Large Scale Protein Side-chain Packing Based on Maximum Edge-weight Clique Finding Algorithm

Dukka Bahadur K.C.[1] **J.B. Brown**[1] **Etsuji Tomita**[2] **Jun'ichi Suzuki**[2] **Tatsuya Akutsu**[1]

[1]*Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan*
[2]*Graduate School of Electro-communications, The University of Electro-communications, Tokyo, Japan*
Email: {dukka,jbbrown,takutsu}@kuicr.kyoto-u.ac.jp, {tomita,jsuzuki}@ice.uec.ac.jp

ABSTRACT:

The protein side-chain packing problem (SCPP) is known to be NP-complete. Various graph theoretic based side-chain packing algorithms have been proposed. However as the size of the protein becomes larger, the sampling space increases exponentially. Hence, one approach to cope with the time complexity is to decompose the graph of the protein into smaller subgraphs. Some existing approaches decompose the graph into biconnected components at an articulation point (resulting in an at-most 21-residue subgraph) or solve the SCPP by tree decomposition (4-, 5-residue subgraph).

In this regard, we had also presented a deterministic based approach called as SPWCQ using the notion of maximum edge weight clique in which we reduce SCPP to a graph and then obtain the maximum edge-weight clique of the obtained graph. This algorithm performs well for a protein of less than 500 residues. However, it fails to produce a feasible solution for larger proteins because of the size of the search space.

In this paper, we present a new heuristic approach for the side-chain packing problem based on the maximum edge-weight clique finding algorithm that enables us to compute the side-chain packing of much larger proteins. Our new approach can compute side-chain packing of a protein of 874 residues with an RMSD of 1.423$\AA$.

1 Introduction

The side-chain packing problem is a central problem in the field of structural bioinformatics. Any method for protein structure prediction has to rely on side-chain packing, as the structure of the protein largely depends upon the conformation of its side-chains. Additionally, success in protein-protein docking requires accurate modeling of side-chain conformations. Hence, the side-chain packing problem has an immense application in the field of structural bioinformatics.

Many early and current methods for protein side-chain packing are based on deterministic as well as heuristic approaches. Some of the deterministic approaches are based on Dead-end-elimination approaches[5] and linear programming approaches[6].

Recently, we developed a new protein side-chain packing algorithm, SPWCQ, to solve the protein side-chain packing problem. Unlike most of the existing side-chain packing methods, we used a deterministic approach based on efficient reduction of the side-chain packing problem to the maximum edge-weight clique finding problem and solve this clique finding problem by using one of the fastest clique finding algorithms developed by our co-authors[7, 8]. Moreover, unlike the majority of the existing methods which use a rotamer library for the sampling of side-chain space, we used discrete rotation angles for sampling of side-chain space. Although SPWCQ performs well for a relatively middle size protein of about 500 residues, the algorithm fails to produce a feasible solution in feasible time for larger proteins due to the combinatorial nature of the protein side-chain packing problem. As the size of the protein becomes larger, the sampling space increases exponentially.

One approach to cope with this exponential increase in the search space is to decompose the graph of the protein into smaller subgraphs. Recent methods decompose the graph into biconnected components at articulation points by which the resulting graph is reduced to subgraphs of at most 21 residues[2], or solve the side-chain packing problem by tree decomposition where the graph of the protein is decomposed into subgraphs consisting of 4-5 residues[3].

In this regard, in this paper we present a new heuristic approach for large-scale prediction of protein side-chain packing based on decomposition of the original graph into subgraphs such that each subgraph will have a maximum edge-weight clique. We show that the new method helps in attaining an increase in the size of proteins whose side chain packing can be calculated. Finally, we show the results of the new method on a set of proteins and illustrate how the new heuristic approach helps in achieving an increase in the size of proteins for our clique based algorithm.

We begin by describing the new heuristic approach, followed by graph generation, and then Computational experiments and Results. Finally, we conclude by discussing the implications of the results and some future directions of the current research.

2 Our Method

Protein side-chain packing is a type of combinatorial optimization problem and involves large amounts of computational time. Especially, when the size of the protein becomes larger the computational time grows significantly.

The fundamental property of side-chain conformations appearing in the final native-like structure is that they have some sense of interconnectedness with some geometrical and biophysical restrictions. In this respect, no side-chain conformation can be correctly predicted without considering the influence of the other residues in the same protein.

