

# PARALLEL COMPUTING IN PROTEIN STRUCTURE TOPOLOGY DETERMINATION

Weitao Sun<sup>1,2</sup>, Saeed Al-Haj<sup>1</sup>, Jing He<sup>1,\*</sup>

<sup>1</sup>*Department of Computer Science, New Mexico State University, Las Cruces, NM 88003*

<sup>2</sup>*Zhou Pei-Yuan Center for Applied Mathematics, Tsinghua University, Beijing, 100084, China*

## ABSTRACT

The knowledge of 3-dimensional virus structures is essential in understanding the mechanism of viral pathogenesis. It also provides insights to the stabilizing mechanisms of a nano-sized particle, since many viruses are less than 100 nanometers in diameter. This paper reports the results towards the development of a scalable parallel code for structural prediction of virus particles through ab initio structure prediction using geometrical constraints. One of the critical steps in computational derivation of a protein structure is to reduce the huge number of topologies of the secondary structures, such as helices and strands, of a protein chain. In this paper, we study a particular question emerged from experimental data that carry the geometrical relationship of the secondary structures. We explored the question if the native topology is likely to be identified among a large set of all possible topologies. The secondary structure topology in this paper refers to the order and the directionality of the secondary structures. For a given protein sequence  $N$  helices and  $M$   $\beta$ -strands, the number of possible secondary structure topology is  $(N!2^N)(M!2^M)$ , a huge number to compute even when  $N$  and  $M$  are small numbers. We have developed a computational method and its parallel code to generate all the possible topologies and to evaluate the energy of each topology. By mutating residue side chains of the secondary

structures, connection orders are switched and a new topology is created. The large number of permutations is partitioned and distributed to different CPUs. We compared the speedup between two approaches of distributing the work: the even distribution and the dynamic distribution. Our current parallel algorithms can handle the computation when  $N$  is less than 7 on a small scale cluster for testing the algorithm. A large cluster is needed to extend the scale of computation.

## 1. INTRODUCTION

In the recent years, emerging viruses have caused great concerns around the world. The knowledge of 3-dimensional (3D) structure of the viruses is essential for understanding the pathogenesis of the viruses and antiviral drug design. In addition to the biological interest of viruses, viruses also have strong engineering potential. Many viruses are naturally available nano-sized particles that are self assembled by many copies of its proteins. It has been an emerging research area on how to engineer the virus particle so that the mutated virus particles can be a safe vehicle for a specific delivery of interest.

Protein structure prediction, also known as the protein folding problem, has been attracting scientists

---

\* Corresponding Author

from many disciplines in the past five decades. It is believed that the 3D structure of a protein is determined by its amino acid sequence. Although the prediction of the secondary structures, such as helices and strands, generally has accuracy of 70%-80% (Jones 1999; Pollastri, Przybylski et al. 2002), the tertiary structure prediction is still challenging. There has been an effort to combine the experimental data with the computational structure prediction to derive the 3D structure of the proteins (Topf, Baker et al. 2005; Wu, Chen et al. 2005; Baker, Jiang et al. 2006; Topf, Baker et al. 2006; Lu, He et al. 2008).

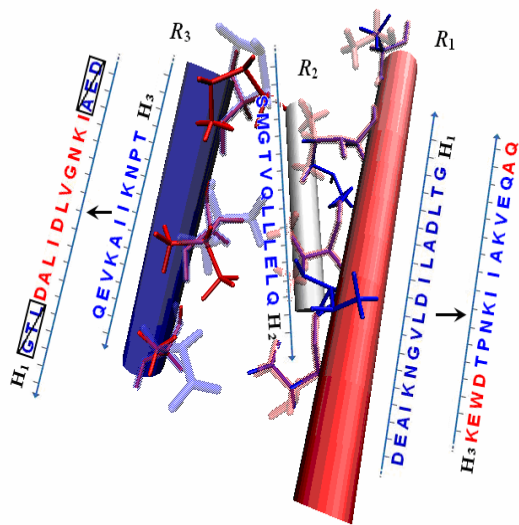
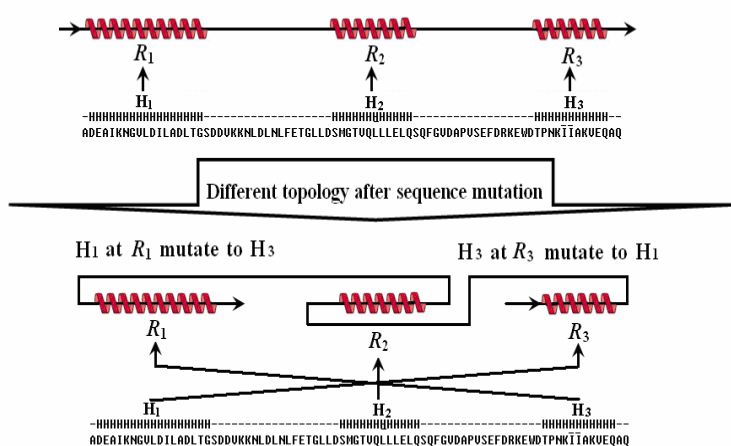


