Protein Side-chain Packing Problem: A Maximum Edge-weight Clique Algorithmic Approach

Dukka Bahadur K.C.† **Tatsuya Akutsu†** **Etsuji Tomita ‡** **Tomokazu Seki‡**

†Bioinformatics Center Kyoto University,Kyoto 611-00011, Japan
‡Graduate School of Electro-communications, The university of Electro-Communications, Tokyo, Japan
Email: dukka@kuicr.kyoto-u.ac.jp

Abstract

Protein side-chain packing has an important application in homology modeling, protein structure prediction, protein design, protein docking problems and many more.

Protein side-chain packing problem is computationally known to be NP-hard (Akutsu, 1997) (Chazelle,Kingsford & Singh, 2003) (Pierce & Winfree, 2002). In the field of computer science, the notion of reduction of a problem to other problems is quite often used to design algorithms and to prove the complexity of a certain problem. In this work, we have used this notion of reduction to solve protein side-chain packing problem.

We have developed a deterministic algorithm based approach to solve protein side-chain packing problem based on clique-based algorithms. For this, we reduced this problem to the maximum clique finding problem. Moreover, in order to incorporate the interaction preferences between the atoms, we have then extended this approach to maximum edge-weight clique finding problem by assigning weights based on probability discriminatory function. We have then solved this clique finding problem by using the clique finding algorithm developed by two of the authors (Tomita & Seki, 2003) and its variants(Suzuki, Tomita & Seki 2002).

We have tested this approach to predict the side-chain conformations of a set of proteins and have compared the results with other existing methods. We have found considerable improvement in terms of the size of the proteins and in terms of the efficiency and accuracy of the prediction.

Keywords: Protein side-chain packing, Clique, Homology Modeling, Protein Design, Protein Docking

1 Introduction

Various structural genomics consortium are undertaking pilot projects targeting representative proteins to provide coverage of fold space and in some years the approimate folds will be known. However, these approximate folds are not sufficient in order to be used for structural based drug design and detail analysis of metabolic pathways. Hence, the methods like 'Homology Modeling' will still be an important player in the determination of protein structure.

For any methods dealing with protein structure prediction, finding the optimal side-chain positioning is a crucial step. Efficient and accurate prediction of side-chain positioning or conformations is an important step of homology modeling methods to predict the protein structures, of x-ray crystallography methods for the initial placement of electron-density maps in experimental determination, of protein design methods to design an amino acid sequence that folds in particular function, and also of protein-ligand and protein-protein docking methods to calculate the docked conformation.

Although a large number of methods for the prediciton of side-chain packing have been published in recent years, the problem of protein side-chain packing still remains unsolved in the case of larger proteins. This is due to the combinatorial nature of the protein side-chain packing problem.

In this paper, we formulate the protein side-chain packing problem as maximum clique finding problem and then solve this clique problem to find the solution of protein side-chain conformations. In the following sections, we briefly discuss about the protein side-chain packing problem, existing methods for this problem and then we describe our methodology and results.

2 Protein Side-chain Packing Problem

Protein folds comprises of two classes of degrees of freedom: the ϕ-ψ angles which determine the main-chain of the structure and the χ torsion angles which determine the side-chain packing. The side-chain packing is important due to the fact that the overall stability of the structures is determined by packing. In computational biology, the *Protein Side-chain Packing Problem (PSPP)* can be defined as a problem of finding a side-chain conformation with the minimum potential energy given an amino acid sequence and spatial information on the main chain of a protein.

In other words, given a protein main chain conformation, constructing side chains by exploring all possible rotamer conformations simultaneously is called protein side-chain packing problem. More precisely, it seeks a set of χ $(\chi_1, \chi_2, \ldots)$ angles whose potential energy becomes the minimum, where positions of atoms in the main chain are fixed.

Essentially, in this problem calculation of the side-chain conformations of each amino acid given the amino acid sequence and the main chain model is performed. The major problem encountered when modeling side-chains is the extremely large number of possible combinations of side-chain conformations.

Basically, all the methods to solve protein side-chain packing comprises of two basic two steps. The first step is the representation of the problem i.e. the representation of the side-chain search space and the second step is the searching step to search through the represented search space.

However, for practical purposes following simplification is done: the search space is discretized and the search space is reduced by limiting the number of possible conformation to some finite set of possible torsion angles.

Generally, methods for side-chain packing use the fact that the side-chain conformations exist in a limited number of shapes called rotamers. Another assumption made in order to simplify the packing process, is that the backbone conformation does not change with the placement of side-chains thus, restraining the problem to the fixed backbone.

Finally, an energy function is introduced in order to refine the model. These functions ranges from simple functions like Van der Waals interaction to more complex functions like the calculation of free energy.

3 Existing Methods for Side-chain Packing

The main difficulty in predicting the conformation of side-chains is the enormous number of conformations possible for even a small residue protein. Considering a small protein of only 10 torsion angles, and assuming that side-chain torsions are rigid rotations divided into discrete 10 deg steps (so that each χ angle has 36 distinct possible states), this problem results into searching of around 2.4×10^{14} conformations. Although, it has been shown experimentally that protein torsion angles actually have some preferences, even then the problem of finding correct side chain positioning for a larger protein involves enormous search space. In this regard, protein side-chain packing problem is already proved to be NP-hard (Akutsu, 1997), (Chazelle et al., 2003), (Pierce & Winfree, 2002).

There are three major approaches to solve a problem in NP-hard: 1)Using heuristic approaches 2) using deterministic approaches like branch and bound techniques and 3) using approximation techniques (i.e. designing approximation algorithms).

In this regard, the search procedure for the existing methods can also be broadly characterized into either belonging to methods based on heuristic approaches or to methods based on deterministic approaches.

As discussed above, protein side chain packing problem consists of two important aspects viz. representation of the search space and searching through this search space.

3.1 Side-chain Sampling

For the search space representation, some methods try to reduce the search space by restricting the number of conformations allowed for each type of side-chains to only a few basic rotamers. This method has the advantage that the search space is significantly reduced but in the same time has an important drawback of assuming that the rotamer library used represents the search space efficiently, which may not be always true due to the fact that there are still a large number of proteins whose structures are still not known and incorporating these unknown protein structures may change the rotamer library itself.

Other methods, instead of using predefined rotamers, try to explore all the possible conformations of the side chains without restricting the search space. This type of method generally involves discretizing the torsion angles in some discrete values. This type of method can be considered exhaustive conformational space representation but has the demerits that the number of possible conformations for even a small residue becomes large and is often not applicable to predict all the side-chain conformations of a large protein as the combinatorial nature of the problem makes this problem outside the scope of existing computational resources.

