# University of Waterloo
# Technical Report CS-2007-35

# Rapid and Accurate Protein Side Chain Prediction Using Local Backbone Information Only

Jing Zhang[1,2⋆], Xin Gao[1⋆], Jinbo Xu[3⋆⋆], and Ming Li[1⋆⋆]

[1] David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 6P7
[2] The Institute for Theoretical Computer Science,
Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China
[3] Toyota Technological Institute at Chicago, Chicago, IL, USA, 60637
{j2zhang, x4gao, mli}@cs.uwaterloo.ca, j3xu@tti-c.org

**Abstract.** High-accuracy protein structure modeling demands on accurate and very fast side chain prediction since such a procedure must be repeatedly called at each step of structure refinement. Many known side chain prediction programs, such as SCWRL and TreePack, depend on the philosophy that global information and pairwise energy function must be used to achieve high accuracy. These programs are too slow to be used in the case when side chain packing has to be used thousands of times, such as protein structure refinement and protein design.
We present an unexpected study that local backbone information can determine side chain conformations accurately. LocalPack, our side chain packing program which is based on only local information, achieves equal accuracy as SCWRL and TreePack, yet runs many times faster, hence providing a key missing piece in our efforts to high-accuracy protein structure modeling.

**keyword:** side chain prediction, local backbone features, multiclass Support Vector Machines.

## 1 Introduction

Protein side chain packing is a key step towards accurate protein structure modeling and has been studied for three decades [27, 7, 33, 44]. Given the backbone conformation of a protein, side chain prediction determines the coordinates of all the side chain atoms. Accurate and very fast side chain prediction is vital to accurate protein structure modeling since such a procedure needs to be repeatedly called at each step of the entire protein structure refinement process, which usually samples a very large number of backbone conformations. Protein side chain packing is also an indispensable component of protein design, which finds a protein sequence that can fold into a given three-dimensional protein structure [14, 12]. Whenever a protein backbone conformation (in protein structure modeling) or its primary sequence (in protein design) is changed, side chain packing has to be conducted to re-determine the coordinates of the affected side chain atoms or even all the side chain atoms. Many known side chain prediction programs, such as SCWRL [16] and TreePack [51, 52], predict the positions of side chain atoms using global information and pairwise energy function, in order to achieve high accuracy. Thus these programs are too slow to be called tens of thousands of times in high-accuracy protein structure modeling or protein design. Therefore, an ultra-fast side chain prediction method is urgently needed.

[⋆] The first two authors contributed equally to this paper.
[⋆⋆] Corresponding authors.

An important discovery on side chain conformation is that the side chains have a few frequently occurred conformations (referred to as rotamers) [27, 33, 17, 50, 16]. Thus, most side chain prediction methods assume side chains can only take several highly probable rotamers, while others consider conformations sampled around rotamers.

*Problem Description.* Given a finite set of side chain rotamers for each amino acid type, and a backbone conformation $b$. Let $p$ denote a possible side chain conformation vector indicating the rotamer choice for each residue position. Traditional side chain prediction problem can be formulated as a combinatorial search problem:

$$p^* = \arg\min_p [E_{SS}(p,p) + E_{SB}(p,b) + E_{BB}(b,b)] \tag{1}$$

where $p^*$ denotes the optimal side chain conformation, $E_{SS}(p,p)$ is a pairwise energy item representing interactions among side chain atoms, $E_{SB}(p,b)$ represents interaction energy between side chain atoms and backbone atoms, and $E_{BB}(b,b)$ represents backbone-backbone interaction energy. Among them, $E_{BB}(b,b)$ can be considered as a constant since the backbone conformation is fixed.

Following this formulation, almost all side chain prediction methods employ a pairwise energy function and a rotamer library, then apply a global or local search strategy to find the optimal solution for this combinatorial problem.

*Rotamer Libraries.* A rotamer library is a finite set of rotamers, each of which has an occurring probability. Rotamer libraries can be either backbone-independent [9, 7, 6, 37, 29, 32] or backbone-dependent [27, 33, 18, 41, 19, 17, 16], according to whether the occurring probability of a rotamer is estimated based on backbone information. Chandrasekaran *et al.* developed the first backbone-independent library [9]. Janin *et al.* [27] and McGregor *et al.* [33] examined the relationship between side chain conformation and secondary structure and then developed a secondary-structure-dependent rotamer library. Dunbrack *et al.* developed the first backbone dihedral angle based rotamer library [18] and refined it by Bayesian statistical analysis [17].