Figure 1: Mapping sequence segments to skeletons of protein with PDB ID 1DV5.

Electron cryo-microscopy (Cryo-EM) technique has been widely used by virologists to study the virus structures, due to its ability to handle large molecular complexes such as viruses. Currently, this technique can produce the electron density map of viruses up to 5 to 10 Å resolution (Böttcher, Wynne et al. 1997; Conway, Cheng et al. 1997) (Zhou, Dougherty et al. 2000; Ludtke SJ 2004). Since the density map at this resolution does not provide enough information to derive the entire protein chain, the emerging research direction is to predict the 3D structure using the density map as geometrical constraints. Although the entire chain of the protein backbone can not be determined from a density map at 5 to 10 Å resolution, partial information about the chain can be computationally detected (Jiang, Baker et

al. 2001; Kong and Ma 2003; Kong, Zhang et al. 2004; Del Palu, He et al. 2006). The skeleton (i.e. sticks in Figure 1) of the secondary structures, such as helices and  $\beta$ -sheets, can be detected in a protein density map at such resolution. However, it is not known which segments of the protein amino acid sequence are related to the secondary structures detected from the density map (He, Lu et al. 2004; Wu, Chen et al. 2005; Lu, He et al. 2008). One way to map the segments of the protein sequence to the detected secondary structures is to use the segments suggested by the secondary structure prediction tools such as PSIPRED and PHD. These tools



can predict the sequence segments of helices and  $\beta$ -strands to about 80% of accuracy (Jones 1999; Przybylski and Rost 2002). For a protein with  $N$  helices and  $M$   $\beta$ -strands, the total number of ways to map the sequence segments and the skeletons is  $(N!2^N)(M!2^M)$ . This is due to the fact that there are  $N!$  permutations to map  $N$  segments to  $N$  skeleton helices, and two directions to associate each sequence segment with a skeleton helix. The topology of secondary structures in this paper refers to the order and directionality of the secondary structures. We describe here the parallel work we developed to generate and to evaluate all possible topologies of the secondary structures. The results suggest that the native topology has near minimum contact energy among all possible topologies.

## 2. MATERIALS AND METHODS

### 2.1 Secondary structure topology generation

We randomly selected fifty-one proteins from the Protein Data Bank that satisfy the following criteria: (1) has single domain; (2) has 1.5 Å or better resolution; (3) share less than 30% sequence similarity; (4) has less than 8 secondary structures, due to the amount of computation. For each of the proteins, we generated all the possible topologies for their secondary structures. Due to the size limitation of the paper, we show the result of eight of the fifty-one tested proteins in this paper. The central question of this work is whether the native topology is among a small set of topologies that are most comfortable in terms of energy. Each topology differs from another by either the mapping of the sequence segment or the direction of the segment with respect to the skeletons. We used the location of the  $\alpha$ -carbon atoms of the secondary structures to represent the location of the skeletons. By replacing the side chain of the amino acids on the skeletons, one topology is changed to another topology. The mutated topology will then be optimized for their side chain conformations to finish the generation of a new topology. Figure 1 illustrates the mapping between the sequence segments  $H_1, H_2, H_3$ , and the skeleton  $R_1, R_2, R_3$ . Two possible topologies are given in Figure 1. When the length of a sequence segment does not match the length of the skeleton, the length of the skeleton is used as a reference. In such cases, the segment is either truncated (crossed out boxes in Figure 1 left) or padded at the two ends. During the optimization of the side chains, simulated annealing was used to select the side chain conformation from a library of rotamers.