To our knowledge most of the existing methods (Holm & Sander 1992, Desmet, De Maeyer, Hazes & Lasters 1992, Dunbrack & Karplus 1993, Laughton 1994, Samudrala & Moult 1998) use rotamer library in order to reduce the possible side chain search space, the only methods using the discrete rotation angle is the one by Lee *et al.*(Lee & Subbiah, 1991).

3.2 Searching Algorithms or Methods

The other important aspect of the protein side-chain packing problem is the search algorithm or procedures the method utilizes to search through the sampled side-chain space. The search methods utilized can be broadly classified into two groups: 1) Using heuristic approaches 2) Using deterministic approaches. The methods based on simulated annealing, monte carlo simulation and other genetic based algorithms are the examples of heuristic based approaches(Voigt, Gordon & Mayo, 2000) have demonstrated that these stochastic based algorithms are trading accuracy for speed i.e. these methods are not guaranteed of finding a global minimum configuration of the side-chain packing.

3.2.1 Heuristic approaches

Methods based on monte carlo simulation (Holm & Sander, 1992), cyclical search, simulated annealing (Lee et al., 1991), other genetic algorithms and mean field optimization (Koehl & Levitt, 1999) belong to this group of methods. Since these methods apply some heuristics in the calculation, these methods are approximate. Even though these methods can find a low-energy conformation in a reasonable time, the solutions obtained by these methods are not guaranteed to be a global optimal solution.

3.2.2 Deterministic Approaches

One of the most widely used deterministic approach in protein side-chain packing are the methods based on Dead-end Elimination(DEE)(Desmet et al., 1992). DEE methods are based on a simple criteria that allows pruning of rotamers that cannot be the members of the Global Minimum Energy Conformation. Thus, applying this criteria subsequently, the search space is significantly reduced and most probably leads to a single solution. These methods like Dead-End Elimination and A^* (Leach & Lemon, 1998) algorithms are guaranteed of finding an optimal solution if the methods converge. However, there are times when the methods do not converge and some appropriate methods have to be introduced in order to cope up with this problem. Another main problem with this approach is that this method has not been successfully implemented for larger proteins due to the fact that there is a limit beyond which this method fails to converge. Moreover, the algorithms based on this approach are problem specific. Another major disadvantage of these methods is that the accuracy and speed of the method depends largely on the rotamer library used.

Hence, with the motive of designing an algorithm based on deterministic approach that can be applied for real size proteins in plausible time, we have done this current work.

4 The side-chain packing as a maximum clique finding problem

The main difficulty in the side-chain packing problem is the tremendous size of the solution space that is needed to be searched. One of the fundamental reason for an exponential number of possible conformations of an amino acid sequence needed to be analyzed in order to determine the best conformation is due to the fact that each residue conformation in a sequence cannot be built in isolation of other residue conformations, as different regions of the sequence influence each other in the 3D structures.

A conformation of a amino acid sequence in which a side chain of a residue colliding with another side chain of another residue cannot be a native-like conformation. Similarly, if side-chain of a residue collides with the main chain of the other residues then also the conformation of the amino acid sequence cannot be regarded as native-like conformation. In this regard, a sense of interconnectedness can be observed in between the protein side-chains and protein main-chain.

The notion of graph used in computer science exemplifies the interconnectedness of protein side-chains and main-chain. Various graph-theoretic approaches have been utilized in the field of computational chemistry and biology for the interconnected networks as that of protein conformation. Since our goal is to find the best set of interactions in a protein structure given a variety of side chain and a fixed main-chain, we represent this interaction as a graph and search for a clique which represents the native-like conformation.

4.1 Overview of SPMCQ

Let us call this version of the algorithm for side-chain packing as SPMCQ. In SPMCQ, the protein side-chain packing problem is reduced into an undirected graph. In this reduction, each possible conformations of a residue in an amino acid sequence is represented using the notion of a node in a graph. Then, the edges are drawn between compatible nodes. Once the entire graph is completed, the clique finding algorithm is used to find a maximum clique and this maximum clique corresponds to the solution of protein side-chain packing problem i.e the native conformation of the protein structure.

4.2 Reduction to Clique

In computer science the notion of reduction is widely used in the theoretical aspects of algorithms. Especially, in order to prove the complexity of a given problem, a problem already known to belong to a certain class of problems is reduced to the new problem. Reduction can be defined as the computable transformation of one problem into anther. Here we use the notion of reduction and reduce the protein side-chain packing problem to the maximum clique finding problem.

In our reduction, every possible conformation of a side chain residue is represented as a node and then edges are drawn between these nodes if these nodes satisfy some criteria. After the entire graph is generated, the clique finding algorithm is applied to enumerate every nodes and find a maximum clique of the graph. This maximum clique represents the conformation which is most closely related to the native structure i.e. the solution of the protein side-chain packing problem.

A brief overview of clique and then the clique algorithms developed by two of the authors (Tomita & Seki, 2002) is presented below.

4.3 Clique Algorithm MCQ

Let us call this version of the algorithm to find the maximum clique as MCQ. Consider a set of vertices V and a set $E \subseteq V \times V$ of edges then $G = (V, E)$ represents a graph. For a graph $G(V, E)$ in which $\forall i, j \in V$ and $i \neq j, (i, j) \in E$ i.e. all two vertices are adjacent, the graph $G = (V, E)$ is called as a complete graph. For a subset C of G, if $G(C)$ is complete, then C is called a clique. If this obtained clique is not the real subset of other cliques obtained from the same graph, then it is called as maximal clique.

Similarly, maximal clique with the maximum number of nodes is called as the maximum clique. The problem of finding all the maximum cliques of a given graph G is called the maximum clique finding problem. For a given graph G, the problem of finding maximum clique belongs to a group of problem called NP.

There are many well established maximum clique finding algorithms like (Bron & Kerbosch 1973) but the new clique finding algorithms (Tomita & Seki, 2002) (Tomita & Seki, 2003) developed by two of the authors is utilized in this study. This maximum clique finding algorithm has been proved to be many times faster than the basic Bron and Kerbosh algorithms. We would like to direct queries regarding the efficiency and other details of maximum finding algorithm to Tomita Etsuji(tomita@ice.uec.ac.jp).

Let $G = (V, E)$ be an undirected graph, where V is the set of vertices and E is the set of edges. For each $v \in V$, $\Gamma(v)$ denotes the set of vertices adjacent to v (i.e., $\Gamma(v) = \{w | \{v, w\} \in E\}$) and $deg(v)$ denotes the degree of v.

