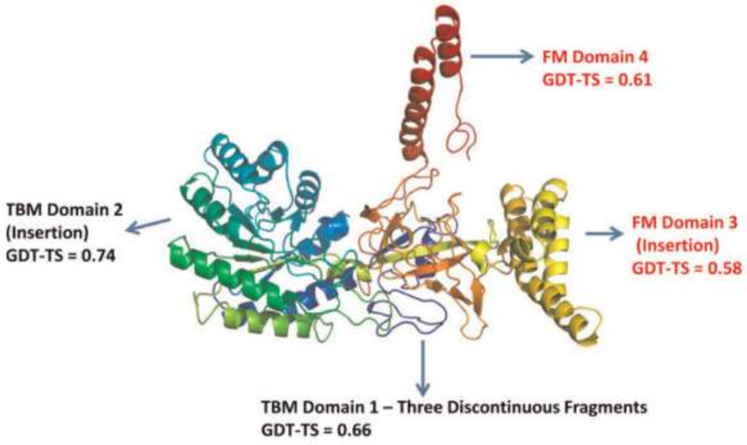
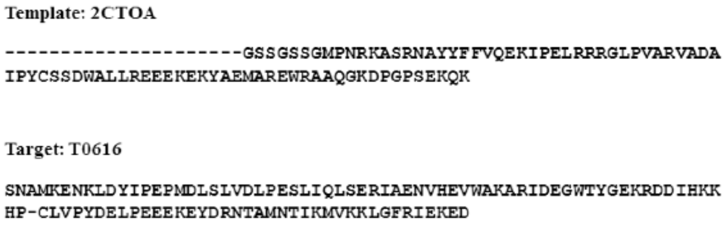


Fig. 3.

The model predicted by MULTICOM-REFINE for target T0547. The template-based domain 1 has three discontinuous segments flanked by template-based domain 2 and template-free domain 3. Domain 4 is a template-free domain. According to the GDT-TS scores, these four domains are among the top server models in CASP9.



```
Template: 2CTOA

---------------------GSSGSSGMPNRKASRNAYYFFVQEKIPELRRRGLPVARVADA
IPYCSSDWALLREEEKEKYAEMAREWRAAQGKDPGPSEKQK


Target: T0616

SNAMKENKLDYIPEPMDLSLVDLPESLIQLSERIAENVHEVWAKARIDEGWTYGEKRDDIHKK
HP-CLVPYDELPEEEKEYDRNTAMNTIKMVKKLGFRIEKED
```

Fig. 4.

Alignment between CASP9 target T0616 and a structure template (PDB code: 2CTO; chain A).

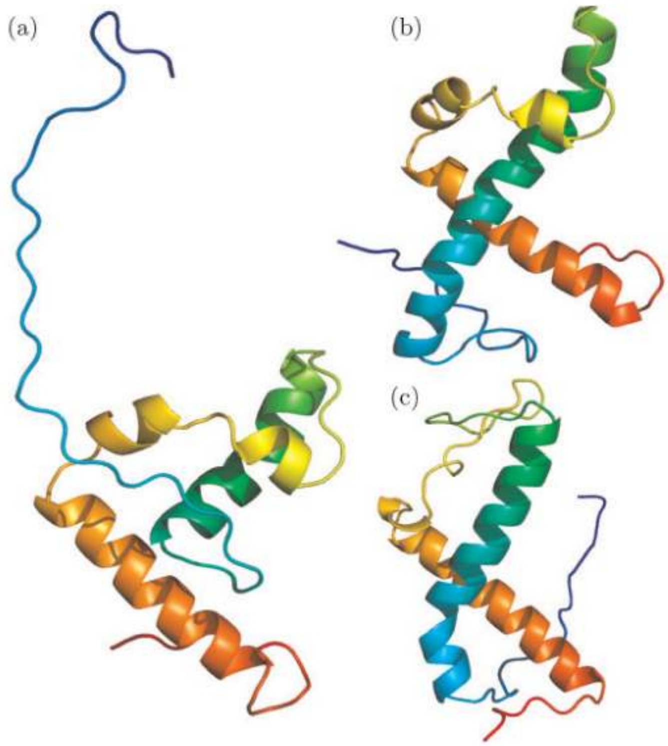
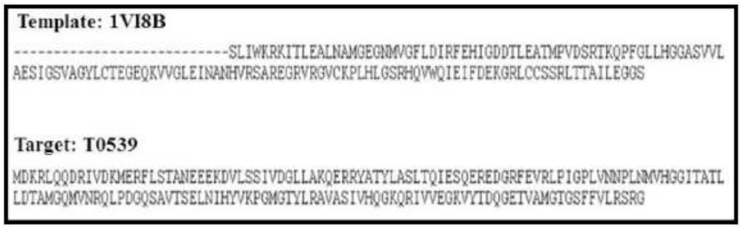