Moreover, for deterministic approaches there is always a trade-off between the size of the protein that can be handled and the efficiency of the method.

Our previous approach SPWCQ reduces the SCPP into an edge-weight graph and then computes the maximum edge-weight clique to give an optimal solution. Similar to that of other deterministic approaches, SPWCQ also does not produce a plausible solution if the size of the protein becomes larger than 500 residues. Hence, in this paper we describe a heuristic technique that we have developed in order to cope with the increasing size of the proteins.

In this approach, the original protein is first divided into a number of polymers, and edge weight subgraphs for each polymer are generated. Moreover, the maximum edge-weight clique finding algorithm is applied for all of the generated subgraphs and the final solution is obtained by the union of the vertices of the cliques for these subgraphs.

2.1 Decomposition of Protein into Polymers

For deterministic based algorithms, as the size of the protein increases the search space becomes intractable. Hence, one of the approaches to overcome this intractability is to decompose the search space into smaller search spaces. In our approach, the protein under consideration is first divided into some (say p) equal divisions. Although, more insight into the problem would let us divide the protein at loop regions which are not very important from the structural view point, in this work for theoretical purpose, we divide the protein into equal residue divisions.

Since the main purpose of the paper is to show that the heuristic techniques to divide a protein into smaller subgraphs indeed work, we do not really give much emphasis to the division of a protein at some bio-chemically relevant residue positions although that will be our future direction of the research.

Let us define the graph corresponding to the original PDB as the original graph, and the graphs belonging to polymers as subgraphs.

Let $r_1,\ldots,r_{i-1},r_i,\ldots,r_{l-1},r_l,\ldots,r_m,\ldots,r_n$ be the residues of the protein whose side-chain packing has to be calculated. Moreover, let us consider that we divide the original protein into p polymers with equal number of residues viz. $P_1,P_2,\ldots$ and P_p where polymer P_1 is comprised of residues from r_1 through r_{i-1}, polymer P_2 is comprised of residues from r_i through r_{l-1} and polymer P_p is comprised of residues from r_m through r_n.

2.2 Sampling of Side-chains

After the decomposition of the protein into p polymers, for each amino acid belonging to all of the polymers, the sampling of side-chain search space is done by generating different side-chain conformations by rotating each side-chain of the respective amino acid by an interval of $(2\pi/k)$ where $k = 0,..,K-1$ along the χ_1 axis, generating $2\pi/K$ conformations for each amino acid.

Also, to cope with the capacity of the clique algorithm, only the rotation of side-chain atoms along the χ_1 axis is considered. The value of K is taken to be 18 based on some preliminary experiments, as in SPWCQ.

One of the things to be noted here is that unlike most of the other proposed algorithms, our method does not use a rotamer library that has a much lower number of possible conformations for each amino acid residue.

2.3 Construction of Graph for Each Polymer

2.3.1 Generation of Vertices

After the sampling for the side-chain search space is completed, the corresponding subgraph for each polymer is generated. Let us suppose that we are generating the subgraph corresponding to polymer P_i.

While generating the graph for each polymer, both local consistency (the consistency of the vertices and edges of the polymer with vertices and edges of the same polymer) and global consistency (the consistency of the vertices and edges of the polymer with vertices and edges of other polymers) has to be maintained.

Let $r^i_{u,v}$ be the u^{th} residue in polymer P_i whose side-chain atoms are rotated by $(2\pi v/K)$ radians. Then, all the rotamers $r^i_{u,v}$ that do not collide with the main chain of the original protein are the candidates of the vertices of the subgraph corresponding to polymer P_i.

The rotameric conformation (rotamer) $r^i_{u,v}$ is said to collide with the main chain if the minimum distance between the atoms in $r^i_{u,v}$ and the atoms in the entire main chain is less than $1\AA$.

In a similar manner, a set of candidate vertices for each subgraph of polymers P_1 through P_p is generated.

2.3.2 Generation of Edges

An edge is drawn between a pair of vertices (rotamers) if the vertices are consistent. Two vertices are said to be consistent if the minimum distance between the atoms of these two vertices is greater than $4\AA$.

Moreover, both local consistency and global consistency has to be maintained while generating edges.

Let us consider that we are generating edges for the subgraph corresponding to the polymer P_i. The local consistency and global consistency is maintained in the following manner.