For $S \subseteq V$, $G(S)$ denotes the subgraph induced by S. $G(C)$ is a clique if $G(C)$ has $|C|(|C|-1)/2$ edges. The maximum clique is a clique with the maximum number of vertices.

We define a basic variant of the maximum clique finding algorithm developed by our group.

The basic algorithm finds a maximum clique incrementally using a recursive procedure. It maintains variables Q, Q_{max} and R. Q consists of the vertices of the current clique, Q_{max} consists of the vertices of a maximum clique found so far and R consists of the candidates of vertices which may be added to Q.

In order to avoid enumerating all maximal cliques, *approximate coloring* of vertices is used. A number (color) $No(p)$ is assigned to each vertex p in candidate set R so that the following conditions are satisfied: (i) $No(p) \neq No(q)$ if $\{p, q\} \in E$, (ii) if $No(p) = k$ then $\{No(q) | q \in \Gamma(p)\} \supseteq \{1, 2, \ldots, k-1\}$. It is easy to see that the recursive steps on R can be skipped if $|Q| + \max\{No(p) | p \in R\} \leq |Q_{max}|$.

The pseudo code of the algorithm is as follows:

```
BasicCliqueMCQ(G)
  Q_max := ∅;  Q := ∅;  R := V;  Expand(R);
Expand(R)
  while R ≠ ∅ do
    Let p be a vertex in R such that No(p)
    is the maximum;
    if |Q| + No(p) > |Q_max| then
      Q := Q ∪ {p};  R_p := R ∩ Γ(p);
      if R_p ≠ ∅ then Expand(R_p)
      else if |Q| > |Q_max| then Q_max := Q;
      Q := Q − {p};
    else return;
    R := R − {p};
```

In order to reduce the computational time of computing coloring every time when R_p is updated, approximate colorings are re-computed at each update of R_p. A simple technique is introduced in order to reduce the time for updating approximate coloring.

There are several improved variants, depending on the methods for approximate coloring and the methods for updating approximate coloring.

4.4 Sampling of Side-chains

Most of the existing methods utilize a rotamer library of discrete side-chain conformations obtained from statistical analysis of the data sets in the data bank. Although our methods can be modified in order to incorporate real rotamer angles derived from various databases, we use the discrete rotation angles in our present work. The set of rotation angles is defined by $(2\pi k)/K \| k = 0, ..., K-1$.

Hence, in order to sample the side-chain sampling space for generating different conformations of side chains each side-chain was rotated by an interval of $(2\pi k/K)$ angle along the χ_1 axis, generating $2\pi/K$ conformations for a single side-chain. While doing this, the rotation of side-chain atoms along the χ_1 axis is only considered in order to cope up with the capacity of the clique algorithm. Most of the rotamer library have around three rotamers for a single side chain, whereas our method uses 20 rotamers for each side-chain position. In order to get the appropriate number of conformations for each side chain, an experiment was performed with four sets of angles viz. K=16, 18, 20 and 22. On the basis of the performed experiments, K = 18 is selected and 20 conformations are obtained for each side-chain. These obtained conformations serve as the candidates for the nodes of the graph.

4.5 Generation of the graph

In this work, at first only a simple geometric condition is taken into consideration for side-chain packing. In this reduction, whether or not each side chain collides with the main-chain or the other side chains is checked. Moreover, χ_1 angles (i.e., rotations around the vector defined by C_α and C_β atoms) is considered though our method can be extended so that $\chi_2, \chi_3, \ldots$ angles are also taken into account.

Let $R = \{r_1, \ldots, r_n\}$ be the set of residues of the given protein whose side-chain conformations has to be calculated. Each residue consists of positions of atoms in the side chain, where hydrogen atoms are ignored in the current implementation. The positions of atoms in a side-chain are rotated around the χ_1 axis.

Thus in the reduced graph the nodes and edges are as the one defined below.

4.5.1 Nodes of the graph

After obtaining 20 conformations for a single side-chain, checking was done to justify whether the conformations generated in this way collide with the backbone of the protein sequence or not using the criteria described below. For this, only those conformations of side-chains which do not collide with the main chain as nodes in our graph are considered. This is due to the fact that in a native conformation, side-chain atoms do not collide with the main-chain atoms. In order to check the collision of side-chain atoms and main-chain atom, all the main-chain atoms are considered in order to cope up with the idea that the protein is not an elongated structure rather it is a folded structure and thus, the main chain atoms which are far apart if observed in the sequence can come together in the $3D$ structure.

Mathematically, let $R = \{r_1, \ldots, r_n\}$ be the set of residues of the given protein whose side-chain conformations has to be calculated. Each residue consists of positions of atoms in the side chain, where hydrogen atoms are ignored in the current implementation. The positions of atoms in a side-chain are rotated around the χ_1 axis.

Let $r_{i,k}$ be the i-th residue whose side-chain atoms are rotated by $(2\pi k)/K$ radian. It is said that *residue $r_{i,k}$ collides with the main-chain* if the minimum distance between the atoms in $r_{i,k}$ and the atoms in the main-chain is less than $L_1\mathring{A}$.

4.5.2 Edges of the graph

Similarly, edges are drawn between a pair of nodes generated in the above step. While drawing edges, edges are not drawn between pairs of nodes if the nodes collides ,i.e. if the minimum distance between the atoms in the pairs of nodes under consideration is less than $L_2\mathring{A}$. Moreover, edges are not drawn between different conformations of the same residue.

Let $r_{i,k}$ be the i-th residue whose side-chain atoms are rotated by $(2\pi k)/K$ radian and $r_{j,h}$ be the j-th residue whose side-chain atoms are rotated by $(2\pi h)/K$ radian. Then edge is drawn between these two conformations if the conformation $r_{i,k}$ does not collide with the conformation $r_{j,h}$ i.e. if the minimum distance between the atoms in $r_{i,k}$ and the atoms in $r_{j,h}$ is less than $L_2\mathring{A}$. In this work, $L_1 = 1.5\mathring{A}$ and $L_2 = 4.0\mathring{A}$ are used.

4.5.3 Graph

As described in the above section, edges and nodes are generated in the consistent manner for every amino acids of the protein whose side-chain positioning has to be calculated. Finally, a graph $G = (V, E)$ which is defined by $V = \{r_{i,k} \| where\ r_{i,k}$ does not collide with the main chain $\}$, $E = \{\{r_{i,k}, r_{j,h}\} |\ i \neq j,\ r_{i,k}$ does not collide with $r_{j,h}\}$ is obtained.