Backbone-dependent rotamer library is widely used to predict side chain conformations [31, 8, 35, 10, 28, 51, 52, 26, 54]. Rotamer library not only can make side chain prediction a discrete-optimization problem, but also can provide the probability of each rotamer in energy function calculation. However, since many side chain prediction methods use rotamer probabilities in their energy functions, their performance is sensitive to these values which are hard to be estimated accurately.

*Energy Functions.* The energy functions are considered to be a bottleneck for existing side chain prediction methods. Although many studies aim to improve the accuracy of side chain packing energy functions [39, 43, 34, 31, 54], all side chain predictors claim that their methods can perform much better if the energy function is more accurate. As mentioned above, energy functions used in side chain prediction contain both side chain-backbone interaction energy and side chain-side chain interaction energy.

Roitberg *et al.* [39] used a mean field approximation, which probably has the same global minimum as the original system, to direct their search strategy. A much more accurate energy function was developed by Liang *et al* [31]. Their energy function contains contact surface, volume overlap, backbone dependency, electrostatic interactions, and desolvation energy. In [54], ROSETTA's energy function [38], which is the sum of Lennard-Jones potential, rotamer energy, atomic clash penalty, and hydrogen-bonding potential, was improved by the tree-reweighted belief propagation (TRBP) technique.

*Search Methods.* A large number of search methods have been developed to optimize the energy function and find the side chain conformation search with minimum energy, such as Metropolis

Monte Carlo [23], Gibbs sampling Monte Carlo [49], genetic algorithm [48], dead-end elimination (DEE) [15, 32], neural networks [25], simulated annealing [30, 25], graph theory methods [8, 51, 52], semidefinite programming [10], and integer linear programming [21, 28].

Besides the energy function, search strategy is another bottleneck for side chain prediction. The side chain prediction problem has been proved to be NP-hard [5, 36] if pairwise or multi-body energy function is used. Heuristics such as Monte Carlo or genetic algorithm can find local minimum relatively quickly, but cannot guarantee to find the optimal solution of the energy function. On the other hand, some global search methods can find the global optimal solution at the cost of running time. For example, the widely-used program, SCWRL3.0 [8], can optimize its energy function to its optimal by first decomposing a protein backbone structure into some substructures and than employing a divide-and-conquer strategy to determine the positions of side chain atoms. SCWRL is not fast enough to be used for iterative refinements and protein design. Another global search method, TreePack [51, 52], achieves similar accuracy as SCWRL3.0, but runs much faster. In contrast to SCWRL, TreePack can decompose a protein structure into much smaller substructures without losing accuracy, and thus reduce running time dramatically. However, both SCWRL and TreePack are likely to fail in the case when the backbone conformation implies heavy steric atomic clashes and thus cannot be cut into small substructures without losing accuracy.

In this paper, we present a study among the relationship between local backbone information and side chain conformations, and develop a side chain packing program LocalPack. LocalPack predicts the side chain conformations using local backbone information only and is as accurate as SCWRL, a program that uses pairwise energy function and global search method. We first reformulate side chain packing problem and then solve it using multi-class Support Vector Machines (multi-class SVM). Our method has the following three features: 1) Instead of using the occurring probabilities contained in a rotamer library, our method only use the angle values of rotamer candidates. 2) Our method does not use any pairwise energy function. Instead, only local backbone information is employed to predict side chain positions. Furthermore, these local backbone features can be calculated extremely fast. 3) We do not need to optimize an energy function. By contrast, our method generates a set of linear classifiers based on local backbone features and then use these classifiers to predict side chain positions.

The rest of this paper is organized as follows: In Section 2, we introduce our new formulations of side chain prediction problem. Section 3 describes our multi-class SVM model and the features used to construct the classification rule. A cutting plane algorithm is proposed to obtain solutions to the multi-class SVM. In Section 4, we present some experimental results and compare our method to existing methods on both native and nonnative backbones. We also analyze the relative importance of the features in our model. Finally, Section 5 makes some discussions.

## 2 New Formulation for Side Chain Prediction

Given a position on a protein backbone sequence, we can calculate a set of backbone related local features on this position. Starting from a rotamer library, our basic assumption is that a certain set of local features can determine the correct rotamer of the side chain on this position.

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ denote the set of feature vectors for a given protein with length $n$, where vector $A_j = \{a_1^j, a_2^j, \ldots, a_k^j\}$ denote the set of backbone related features on the $j$-th position, either continuous values, such as solvent accessibility, or discrete values, such as secondary structure and amino acid type. Let $\mathcal{R} = \{r_1, r_2, ...., r_m\}$ denote an arbitrary rotamer set. Table 1 shows some examples of feature vectors, according to which the rotamer choice for each residue position is determined.