### 2.2 Parallel Implementations

We implemented the topology generation in two parallel schemes: the static and the dynamic. In the static model, the total number of permutations ( $nperm$ ) are attempted to be distributed evenly among the processors ( $mpi\_np$ ). The list of all the permutation jobs is shown in

Figure 2A. Processor 1 will be given job 1 and generate the work of job 4 when job 1 is done (Figure 2A). Each processor will call *next()* to generate the next piece of job (algorithm in this section). The advantage of this assignment is that each processor knows what the next job is and will generate by itself when needed. The communication among the processors is minimized since there is no adjustment in the work assignment. The disadvantage of the static assignment is that the different jobs may not have equal amount of work, and hence some processors finish earlier and wait for the last processor to finish. This is particularly bad when certain nodes have load from other users.

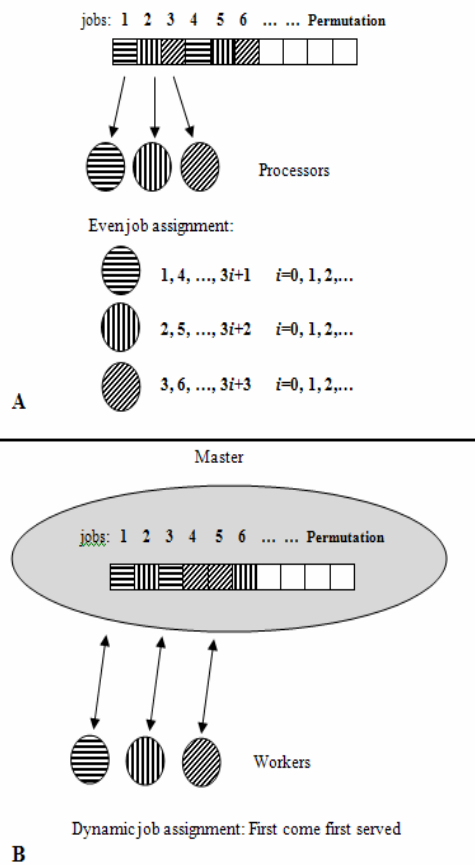


Figure 2: Two models of work distribution. A) static distribution method, B) dynamic distribution method.

#### Algorithm for a processor

Assigning initial jobs for each processor.

For processors from 0 to  $mpi\_np-1$

1. If this is the 1st processor, assign the first permutation to the current processor;

2. Otherwise, call sub-function `next( )` to create the next permutation;

(a) If `next( )` succeed, which means there is still another new permutation, assign it to current processor.

(b) Otherwise, there are no more new permutations. Set current processor index ID as the total loop number and exit the loop.

End loop

After the processors get the initial jobs, the following algorithm is run for each processor to work through all the assigned permutations ( $nperm$ ).

Go through all permutation jobs for current processor:

Loop of permutations, from 0 to  $nperm-1$ .

1. If current permutation index  $k$  is not less than  $mpi\_np$ , which means that the initial permutation has been finished in this processor, call sub-function `next( )` for  $mpi\_np$  times to create a new sequence permutation.

(a) If `next( )` succeed, a new permutation is created for this processor.

(b) Otherwise, there are no more permutation jobs for this processor. Then exit the loop for this processor.

2. Otherwise, current permutation index satisfies  $k < mpi\_np$  and it is the known initial permutation for this processor. There are no need to call `next( )` to create new permutation.

3. Mapping sequence segments to skeletons according to current sequence permutation.

4. Relax protein residue sidechains to avoid spatial overlaps.

5. Calculate contact energy for current protein structure. Compare this energy with native energy and increase number counter if current energy is lower than native one.

End loop

At last, collect computation results from all processors by adding the number of topologies with lower than native contact energy.

We also implemented a dynamic scheme to assign the work. The dynamic approach has a master who manages the assignment. Each processor asks for a job from the master who determines dynamically what job to be assigned. In order to get each processor as busy as possible, the master can send out multiple pieces of jobs depending on how many number of jobs are left in the pool. For example, the master sends out three pieces of job when the ratio between the number of the jobs left and the number of processors is greater than or equal to three. Then it sends a reduced amount of work when the leftover job pool is smaller. A typical set of data that is sent from a master to a slave is an integer array whose number of elements is the same as the number of secondary structures in the protein. This data are generated by the master by calling `next( )`, a function used by the individual processor in the static model.