It is easy to see that a maximum clique of size n corresponds to a consistent configuration (i.e., a configuration in which any two atoms do not collide).

For the sake of better comparison among the maximum clique approach, weighted maximum clique approach and various other methods, the results are presented along with the results obtained by the weight-clique approach. Here, it is concluded that since no energy functions or atom interaction preferences were used, better results could be obtained if these criteria are included in our method. Hence, the extension of this method to maximum edge-weight clique approach is performed.

5 Towards Maximum edge-weight Clique

Up to this stage, the side-chain packing problem is defined as a simple geometric problem without considering any energy functions or functions concerning interactions between atoms of the side-chains and main-chain. But in reality, the atoms of side-chains and main-chain interact with each-other. These interactions may occur due to attraction between the atoms or the interaction occurring due to the repulsion between the atoms.

As atoms do have some sort of preferences for other atoms for interaction, it can be concluded that the protein side-chain packing problem would be defined more precisely if some energy functions are considered along with those geometrical characteristics that are considered above.

In such a case, 'the maximum clique' can be replaced by 'the maximum weight clique' so that the

energy between residue pairs can be taken into account. It seems that maximum weight clique algorithms are also applicable to other problems.

Hence, protein side-chain packing problem can be better characterized by reducing the problem into a weighted graph so that maximum weight clique finding algorithms are applicable. Thus, the extension of non weighted version of the clique algorithm to the weighted version is done.

6 Overview of SPWCQ

Let us call the weighted version of the side-chain packing algorithm as SPWCQ. The reduction of protein side-chain packing to the graph is the same as the one described in maximum clique approach. The only thing that is different from the maximum clique approach is the assigning of weights to edges of the graph. In this approach, after obtaining a graph as described in the above section, a weight is assigned to every edges of the graph using a discriminatory function, which is described below. After the graph is generated, the maximum weight clique finding algorithm is applied to achieve the maximum weight clique of the graph.

6.1 Description of Nodes

Nodes are defined as in the previous definition of nodes in the non-weighted version of the maximum clique section.

6.2 Description of Edges

Similarly the edges are defined as in the previous definition in the non-weighted version of the maximum clique section.

6.3 Weights of edges

Our objective here is to assign weights to edges of a graph by determining the strength of interactions of a side-chain to the local main-chain and by determining the strength of interaction between two side-chains. Since the important aspect of this work is to focus more on the computational aspect i.e. designing of an algorithm, it was decided to use the existing discriminatory function for the purpose of assigning weights. Hence, to assign weights in terms of the atomic preferences the edges of the graph, survey of various existing methods like force fields and other knowledge based discriminatory functions was performed.

In order for the function to be useful in our approach, the weight function should be easily calculated and should be able to discriminate the atomic preferences. It was decided that the function based on residue-specific all-atom probability discriminatory function proposed by Samudarala *et al.*(Samudrala & Moult 1998) to best fit our purpose.

For this, an all-atom distance dependent conditional probability-based discriminatory function is used to calculate the conditional probability of contacts of a given pair of atom types in a given distance for a given conformation of interest. The conditional probabilities for the residue-specific all atom probability discriminatory function are compiled by counting frequencies between pairs of atoms with in a specified distance in a database of protein structures.

Since one of the most important part of our approach is enhancing the efficiency of the side-chain packing methods, pre-calculation of the score table is performed based on the above discriminatory function.

6.4 Simple description of Weight Function

A brief review of the probability discriminatory function proposed by Samudrala *et al.*(Samudrala & Moult, 1998), used in this work is given below.

The possible conformation of a structure is divided into two types viz. the set of correct conformations C and the set of incorrect conformations I.

A set of inter-atomic distance within a structure d_{ab}^{ij}, where d_{ab}^{ij} is the distance between atoms i and j, of type a and b respectively. Here, we are interested in calculating $P(C|d_{ab}^{ij})$,the probability that the structure belongs to the group of correct ones given the inter-atomic distances d_{ab}^{ij} of the structure. From the chain rule of probability, $P(C|d_{ab}^{ij})$ can be expressed as a function of expressions that can be obtained from the experimental structures. Thus,

$$P(C)P(d_{ab}^{ij}|C) = P(d_{ab}^{ij})P(C|d_{ab}^{ij}) \quad (1)$$

where,$P(d_{ab}^{ij}|C)$ is the probability of observing a set of distance d between two atoms i and j of atom types a and b given a correct structure, $P(d_{ab}^{ij})$ is the probability of such set of distance in any structure and $P(C)$ is the probability that any structure picked at random is a correct one.

Here, $P(d_{ab}^{ij}|C)$ can be expressed as the product of the probabilities of observing each distance. Thus,

$$P(d_{ab}^{ij}|C) = \prod_{ij} P(d_{ab}^{ij}|C); P(d_{ab}^{ij}) = \prod_{ij} P(d_{ab}^{ij}) \quad (2)$$

So the equation 1 can be rewritten as,

$$P(C|d_{ab}^{ij}) = P(C)\prod_{ij} \frac{P(d_{ab}^{ij}|C)}{P(d_{ab}^{ij})} \quad (3)$$

Considering $P(C)$ as a constant and taking the log of equation 3 and taking the negative of that, a scoring function is obtained as:

$$S(d_{ab}^{ij}) = -\sum_{ij} \ln \frac{P(d_{ab}^{ij}|C)}{P(d_{ab}^{ij})} \propto -\ln P(C|d_{ab}^{ij}) \quad (4)$$

In order to be able to calculate the score function, the terms $P(d_{ab}^{ij}|C)$ and $P(d_{ab}^{ij})$ are to be calculated. The terms are calculated directly from the experimental structures of protein as follows:

From a set of experimental structures, the observations of atom-atom contacts can be made. And also observations of atom-atom contacts in any particular distance bins can be made. The probability of observing atom types a and b in a particular distance bin d in a correct conformation can be written as:

$$P(d_{ab}|C) = f(d_{ab}) = \frac{N(d_{ab})}{\sum_d N(d_{ab})} \quad (5)$$

where $f(d_{ab})$ is the frequency distributions obtained from the experimental structures. $N(d_{ab})$ is the number of observations of atom types a and b in a particular distance bin d and $\sum_d N(d_{ab})$ is the total number of observations of atom types a and b for all distance bins d.