**Table 1.** An example of the basic assumption of this paper: a backbone related feature vector $A$ can determine the rotamer choice. Except for the last column, the first 6 columns show examples of possible backbone related feature vectors. The last column shows $\chi_1$ rotamer values corresponding to the feature vectors.

| Residue Type | $\phi$ | $\psi$ | Secondary Stru. | Solvent Access. | # Contacts | $\chi_1$ Rotamer |
|---|---|---|---|---|---|---|
| ARG | 60° | 45° | Helix | 82.75% | 11 | 63° |
| PHE | 112° | 42° | Helix | 10.23% | 4 | 114° |
| GLN | 34° | 16° | Loop | 8.65% | 6 | 125° |
| MET | 156° | 107° | Sheet | 65.22% | 19 | 178° |

Based on our assumption, given a rotamer set $\mathcal{R}$, we can consider side chain predictor as a function $f(A_j)$ that maps from a given feature vector $A_j$ to a rotamer. $f(A_j)$ is defined to be

$$f(A_j) = \arg\max_{i,r_i \in R} h(A_j, r_i), j = 1, \ldots, m \tag{2}$$

in which $h(A_j, r_i)$ is a score function that evaluates the score of assigning the rotamer $r_i$ to the $j$-th position with feature vector $A_j \in \mathcal{A}$. We aim to find a function $h(A_j, r_i)$ such that $f(A_j)$ matches the correct rotamer choices as well as possible for all the position $j$.

The formulation 2 is based on a general rotamer library $\mathcal{R}$. Studies on backbone-dependent rotamer libraries [18, 19, 17, 16] show that side chains do prefer some rotamers for a fixed amino acid type and a fixed pair of $\phi$, $\psi$ backbone dihedral angles. This kind of rotamer libraries can also fit into our model easily by removing the features $(\textit{amino acid type}, \phi, \psi)$ from vector $A_j$ and finding $h$ on a rotamer library which is a $(\textit{amino acid type}, \phi, \psi)$-dependent subset of the original rotamer library $\mathcal{R}$. We will introduce how to find the score function $h$ in next section.

## 3 Multi-class SVM Model to Solve Side Chain Prediction Problem

### 3.1 Multi-class SVM Model

In this paper, we consider side chain prediction problem as described in formulation 2 that is a linear function on feature vector $A$. That is, $h(A_j, r_i) = w_i \cdot A_j$, where $w_i$ is a parameter vector for rotamer $i$ that we want to learn. Thus, according to formulation 2, side chain prediction problem can be formulated as a classification problem:

$$f(A_j) = \arg\max_{i,r_i \in R} w_i \cdot A_j, \quad j = 1, \ldots, n, \tag{3}$$

in which we want to find such a $f$ that matches correct rotamer choices as well as possible.

To learn the parameter vectors $w_i$ from a training example set $S = \{(A^1, r^1), \ldots, (A^p, r^p)\}$ with size $p$, where $A^j$ is the feature vector of a residue and $r^j$ is the experimentally determined rotamer of this residue, we applied a multi-class Support Vector Machine (multi-class SVM) model. Multi-class Support Vector Machines provide powerful approaches to deal with the general problem of learning a mapping from a high dimensional feature space to a discrete set[11]. However, traditional multi-class SVM do not directly fit into the side chain prediction problem. The reason is that the number of rotamer labels is usually very large in the real world, which will result in a large number of constraints in multi-class SVM. This will make the traditional quadratic programming based algorithm unfeasible to solve the side chain prediction problem.

To solve this large class problem, we borrowed the idea of loss function $\triangle$ from structured SVM[46, 47], a generalized version of multi-class SVM. Different from multi-class SVM, which were developed to solve classification problems on discrete set $\mathcal{Y} = \{1, \ldots, k\}$, structured SVM were developed to solve classification problems that involve features extracted jointly from the inputs and the outputs, such as sequences, strings, graphs, or labeled trees. Loss function $\triangle$ is widely used in structured SVM[46, 47] to deal with the case in which $|\mathcal{Y}|$ is large. In our side chain prediction problem, we borrowed the concept of loss function and defined it to be: $\triangle : \mathcal{R} \times \mathcal{R} \rightarrow \{0, 1\}$, where $\triangle(y', y)$ returns 1 if $y' = y$, and 0 otherwise. $\triangle(y', y)$ quantifies how "bad" it is to predict $y'$ when $y$ is the correct label.