- Local Consistency

  Let $r^i_{u,v}$ be the rotamer generated for u^{th} residue whose side-chain atoms are generated by $(2\pi v)/K$ radians and belonging to polymer P_i, and $r^j_{w,x}$ be the rotamer generated for w^{th} residue whose side-chain atoms are generated by $(2\pi x)/K$ radian and belonging to polymer P_j. In case of local consistency, the polymer is the same, i.e., $i = j$.

  Hence, an edge is drawn between rotamers $r^i_{u,v}$ and $r^j_{w,x}$ where $i = j$, if these two conformations do not collide(i.e the minimum distance between the atoms of these two conformations is not less than $4\AA$) and an edge is not drawn between these two conformations if the minimum distance between their atoms is less than $4\AA$. Moreover, for the weight of the edge, respective weight is also assigned according to the probability distribution function developed by Samudrala et al [9].

Moreover, edges are not drawn between the conformations of the same residue (i.e. no edge is drawn between $r^i_{u,v}$ and $r^j_{w,x}$ where $i = j$ and $u = w$), as each residue can have at most one conformation in the final native-like structure. This step corresponds to considering the collision of conformations within the same polymer.

However, we also have to take into consideration the possible collisions of the rotamers outside their polymer bound. Hence, we also have to check for global consistency.

- Global Consistency

  Global main chain consistency is maintained in the step of vertices generation as the candidates of vertices are checked for consistency against the whole main chain, whereas, in the edge generation until this step, only local side-chain consistency is taken into account. Hence, we also have to check for global consistency for side-chain conformations.

  The conformations of side-chains appearing in the final native-like structure from one polymer should be consistent with the conformations of side-chains from another polymer. This consistency is maintained in the following manner.

  Let us consider that the subgraph for polymer P_i is being generated. While checking for side-chain consistency, it is also necessary to check the consistency of the side-chain conformation of every vertex $r^i_{u,v}$, which is the conformation generated for the u^{th} residue whose side-chain atoms are rotated by $(2\pi v)/K$ radians and belonging to polymer P_i, to all other vertices in other polymers P_j.

  Let $r^j_{w,x}$ be the conformation of the w^{th} residue whose side-chain atoms are generated by rotating $(2\pi x)/K$ radian and belonging to polymer P_j, and combined with $r^i_{u,v}$, be the two vertices and let us suppose that $i \neq j$ and $i < j$. If the minimum distance between atoms of these two rotameric conformations is less than $4\AA$, i.e. if these two rotameric conformations collide with each other, then all the edges connected to this particular vertex ($r^i_{u,v}$ of polymer P_i) in the subgraph of P_i are set to zero as this vertex cannot be included in the final solution; otherwise edges are kept untouched.

  This process of generation of edges is repeated until the consistency with all the polymers is checked. In this way a maximum edge-weight graph corresponding to polymer P_i is generated.

  Once this process is repeated for all polymers from P_1 through P_p, edge-weight subgraphs for all polymers are obtained. Although there is no theoretical proof that each polymer will have a maximum clique with the number of vertices being equal to the number of amino acids in each corresponding polymer, in all our computational experiments we were able to obtain a clique with the size of the number of amino acids in the polymer.

  Particularly, for the generation of the subgraph for the first polymer, we have to take into consideration the consistency of the side-chain with all remaining polymers. However, as we proceed to second polymer and so on, we do not have to check the packing consistency with the vertices of the subgraphs belonging to the polymers prior to the polymer in consideration, i.e. for the polymers for which the corresponding subgraphs are already generated. This step is particularly helpful in reducing the computational time for the generation of subgraphs for consecutive polymers and the simultaneous calculation of the cliques in each polymer.

  In this way, subgraphs are generated for each polymer. Finally, one of the fastest clique finding algorithms [7, 8] is utilized for each subgraph and the final solution is obtained by taking the union of the vertices of the maximum edge-weight cliques of each subgraph.

2.3.3 Weight Function

We use the all-atom distance dependent conditional probability based discriminatory function proposed by Samudrala *et al.*[9] to assign weight to the edges of the subgraphs.

Readers are requested to refer to Samudrala *et al.*[9] for the detailed exposition of the function. In essence, this probability function gives some value given two atoms and the distance between these two atoms. Especially, while assigning the weight to the edge between two vertices, the distance between each pair of the atoms of each vertex (conformation) is calculated and assigned some weights. The total weight corresponding to the summation of weights for all pairs of atoms gives the weight of the edge joining any two vertices.