Similarly, assuming that averaging over different atom types in an experimental conformation sufficiently represents a random arrangements of atoms, we can express $P(d_{ab})$, the probability of finding atom

types a and b in a distance bin d in any compact structure can be expressed as:

$$P(d_{ab}) = P(d) = f(d) = \frac{\sum_{ab} N(d_{ab})}{\sum_d \sum_{ab} N(d_{ab})} \quad (6)$$

where $P(d)$ is the probability of observing any two atom types in a distance bin d, $\sum_{ab} N(d_{ab})$ is the total number of observations of atom types a and b in a particular distance bin d summed up over every atom types and $\sum_d \sum_{ab} N(d_{ab})$ is the total number of observations of all pairs of atom types a and b summed over all distance bins d.

Hence the final form of the equation becomes:

$$S(d_{ab}) = -\ln \frac{N(d_{ab}) / \sum_d N(d_{ab})}{\sum_{ab} N(d_{ab}) / \sum_d \sum_{ab} N(d_{ab})} \quad (7)$$

6.5 Generation of Score-table

For learning purpose of the discriminatory function, 200 proteins from PDB with more than 2$\mathring{A}$ were selected . In order to remove the bias in the learning set, only those proteins with resolution $\geq 1.8\mathring{A}$ and mutual sequence of less than 20% sequence identity were selected.

These proteins are selected by using the tool PDBselect of the PDB data base. Using equation 7, and a set of 200 experimental structures of proteins the negative log score is calculated. Altogether 167 atom types are considered and while considering atom types only all non-hydrogen atoms are considered.

Moreover, the atom types considered are residue specific i.e. a atom of one residue is different from the same atom of another residue.

For distance bins, the distances observed was divided into 18 distance bins in such a way that distance ranging from 0.0$\mathring{A}$-3.0$\mathring{A}$ are placed in one bin and from 3$\mathring{A}$-20$\mathring{A}$ are placed in 17 different bins with 1.0$\mathring{A}$ ranges.

Using above equation, table of negative log conditional probability is compiled for all possible pairs of the 167 atom types for the 18 distance ranges.

6.6 Maximum edge-weight Clique

An overview of maximum edge-weight clique algorithm(Suzuki et al., 2002) is given. As discussed in the previous section of clique, given a graph $G = (V, E)$ where V represents the set of vertices and E represents the set of edges, if in the subset G' (of G) every two vertices are adjacent, then the subgraph G' is called the clique of graph G. A clique with maximum number of vertices is called the maximum clique. Let us take ω as the number of vertices in the maximum clique, and suppose that the weight between two vertices p, $q \in V$ is defined as $w(p, q)(= w(q, p))$, then the weight of the clique $G'(= (V', E'))$ of graph G is represented as $W(V')$.

In a graph G, it is not necessary that it contains only one maximum clique so let us take the number of maximum clique to be $N(\omega)$. The maximum clique with the total weight of edges being the maximum is called as the maximum weight clique.

6.7 Clique Algorithm WC

Let us call the edge-weighted version of the maximum clique that is discussed in earlier section as WC. The readers are request to refer to (Suzuki et al., 2002) for the details of this algorithm.

MCQ is an algorithm to extract one maximum clique from a given non-directed graph. This algorithm extracts a clique by searching in a depth-search-first fashion. If all the candidates nodes are enumerated, the running time of algorithm worsens. Therefor, the upper limit of the size of each clique is calculated obtained by enumerating each candidate vertices by using approximate coloring. This approximate coloring corresponds to the NUMBERING-ARRANGING sub-routine of the algorithm and on the basis of this information, the branch and bound condition is ascertained.

In WC, all the maximum cliques of the given graph are extracted and their total edge-weights are compared and the clique with the largest weight is returned as an output. In MCQ, the set consisting of candidate vertices in the process of searching is Q, the number(i.e. the upper limit of the size of the clique that can be obtained by searching p) obtained by NUMBERING_ARRANGING is No(p) and the set of vertices that holds the clique with the maximum number of vertices till this point is Q_{max}.

By changing the branching condition from $|Q| + No(p) > |Q_{max}|$ of MCQ to $|Q| + No(p) \geq |Q_{max}|$, even other maximal cliques can be extracted with same number of vertices.

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 $W(Q) > W(Qmax)$, 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.

WC essentially finds all the maximum cliques in the given graph and returns the clique with the maximum weight as output.

As defined above, the basic algorithm finds a maximum clique incrementally using a recursive procedure. It maintains variables Q, Q_{max} and R. Q consists of the vertices of the current clique, Q_{max} consists of the vertices of a maximum clique found so far and R consists of the candidates of vertices which may be added to Q. In order to skip the recursive steps on R, R is not expanded if $|Q| + \max\{No(p)|p \in R\} \geq |Q_{max}|$.

For the calculation of weights on edges, $W(Q)$ (weight of Q before adding a new vertex) is assigned to W_{pre} so that in order to retrieve the weight of the clique one step prior to the current step W_{pre} can be used. This step enables us to make the weight calculation efficient.

In the process of calculation if $W(Q) > (Q_{max})$, the current maximum clique Q_{max} is upgraded with Q thus, upgrading the current maximum weight clique. Finally, the Q_{max} obtained after all the maximum cliques are extracted corresponds to the maximum weight clique.

Essentially, the algorithm WC runs as follows:

Procedure-WC($G = (V, E)$)
begin
  Q:= null; Q_{max}:= null;
  $W(Q)$:= 0; W_{Qmax} := 0;

```
    Sort vertices of V in non-increasing order
    with respect to their degrees;
    NUMBERING-ARRANGING(V,No);
    EXPAND-WC(V, No)
    output Q_max
  endof WC

Procedure EXPAND-WC(V, No)
begin
  while V ≠ φ do
    p:= a vertex in V such that
      No(p) = MaxNo(q)|q ∈ R
    if |Q| + No(p) ≥ |Q_max| then
      W_pre := WQ;
      for i := 1to|Q| do
        W(Q) := W(Q) + w(p, Q[i]);
      od
      Q := Q ∪ p
      R_p := R ∩ Γ(p);
      if R_p ≠ φ then
        NUMBERING-ARRANGING(R_p, Vo')
        EXPAND-WC(R_p, Vo')
        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
endof EXPAND-WC
```

6.8 Changing Branching Conditions using problem definition

From the initial assumption, in protein side-chain packing exactly only one conformations from each side-chain appear in the final clique and the number of vertices in the maximum clique is equal to the total number of amino acid residues. Hence, in protein side-chain packing, the maximum number of vertices(ω) to be included in the final maximum clique is known beforehand. Hence, the above version of the clique algorithm is modified to incorporate the information that the number of vertices in the maximum clique ω is already known. Thus, the branching of the above algorithm was changed as follows: $|Q| + No(p) \geq \omega$. This follows from the fact that after enumerating a vertex p, the upper limit of the maximum clique is $|Q_{max}|$ and $|Q_{max}| < \omega$, from the branching condition one is supposed to enumerate p but since the maximum number of vertices in the clique is ω, it is unnecessary to enumerate the vertex p.