Fig. 5.

As example of recursive modeling on CASP target T0616. (a) a model generated solely by template-based modeling (GDT-TS = 0.34); (b) a model integrating both template-based modeling (GDT-TS = 0.39); (c) the native experimental structure.



Template: 1VI8B

```
---------------------------SLIWKRKITLEALNAMGEGNMVGFLDIRFEHIGDDTLEATMPVDSRTKQPFGLLHGGASVVL
AESIGSVAGYLCTEGEQKVVGLEINANHVRSAREGRVRGVCKPLHLGSRHQVWQIEIFDEKGRLCCSSRLTTAILEGGS
```

Target: T0539

```
MDKRLQQDRIVDKMERFLSTANEEEKDVLSSIVDGLLAKQERRYATYLASLTQIESQEREDGRFEVRLPIGPLVNNPLNMVHGGITATL
LDTAMGQMVNRQLPDGQSAVTSELNIHYVKPGMGTYLRAVASIVHQGKQRIVVEGKVYTDQGETVAMGTGSFFVLRSRG
```

Fig. 6.

Alignment between CASP9 target T0539 and a good homologous template (PDB code: 1VI8; chain B). The first ~20 N-terminal residues of the target that do not have alignment cannot be modeled well using template-based modeling alone.

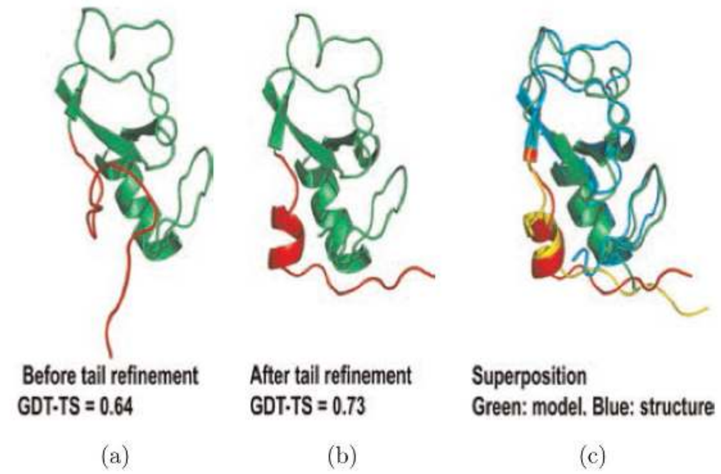


Fig. 7.

An example of applying recursive modeling to CASP9 target T0539 by the server MULTICOM-CONSTRUCT. (a) Template-based modeling is used to model the aligned/ certain part (green) of T0539 while leaving the unaligned/uncertain region (red) free; the GDT-TS score of the model is 0.64. (b) Fragment assembly is used to model the uncertain region (red) while keeping template-based core (green) fixed; the GDT-TS score of the model composed of both template-based and template-free components is 0.73, 14% higher than the model in (a). (c) The superposition of the composed model (green + red) with the experimental structure (blue + yellow), showing that the uncertain tail (i.e. loop-helix-loop) is well packed with the template-based core in the model. Particularly the helix–helix interaction is reproduced in the composed model, which may not be possible by using either template-based modeling or template-free modeling independently.