For our calculation, the score table is pre-calculated to enhance the efficiency of the algorithm. This score table is a three dimensional table whose elements represent the score between two atom types separated by some specific distance. Hence, if the atom types and the distance between these atom types are known, by referring to the corresponding element of the score-table, score can be easily calculated. For the calculation of weights between any two conformations (vertices) of the sub graph generated, the sum of the scores of corresponding atoms of the conformation is calculated and assigned as the score to the edge between these two corresponding vertices.

2.4 Clique finding algorithm: WCQ

The maximum clique finding algorithm developed by coauthor(Suzuki and Tomita) is applied for finding the maximum edge-weight clique of the graph generated above. For the detailed description of the algorithm, the readers are requested to refer to [7, 8]. Here, a brief overview of maximum edge-weight clique algorithm is presented.

Let the number of vertices in the maximum clique be ω and the weight between two vertices p and $q \in V$ be defined as $w(p,q)(= w(q,p))$, then the weight of the clique $G'(= (V',E'))$ can be represented as $W(V')$.

WCQ essentially finds all the maximum cliques in the given graph and returns the clique with the maximum weight as the output. The algorithm basically maintains four variables Q,

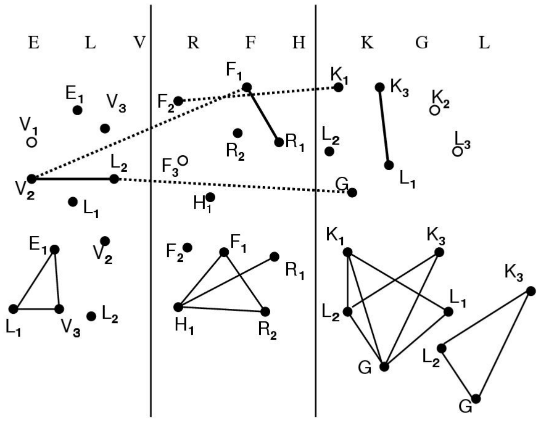

Figure 1: An example showing the side-chain packing of a protein of sequence ELVRFHKGL. In this idealized example, the protein is divided into three polymers at residues represented by vertical lines. Hence, sequence ELV makes the first polymer, sequence RFH makes the second polymer and the sequence KGL makes the third polymer. At first, an edge weight graph is obtained for each polymer as described in the graph generation section. Let us consider that V_1, V_2, V_3, E_1, L_1 and L_2 are the conformations of the respective side-chains of amino acids V, E, and L in the first polymer. The conformations represented in small white circles represent that the conformation collides with the main chain, and hence will be deleted from the set of vertices. The conformations represented by small black circles denote that these conformations do not collide with the main chain. The solid line between two conformations represents that these two conformations collide with each other within the same polymer and the dotted line represents that the two connected conformations collide with each other outside the polymer. So, for the first polymer, V_1 does not appear in the final set of vertices, and there is no edge between V_2 and L_2 as these conformations collide within the polymer. All the edges connected to V_2 and L_2 are deleted as V_2 collides with F_1 of the second polymer and L_2 collides with G of the third polymer. Hence, the final subgraph consists of three vertices E_1, L_1 and V_3. Similarly, in the second polymer, F_3 does not appear in the set of vertices for the final graph. Moreover, all the edges connected to F_2 are deleted and there is no edge between F_1 and R_1. Hence the final subgraph will be as shown in the figure. Finally, for the third polymer, the subgraph would be as shown in the figure. After all the subgraphs are generated, the maximum edge weight clique finding algorithm is utilized on each subgraph and the maximum edge-weight clique is obtained for each of these subgraphs. The final solution is obtained by the union of the vertices of the maximum edge-weight cliques of the respective subgraphs.

Q_{max}, R and $W(Q)$ for the weight. Let us call the number obtained by NUMBERING-ARRANGING (i.e. the upper bound of the size of the clique that can be obtained by searching p) as $No(p)$. In order to skip the recursive steps on R, R is not expanded if $|Q| + No(p) \leq |Q_{max}|$.