By doing this, one can significantly reduce the number of vertices to be enumerated and hence the efficiency of the algorithm can be increased.

It is to be noted here that finding a maximum weight clique is not as easy as finding a maximum clique in any graph. This is due to the combinatorial nature of the possible number of maximum cliques. In the case of finding a maximum clique, finding a single maximum clique corresponds to finding a maximum clique but in order to find the maximum weight clique, it is required that every maximum clique is analyzed and the maximum clique with the largest weight is returned. Hence, finding a maximum weight clique in a graph is computationally very intense problem.

6.9 Implementation Issues

Implementation of the side-chain packing algorithm SPMCQ and SPWCQ was done in C language and the program was run in ORIGIN 3800.

As mentioned in the description of the maximum weight clique algorithm, in the case of protein side-chain packing the number of vertices in the maximum clique (value of ω) is known because from our initial reduction of the problem, the number of residues of the protein is equal to the number of vertices in the maximum clique. Hence, the version of the algorithm with known omega was utilized for the computation of protein side-chain packing in SPWCQ.

Besides, contrary to the maximum clique finding problem in which there is need to find only one maximum clique, for finding the maximum weight clique it is required to enumerate all the cliques and find the clique with the largest weight. Hence, the number of possible cliques for a protein increases as the number of its residues increases. Therefore, in order to find the maximum weight clique of a large protein it is required to limit the number of possible cliques. Although restricting the number of the maximum cliques could render the possible solution not an optimal one, but for practical purposes the solution can be considered a global maximum.

Hence in our computation, number of the maximum cliques was restricted to some specific values depending on the size of the protein. In order to know the change in the weight of the clique from the non-weighted version to the weighted-version, the maximum clique finding algorithm (MCQ) was embedded inside weighted version of the clique (WC) such that initially the maximum clique of the problem is calculated along with the assigned weights. After this, the number of edges in this maximum clique is passed on to the WC and then the enumeration of all the cliques is performed and then the clique with the maximum weight is returned as an output.

6.10 Input & Output

In both the weighted and non-weighted version of the clique, the input to the program is the protein structure file obtained from Protein Data Bank(PDB). The condition for the collision of side-chain with the main-chain and one side-chain with another side-chain is also entered as the initial parameter in the command line.

7 Results

7.1 Criteria Used to Assess Prediction Accuracy

The comparison of side-chain modeling methods is complicated by the different criteria used by different authors to assess the accuracy of their predictions. Two of the most common assessing methods are the comparison of predicted conformations to the X-ray structures obtained from the Protein Data Bank by calculating the root mean square deviation (RMSD) of the side-chain atom positions and comparison of side-chain dihedral angles with the native structure.

Most researchers consider a χ angle to be correctly predicted if it falls within 40° of the dihedral angle found in the crystal structure. However, there are some groups which have used 30° instead of 40°.

While comparing RMSD also some others have selected the C_β angle while others have excluded. So it is to be noted here that it is not easy to compare the results of different side-chain packing methods as the criteria used varies.

7.2 Assessing the predictivity: RMSD of Side-chain Atoms

For assessing the prediction quality of the weighted version of the clique, side chain building on the nine proteins was performed as in table below. The calculation for RMSD and the side-chain torsion angle is also performed. As mentioned above, specific value for the number of the maximum number of cliques to be searched was chosen. The comparison of RMSD obtained by SPMCQ (non-weighted version of the clique), SPWCQ (weighted version of the clique), the methods by (Holm & Sander, 1992) and the methods of (Lee et al., 1991) is summarized in the table 1.

Table 1: Comparison of SPWCQ with other methods: RMSD

PDB	SPWCQ	SPMCQ	H&S	Lee&S
1crn	0.81	1.16	-	1.65
5pti	1.37	1.44	1.90	1.49
1ctf	0.86	1.41	1.70	1.86
7rsa	1.15	1.42	1.80	1.86
1lz1	0.99	1.80	1.60	1.62
3fxn	0.90	1.30	1.90	1.90
3app	1.43	1.87	1.40	1.22
2cro	0.96	1.20	2.30	2.39
3tln	1.56	1.57	1.70	-

It can be observed that for all 9 proteins SPWCQ outperforms other methods compared in this experiment. The best prediction is obtained in case of 1crn (RMSD 0.81$\mathring{A}$) and the worst prediction is obtained in case of 3tln (RMSD 1.56$\mathring{A}$). It is to be noted that even in the worst case the RMSD is not greater than 1.57$\mathring{A}$.

7.3 Percentage Error in Torsion Angles

Similarly, the results of performing the experiments for determining the percentage error in χ_1 angle, on above test set using SPWCQ and the comparison of the results with the methods of (Dunbrack et al., 1993), (Holm & Sander, 1992) and our non-weighted version (SPMCQ) are shown in table 2.

Table 2: Comparison of SPWCQ with various methods: deviation in side-chain angles

PDB	$\#\chi_1$	D&K	SPMCQ	H &S	SPWCQ
1crn	32	8	21	-	10
5pti	42	15	18	22	5
1ctf	46	-	11	19	2
7rsa	105	21	15	21	2
1lz1	103	23	27	12	1
3fxn	115	-	93	39	3
3app	247	-	12	19	2
2cro	52	-	4	43	2
3tln	246	26	31	23	1

It can be seen that the results obtained by SPWCQ showed percentage error in χ_1 torsion angles in the range of 1%-10%. We were very surprised by the amazing results obtained for proteins like $1ctf$, $1lz1$, $3tln$, $2cro$ and $3app$ as the % error was very low. Hence, in order to check the correctness of our method we changed the value of error cutoff from 40% to 30% and examined the results.

On changing the value of cutoff from 40% to 30%, the value of deviated angles changed from 2 to 44. Furthermore, we could observe that out of 46 torsion angles, the 44 angles which were predicted to be errors in case of 30% cutoff, had angle of deviation ranging from 33° to 37°. Hence, it can be concluded that all the 42 angles that were predicted to be correct when the cutoff was 40% were predicted incorrect because the cutoff was changed to 30%.

7.4 Assessment of Consistency

In order to examine the consistency of our methods and for the comparison of SPMCQ and SPWCQ, the side-chain conformations of 3fxc protein and 5cpv protein was generated starting from a random initial state 10 times and the results of the experiment are summarized in figure 1 .