### 3. NUMERICAL RESULTS

Table 1 shows the results from eight of the fifty-one proteins we tested. The name, the length, the number of helices and the number of  $\beta$ -strands it contains are shown for each protein in the first 4 columns. Table 1 suggests that among all the number of topologies (shown in the 5<sup>th</sup> column), only a small subset of them (the 6<sup>th</sup> column) has topologies more comfortable than the native topology. In the case of protein 1F1F, only 0.08% of the 46080 different topologies have lower contact energy (more comfortable) than that of the native topology. The clock time is shown for the 4, 8, 16 and 32 processors when the static model is used. The clock time starts at the initial preparation of work to the step before the `mpi_reduce( )` call that collects the results from different processors. Each node has 2 quad core Xeon 5335 processors, and 16GB of RAM with MPICH2 installed.

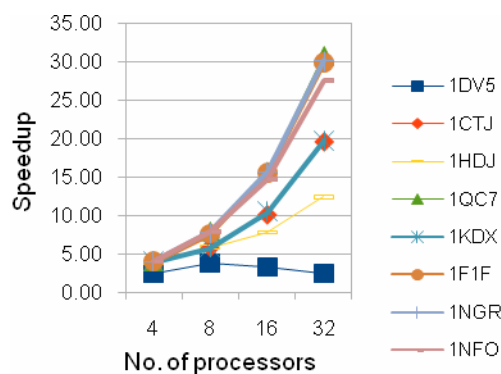
Tale 1. The total number of topologies, the comfortable topologies and the parallel time (Static Model).

PDB ID	$N_{AA}$	$N_a$	$N_b$	$N_{Mut}$	Pct <sub>eff</sub>	Total time (second)			
						4-Processor	8-Processor	16-Processor	32-Processor
1DV5	80	3	0	48	4.17%	5.186	4.219	3.1285	4.079
1HDJ	77	4	0	384	0.26%	23.0705	15.28	11.35	7.111
1CTJ	89	5	0	3968	0.55%	325.31	222.037	128.085	65.806
1KDX	81	5	0	3968	0.16%	428.563	291.235	161.069	85.454
1NFO	131	5	0	3968	0.05%	1535.627	776.641	413.885	221.494
1F1F	88	6	0	46080	0.08%	6150.033	3205.878	1555.229	808.118
1NGR	85	6	0	46080	0.22%	4380.065	2254.021	1112.916	578.604
1QC7	101	6	0	46080	0.03%	8036.353	4054.661	2044.921	1044.061

$N_{AA}$ : Number of Amino Acid;  $N_a$ : number of alpha helix;  $N_b$ : number of beta strand;  
 $N_{Mut}$ : Number of secondary structure mutation ( $N_a! \times N_b! \times 2^{N_a+N_b}$ ), here mutations between helix and strand are eliminated;  
Pct<sub>eff</sub>: the percentage of mutated topologies with lower effective contact energy than that of the native by multi-well function;

Table 2: Speedup for Static Distribution Method

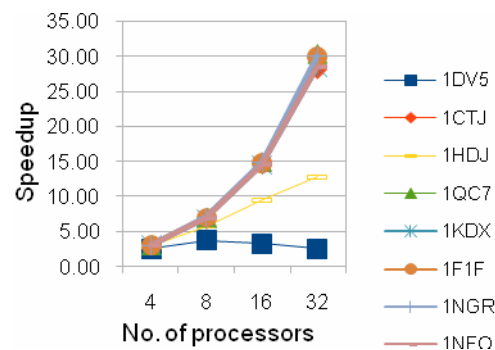
PDB ID	No. of Processors			
	4	8	16	32
1DV5	2.54	3.80	3.28	2.51
1CTJ	3.96	5.81	10.07	19.60
1HDJ	3.83	5.78	7.78	12.41
1QC7	4.00	7.93	15.73	30.82
1KDX	3.93	5.79	10.47	19.73
1F1F	3.92	7.52	15.50	29.83
1NGR	3.98	7.74	15.67	30.14
1NFO	3.96	7.84	14.71	27.48



A

Table 3: Speedup for Dynamic Distribution Method

PDB ID	No. of Processors			
	4	8	16	32
1DV5	2.55	3.75	3.29	2.57
1CTJ	3.00	6.73	14.60	28.13
1HDJ	2.92	5.76	9.53	12.84
1QC7	3.03	7.05	14.97	30.37
1KDX	3.02	6.78	14.51	28.50
1F1F	3.00	6.97	14.80	30.00
1NGR	3.02	6.98	14.86	30.07
1NFO	3.01	6.79	14.52	28.44



B

Figure 3: Speedup curves for two distribution models, A) static model, B) dynamic model.