Regarding the calculation of the weight of the clique, addition of one new node to the set of current maximum clique Q adds $|Q|$ edges to the graph, and one has to add the weights of each of these added edges. In order to cope up with the weights of maximal clique, the weight of the clique before adding these edges is kept as W_{pre} and in the process of searching, backtracking one step can lead us to the weight of $W(Q)$. By doing this, the efficiency of the calculation of the weights is increased as there is no need to calculate the weights of the edge from the very beginning each time. When $W(Q) > W(Q_{max})$, renewal of the weight clique is done by assigning Q to Q_{max}. Finally, when every maximum cliques of the graph is enumerated, Q_{max} is the maximum weight clique of the given graph.

Essentially, the algorithm WCQ runs as follows:

```
procedure WCQ($G = (V, E)$)
begin
  $Q := \emptyset$; $Q_{max} := \emptyset$;
  $W(Q) := 0$; $W(Q_{max}) := 0$;
  Sort vertices of $V$ in non-increasing order
    with respect to their degrees;
  NUMBERING-ARRANGING($V, No$);
  EXPAND-WCQ($V, No$)
  output $Q_{max}$
end of WCQ
```

```
procedure EXPAND-WCQ($R, N$)
begin
  while $R \neq \emptyset$ do
    $p$ := the vertex in $R$ such that
      $No(p)$ = Max $No(q)|q \in R$
    if $|Q| + No(p) \geq |Q_{max}|$ then
      $W_{pre} := W(Q)$;
      for $i := 1$ to $|Q|$ do
        $W(Q) := W(Q) + w(p, Q[i])$;
      od
      $Q := Q \cup \{p\}$
      $R_p := R \cap \Gamma(p)$;
      if $R_p \neq \emptyset$ then
        NUMBERING-ARRANGING($R_p, No'$)
        EXPAND-WCQ($R_p, No'$)
        else if $W(Q) > W(Q_{max})$ then
          $Q_{max} := Q$
          $W(Q_{max}) := W(Q)$
        fi
      fi
    fi
    $Q := Q - \{p\}$;
    $W(Q) := W_{pre}$
    $R := R - \{p\}$
  od
end of EXPAND-WCQ
```

3 Computational Experiments and Results

We perform a set of computational experiments in order to assess the prediction accuracy of our new approach. Moreover, in order to assess the computational time of the approach we also perform a set of experiments.

As mentioned in the Our Method section, in this paper we do not give much emphasis on breaking the original protein into bio-chemically relevant polymers. For simplicity, we just divide the polymer into p polymers where each polymer has equal an number of residues. The computational environment used for the experiments is a PC with an Intel Pentium 2.66GHz CPU running the Linux operating system.

The approach is utilized to predict the side-chain positioning of a set of proteins by dividing each protein into p polymers, where p=2,3,4 or 5 equal divisions and the results of the respective RMSD of the computed structure and the native structure are shown.

3.1 Accuracy of Prediction

The computational experiments for a set of eight proteins are performed for the validation of the new approach. Each protein was divided into an equal number of polymers p, where p=2,3,4, or 5. After the generation of the graph for each polymer of these proteins, the maximum edge-weight clique finding algorithm WCQ is utilized.

The results of the computation of protein side-chain packing for a set of proteins is given in Table 1 along with the respective RMSD for each protein.

PDB	#Residue	p=2	p=3	p=4	p=5
1tdj	514	1.64	1.67	1.71	1.71
1xwl	580	1.66	1.56	1.60	1.74
1gof	639	1.63	1.46	1.69	1.70
1biy	689	1.53	1.64	1.77	1.72
1aa6	696	1.58	1.68	1.61	1.7
1a8i	812	1.56	1.63	1.63	1.75
1lnh	836	1.50	1.63	1.64	1.55
1fiy	874	1.42	1.53	1.63	1.54

Table 1: Performance of our approach for a set of eight proteins. The first column represents the PDB code of the protein and the second column represents the number of residues in the protein. The column under p=2 give the RMSD results when the protein is divided into two polymers; $p = 3$ gives the RMSD of the computed structure when the protein is divided into three polymers, and so on.

3.2 Computational Time

Moreover, in order to know the time complexity of the new approach, we also measured the computational time for the prediction of the side-chain packing of the same set of proteins as in Table 1, when each protein is divided into two, three, four and five polymers. The results of the computational time are presented in Table 2.