In order to make the graph more prominent, two lines are drawn corresponding to the optimum RMSD of each of the two proteins.

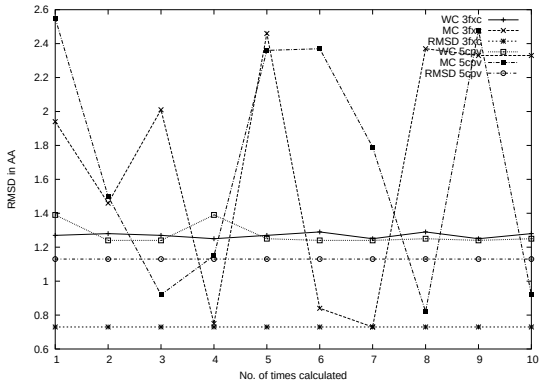

Figure 1: Comparison of RMSD for proteins 3fxc and 5cpv by using SPMCQ Q and SPWCQ

It can be seen that the curve for SPWCQ is more consistent compared to the curve of SPMCQ. This is in harmony with our assumption that the weighted version of the algorithm represent the side-chain packing problem more efficiently.

7.5 Change of Weight from SPMCQ to SPWCQ

For the comparison of weights of the final cliques generated by SPMCQ and SPWCQ, the side-chain conformations of 2ovo protein starting from a random initial state for 10 times was generated using SPMCQ and SPWCQ (ω set to 900), and the results of the experiment are summarized in figure 2.

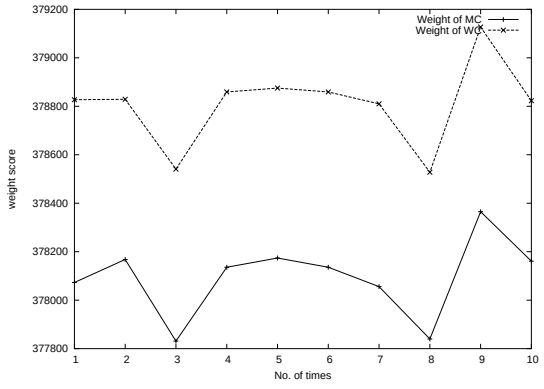

Figure 2: Relative Weight of 2ovo protein calculated by SPMCQ and SPWCQ

It can be seen that in each case, the weight of the clique obtained by SPWCQ is greater than the weight of the clique obtained by SPMCQ. But it is to be noted that the weight of the clique generated by SPWCQ is not the weight of the optimal clique .

In order to avoid the enumeration of all the maximum cliques, certain cutoff values according to the size of the proteins were selected. In overall, SPWCQ performs well in terms of RMSD of side chain atoms as well as in terms of deviation in side chain angles. Hence, it can be said that our algorithm can be applied to the proteins of realistic sizes.

8 Conclusion and Discussion

As mentioned in the introduction, our motivation for this research was to design an algorithm for the protein side chain packing problem which guarantees of finding an optimal solution and which is applicable to the proteins of real sizes.

Since we applied a branch and bound procedure for solving the problem, theoretically we were able to design an algorithm which guarantees an optimal solution. Especially in case of SPMCQ (non weight maximum clique approach), we were able to obtain the maximum clique of the reduced graph. But, whether the optimal solution in terms of the clique algorithm really correspond to the the optimal solution in terms of biochemical properties also is a question to be discussed.

In order to assess the prediction quality of our method, we collected a set of nine common proteins used in various methods. The results were better or as good as the existing methods for the % error in χ_1 angles when only SPMCQ was used. The percentage error in χ_1 angle was amazingly better than the existing methods in case of SPWCQ. We did not believe the results and then analyzed the results. To our surprise, the results were predicted correctly but the significant fact to be noted was that for almost all of the angles predicted to be correct deviated in the range of 30°-38°.

In addition to this, the results of the RMSD between the corresponding side-chain atoms was also better for almost all cases compared to the existing methods. The best prediction had an RMSD of $0.81\mathring{A}$ and the worst prediction had an RMSD of $1.56\mathring{A}$. Considering the fact that the data in the PDB database have some uncertainty, we can conclude that we were able to predict the side chain packing with significant accuracy.

But it is to be noted here that the values mentioned in the case of SPWCQ are not the optimal ones, in the sense that we restricted the number of maximum cliques to be analyzed due to computational reasons. In this regard, we may even obtain better results than we presented in this paper.

In terms of the size of protein, unlike the most branch and bound based methods we were able to apply our algorithm to a protein of upto 323 residues long. Compared to the branch and bound method of Leach(Leach et al., 1998) or Desmet(Desmet et al., 1992), we were able to extend the size of proteins that can be solved by our method up to a great extent.

While designing the weighted version of the method, we had assumed that the predictivity would increase if some biochemical properties of the interacting atoms were embedded in the method. As per our initial suspicion, we could get better results by applying SPWCQ than SPMCQ, which also proved that only geometrical constraints are not sufficient in order to depict the biochemical world.

Moreover, we did not find any correlation between the length of protein and the RMSD between the predicted structure and the native structure, suggesting that it is not the length of the protein which determines the structure of a protein rather it is the atoms of the proteins that determine the structure.

Hence, with some limitations we were able to fulfill our initial aim. The findings of this study are restricted to protein side-chain packing step in homology modeling.

9 Limitation and Future Work

One of the major problem we encountered in doing this work is the format of data in the PDB data set. In PDB data bank, the dates are not well formatted, especially for the name of atoms, the length varies and it was difficult to design a parser to read those dates correctly. Sometimes, manual intervention was required in order to be able to read those dates by our program.

Another limitation was the weight function that we used to assign weights to edges of the generated graph. Since, the main goal of this research was to develop a deterministic algorithm for protein side-chain packing problem, we did not focus more on designing our own potential functions and thus decided to use the existing potential function (Samudrala & Moult, 1998). Although, our preliminary results showed that the weight of the clique obtained by SPWCQ is always greater than the weight of the clique obtained by SPMCQ, it is not guaranteed that this is always true. Moreover, this distance-dependent discriminatory function also has some limitations, like in formulating the function, it has been assumed that the individual probabilities are independent of each other, which may not be the case in the true sense.

Besides, Lu *et al.* (Lu & Skolnick, 2001) have shown that the parameters like range, the bin size and the types of interaction centers are the critical parameter in these type of distance-dependent potential function. Hence, from this point of view also, the discriminatory function may not be optimal. Besides, it has also been shown that the known structures used to derive distance-dependent potentials also has some influence in the overall performance of the discriminatory function. Mainly, the dependence of discriminatory function on the size of the known structures used to extract the potential cannot be ignored. Since, we did not consider these various influences of parameters in our calculation of score table, our predictivity of the method may increase provided that we take in account all these critical parameters.