We compared the speedup between the static (Table 2 and Figure 3A) and dynamic (Table 3 and Figure 3B) distribution of work. The following observations are obtained by ignoring the two smallest proteins, 1DV5 and 1HDJ that have less than 400 permutations. It appears that the static distribution has advantage when less number of processors is used. This is not surprising since the advantage of having a master only appears when there are many processors, since the master does not process the topology and only manages the assignment. The speedup of the dynamic model is generally better when 32 processors are used. However, the best speedup for either model is less than 31 when 32 processors are used. It is possible to improve our implementation to achieve a better speedup.

#### 4. CONCLUSION

Since protein structure is the product of evolution, it is reasonable to believe that the native topology is the most favorable and can hardly be replaced by other mutated structures. We developed a parallel computational approach to test if this belief is supported by the actual data. Our results suggest that the native topology is among a small set of most comfortable topologies, even if it may not be the most comfortable one. We compared two schemes of the work distribution in the parallel code. The comparison provides a basis for improvement of our parallel algorithms.

#### 5. ACKNOWLEDGEMENT

The work in this paper was sponsored by NSF-HRD-0420407, Army High Performance Computing Center and the Active Researcher Support Foundation of Tsinghua University.

#### 6. REFERENCES

- Baker, M. L., W. Jiang, et al. (2006). "Ab initio modeling of the herpesvirus VP26 core domain assessed by CryoEM density." *PLoS Comput Biol* **2**(10): e146.
- Böttcher, B., S. A. Wynne, et al. (1997). "Determination of the fold of the core protein of hepatitis B virus by electron cryomicroscopy." *Nature* **386**(6620): 88-91.
- Conway, J. F., N. Cheng, et al. (1997). "Visualization of a 4-helix bundle in the hepatitis B virus capsid by cryo-electron microscopy." *Nature* **386**(6620): 91-4.
- Del Palu, A., J. He, et al. (2006). "Identification of Alpha-Helices from Low Resolution Protein Density Maps." *Proceeding of Computational Systems Bioinformatics Conference(CSB)*: 89-98.
- He, J., Y. Lu, et al. (2004). "A Parallel Algorithm for Helix Mapping between 3-D and 1-D Protein Structure using the Length Constraints." *Lecture Notes in Computer Science* **3358**: 746-756.
- Jiang, W., M. L. Baker, et al. (2001). "Bridging the information gap: computational tools for intermediate resolution structure interpretation." *J Mol Biol* **308**(5): 1033-44.
- Jones, D. T. (1999). "Protein secondary structure prediction based on position-specific scoring matrices." *J Mol Biol* **292**(2): 195-202.
- Kong, Y. and J. Ma (2003). "A structural-informatics approach for mining beta-sheets: locating sheets in intermediate-resolution density maps." *J Mol Biol* **332**(2): 399-413.
- Kong, Y., X. Zhang, et al. (2004). "A Structural-informatics approach for tracing beta-sheets: building pseudo-C(alpha) traces for beta-strands in intermediate-resolution density maps." *J Mol Biol* **339**(1): 117-30.
- Lu, Y., J. He, et al. (2008). "Deriving topology and sequence alignment for the helix skeleton in low-resolution protein density maps." *J Bioinform Comput Biol* **6**(1): 183-201.
- Ludtke SJ, C. D., Song JL, Chuang DT, Chiu W. (2004). "Seeing GroEL at 6 A resolution by single particle electron cryomicroscopy." *Structure* **12**(7): 1129-36.
- Pollastri, G., D. Przybylski, et al. (2002). "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles." *Proteins* **47**(2): 228-35.
- Przybylski, D. and B. Rost (2002). "Alignments grow, secondary structure prediction improves." *Proteins* **46**(2): 197-205.
- Topf, M., M. L. Baker, et al. (2005). "Structural characterization of components of protein assemblies

- by comparative modeling and electron cryo-microscopy." J Struct Biol **149**(2): 191-203.
- Topf, M., M. L. Baker, et al. (2006). "Refinement of protein structures by iterative comparative modeling and CryoEM density fitting." J Mol Biol **357**(5): 1655-68.
- Wu, Y., M. Chen, et al. (2005). "Determining protein topology from skeletons of secondary structures." J Mol Biol **350**(3): 571-86.
- Zhou, Z. H., M. Dougherty, et al. (2000). "Seeing the herpesvirus capsid at 8.5 Å." Science **288**(5467): 877-880.