PDB	#Residue	p=2	p=3	p=4	p=5
1tdj	514	481	232	123	190
1xwl	580	758	394	267	129
1gof	639	530	346	294	178
1biy	689	620	416	680	403
1aa6	696	652	580	263	414
1a8i	812	882	972	483	479
1lnh	836	1116	1150	600	350
1fiy	874	1289	438	691	555

Table 2: Computational time of our approach for a set of eight proteins. The first column represents the PDB code of the protein and the second column represents the number of residues in the protein. The third, fourth, fifth and sixth columns represent the computation time in seconds when the protein is divided into two, three, four and five polymers respectively.

4 Discussion

We have developed a new approach for side-chain packing based on a heuristic technique to divide the original graph of the protein into subgraphs and search for the maximum edge-weight cliques in these subsequent subgraphs. After the computation of the maximum edge-weight cliques in each subgraph, the union of the vertices of these maximum edge-weight cliques is obtained, giving the final solution of the protein side-chain packing problem.

Although we do not have any theoretical proof that all the subgraphs will have the maximum edge-weight clique with the number of vertices being equal to the number of residues in each polymer, in all of our computational experiments we were able to obtain a clique with the number of vertices being equal to the number of residues in each polymer. It is to be noted here that for proteins with size less than 500 residues, our previous approach SPWCQ [4] is best suited. Hence in this work, we only perform the computational experiments for proteins larger than 500 residues.

The performance of this approach for a set of proteins ranging from 514-874 residues is very enthusiastic, with the worst RMSD being 1.77$\AA$. Especially, as per our initial suspicion the algorithm works best for p=2 (p is the number of polymers), which is consistent with our initial definition of SCPP as a single maximum edge-weight clique. The computational time shown in Table 2 depicts that the time for the computation of the side-chain packing decreases as the number of polymers increases. Moreover, from Table 1, it can be seen that RMSD does not deteriorate proportionally to the number of decomposed subgraphs, which elucidates that we can still increase the time efficiency of the algorithm without expense of the accuracy of prediction.

One of the future directions of the research is to incorporate biological knowledge in order to break apart the protein at bio-chemically relevant residues. One of the possible relevant residues could be the residues belonging to the loop regions that are not very important from a structural aspect. Furthermore, in this paper we use the discrete rotamers generated by rotating each side-chain through some specific angles. Hence, other possible improvements are the usage of a backbone dependent rotamer library.

Theoretically, our method should work for a protein of any size. However, due to the size of the original graph which has to be taken into account for the subgraph generation, we were not able to compute side-chain packing for much larger proteins. This also owes to the fact that for each side-chain we are considering a lot more rotamers than the usual rotamers in the rotamer library. Hence, we hope to increase the size of proteins that our method can predict by using the available rotamer library.

REFERENCES

[1] T. Akutsu. NP-Hardness results for protein side-chain packing. Genome Informatics, 8:180–186, 1997.

[2] A.A. Canutescu, A.A. Shelenkov, and R.L. Dunbrack. A graph theory algorithm for protein side-chain prediction. Protein Science, 12:2001–2014, 2003.

[3] J. Xu. Rapid protein side-chain packing via tree decomposition. In Proc. 9th RECOMB, pages 423–439, Boston, USA, 2005.

[4] Dukka K.C., E. Tomita, J. Suzuki, and T. Akutsu. Protein Side-chain Packing Problem: A Maximum Edge-weight Clique Algorithmic Approach. Journal of Bioinformatics and Computational Biology, 3(1):103–126, 2005.

[5] L.L. Looger, and H.W. Hellinga. Generalized Dead-end Elimination algorithms make large-scale protein side-chain structure prediction tractable: implication for protein design and structural genomics. Journal of Molecular Biology, 307:429–445, 2001.

[6] C.L. Kingsford, B. Chazelle, M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics,21(7):1028–36, 2004.

[7] J. Suzuki, J., Tomita, E. and Seki, T. An algorithm for finding a maximum Clique with maximum edge-weight and computational experiments . In Technical Report MPS, page 45–48,The Information Processing Society of Japan, 2002.

[8] E. Tomita, and T. Seki. An efficient branch-and-bound algorithm for finding a maximum clique, DMTCS 2003, Lec. Notes in Comput. Sci., 2731:278–289, 2003.

[9] R. Samudrala, and J. Moult. An all-atom distance dependent conditional probability discriminatory function for protein structure prediction, Journal of Molecular Biology, 275:893–914, 1998.