Moreover, we have recently discovered in the literature that a distance-dependent atomic knowledge based potential by Lu & Skolnick (Lu & Skolnick, 2001) , very similar to the one that we have utilized in our calculation, performs better than the one we utilized in our current work(Samudrala & Moult, 1998). Hence, implementing this distance-dependent atomic knowledge based potential can be the next step of our research.

Although it is computationally intense, we think owing to the fact that our center has a good resources for computation, more realistic force-field potential functions or a combination of different model features is also worthwhile. Furthermore, we may also optimize residue level statistical potential like distance-dependent potential, contact-potential, ϕ-ψ dihedral angle potential and accessible surface statistical potential like in Melo *et al.*(Melo, Sanchez & Sali, 2002).

Another important problem encountered was the size of the generated graph itself. The generated graph was usually dense as the remote sidechains do not collide with each other. This resulted in the limitation of size of proteins that we were able to deal

with our algorithm. One of the possible solution to reduce the size of the graphs would be dividing the proteins into smaller fragments and then finding the maximum clique in these separate fragments and finally assembling these cliques.

Moreover, the clique finding algorithm itself has some limitations. Especially, in the case of weighted version of the clique, we had to limit the number of maximum cliques to be analyzed in order to get the maximum weight clique. The designing of a new energy function to assign weight, the designing of a method to reduce the density of the obtained graph and refining the clique algorithm are important future works.

Designing an approximation algorithm for the maximum edge-weight clique algorithm and usage of real rotamer library are the two immediate future works.

10 Acknowledgments

We would like to thank Prof. Minoru Kanehisa for providing us the computational resources to undertake this project and Prof. Toh Hiroyuki for helpful discussions and suggestions. This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas(C) "Genome Information Science" and Grant-in-Aid #13680394 from the Ministry of Education, Science, Sports and Culture of Japan.

References

Alberts, Johnson & Walter (2002), *Molecular Biology of the CELL*, Garland Science, Fourth Edition.

Akutsu, T. (1997), NP-hardness results for protein side-chain packing, *in* 'Genome Informatics', Vol. 8, pp. 180-186.

Bron, C. &Kerbosch,J. (1973), 'Algorithm 457: Finding all cliques of an undirected graph', *in* Comm. ACM, Vol. 16 , pp. 575-577.

Chazelle, B., Kingsford, K. & Singh, M. (2003), The side-chain Positioning-problem: A semidefinite Programming Formulation with new rounding schemes, *in* 'Proc. ACM FCRC'2003, Principles of Computing and Knowledge: Paris Kanellakis Memorial Workshop, 2003, pp. 86-94.

Desmet, J., De Maeyer, M., Hazes, B. & Lasters, I. (1992), 'The dead-end elimination theorem and its use in protein side-chain', *Nature* **356**, 539-542.

Dunbrack, R.L. & Karplus, M. (1993), 'Backbone-dependent rotamer library for proteins: Application to side-chain prediction', *J. Mol. Biol.* **230**, 543-574.

Holm, L. & Sander, C. (1992), 'Fast and simple Monte Carlo algorithm for side-chain optimization in proteins: application to model building by homology',*Proteins: Struct. Funct. Genet* **14**, 213-23.

Hwang, J.K. & Liao, W.F. (1995), 'Side-chain prediction by neural networks and simulated annealing', *Protein Engineering* **8**, 363-370.

K.C., Dukka, B., Akutsu, T., Tomita, E., Seki, T. & Fujiyama, A. (2002)Point matching under non-uniform distortions and protein side-chain packing based on an efficient maximum clique algorithm, *in* 'Genome Informatics' Vol. 13, pp. 143-152.

Koehl,P. & Levitt, M. (1999)'De Novo Protein Design: In search of stability and specificity', *J. Mol. Biol.* **293**, 1161-1181.

Laughton, C.A (1994)'Prediction of protein side-chain conformations from local three-dimensional homology relationships', *J. Mol. Biol* **235**, 1088-1097.

Leach, A. R. & Lemon, R.P. (1998), 'Exploring the conformational spae of protein side-chains using Dead-end elimination and the A* Algorithm', *Prot. Struct. Funct. Genet* **33**, 227-239.

Lee, C. & Subbiah, S. (1991)'Prediciton of Protein side-chain conformation by packing optimization', *J. Mol. Biol.* **217**, 373-388.

Levitt, M. (1991)'Accurate modeling of protein conformation by automatic segment matching', *J. Mol. Biol***226**, 507-533.

Lu, H. & Skolnick, J. (2001), 'A distance-dependent Atomic Knowledge-based Potential for Improved Protein Structure Selection', *Proteins: Struct. Funct. Genet.***44** 223-232.

Melo, F., Sanchez, R. & Sali, A. (2002), 'Statistical potentials for fold assessment', *Protein Science***11**, 430-448.

Pierce, A. Niles & Winfree, E. (2002)'Protein Design is NP-hard', *Protein Engineering***15** no.10 779-782.

Ostergrad, P.R.J, (2002), 'A fast algorithm for the maximum clique problem', *Discrete Appl, Math.* **120**, 197-207.

Samudrala, R. & Moult,J. (1998), 'Determinants of side-chain conformational preferences in protein structure',*Protein Engineering* **11**, 991-997.

Samudrala, R. & Moult,J. (1998), 'An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction', *J. Mol. Biol.* **275**, 893-914.

Tomita, E. & Seki, T. (2003), An efficient Branch-and-Bound Algorithm for Finding a Maximum Clique, *in* 'DMTCS 2003', LNCS 2731, pp. 278-289.

Tomita, E. & Seki, T. (2002), An efficient branch-and-bound algorithm for finding a maximum clique and computational experiments, *in* 'Technical Report',UEC-TR-CAS7, The university of Electro-communications.

Suzuki,J., Tomita, E. & Seki, T., (2002), An algorithm for Finding a Maximum Clique with Maximum Edge-Weight and Computational Experiments, *in* 'Technical Report', MPS, The Information Processing Society of Japan.

Vasquez, M. (1995), 'An evaluation of discrete and continuum search techniques for conformational analysis of side-chains in proteins', *Biopolymers* **36**, 53-70.

Voigt, C.A., Gordon, D.B. & Mayo, S.L. (2000), 'Trading accuracy for speed: A qualititive comparison of search algorithms in protein sequence design', *J. Mol. Biol.* **299**, 789-803.

http://www.rcsb.org/pdb/