Under the framework of multi-class SVM[11], we use loss function $\triangle$ to re-scale the margin as proposed by Taskar *et al.* [45] and formulate the problem of finding parameter vectors $w_i$, $i = 1, \ldots, m$ in the form of the following optimization problem:

$$\min_{w_i, \xi_j} \frac{1}{2} \sum_{i=1}^{m} \|w_i\|^2 + \frac{C}{p} \sum_{j=1}^{p} \xi_j \qquad (4)$$
$$\forall j, l \quad w_{r^j} \cdot A_j - w_l \cdot A_j \geq \triangle(l, r^j) - \xi_j$$

where $m$ is the size of rotamer library, $p$ is the size of training set, $\xi_j \geq 0$ are called *slack variables.* $\|w_i\|$ is the norm of vector $w_i$, which determines the size of margin in SVM. $C > 0$ is a tradeoff between training error minimization and margin maximization.

We then apply a cutting plane algorithm described in [46] to solve this optimization problem. The basic idea of the algorithm is to find a relatively small set of constraints without losing too much accuracy. They achieved this goal by building a nested sequence which successively tights relaxations of the original problem. It can be proved that:

- Accuracy: the cutting plane algorithm can compute arbitrarily close approximation to the optimal solution.
- Efficiency: the number of steps that the cutting plane algorithm needs to converge is polynomial on the number of data points.

In practice, the cutting plane algorithm works very well on solving our side chain prediction problem, which we will show later. For more details about the algorithm, please refer to [46].

### 3.2 Model Features

The relationship between side chain conformations and backbone dihedral angles ($\phi$, $\psi$) has been well studied. Many side chain prediction programs use a backbone-dependent or backbone-independent rotamer library. This work uses the backbone-dependent rotamer library [17, 16] developed by Dunbrack *et al.*. The major problem to be addressed is what kind of backbone structural features a side chain conformation depends on. Many works [33, 19, 20] have been done to analyze the relationship between side chain dihedral angles and local backbone features, such as backbone dihedral angles, secondary structure and solvent accessibility. Here we introduce the local structure features used in our prediction and show how to use them in training and test.

**backbone dihedral angles** Given an amino acid and a pair of ($\phi$, $\psi$) angles, the backbone-dependent rotamer library can provide a set of candidate side chain conformations. We do not use backbone dihedral angles as features in the training. Instead, we divide training data point into many groups according to the amino acid types and $\phi$, $\psi$ angles and develop a classifier for each group based on its corresponding rotamer group.

**secondary structure** Secondary structure is local conformation of a protein backbone. Previous works [33] have shown that secondary structure is highly relevant to the distribution of side chain dihedral angles. We use P-SEA [22] to calculate the secondary structure of a given protein backbone.

P-SEA can generate the secondary structure type for each backbone position. Since SVM can only take numerical values as input, we use the expected occurring probability of each secondary structure type as its feature value. Let $N(\alpha)$, $N(\beta)$, $N(loop)$ denote the numbers of $\alpha$-helices, $\beta$-sheets and loops in a training data set and $N$ their sum. The expected occurring probabilities are calculated as are $N(\alpha)/N$, $N(\beta)/N$ and $N(loop)/N$, respectively.

**solvent accessibility** The accessible surface area is the area of a biomolecule's surface that is accessible to a solvent. It can be calculated by using a sphere of a certain radius to probe the surface of the molecule. A typical radius value is $1.4\mathring{A}$, which approximates the radius of a water molecule. Solvent-accessible surface of atoms have been used to predict conformations of side chains in [20], where they added this term into the energy function during the global optimization and calculated it iteratively. Their results show that the prediction accuracy can be significantly improved by adding the solvent term. This implies the importance of solvent accessibility in modeling side chain conformations. We use Naccess [4] to calculate the backbone solvent accessibility. The output of Naccess is normalized value and we use it as our feature directly.

**contact number** The contact number of a residue in a protein structure is a quantity similar to, but different from solvent accessible surface area. The contact number of a given residue is defined as the number of $C_\alpha$ atoms within a predefined distance $D(= 8\mathring{A})$ to the $C_\alpha$ atom of this given residue. The contact numbers are scaled to values between 0 and 1 using a standard max-min normalization method, such that the smallest contact number becomes zero and the largest number becomes one.

## 4 Results

### 4.1 Implementation Details

We implemented our side chain prediction program with C++. To improve the efficiency of feature calculation, we used a quick K-nearest-neighbor (KNN) algorithm [13, 42] to calculate contact numbers. After extracting backbone related features, such as solvent accessibility, secondary structure, and contact number, we encoded these features into a multi-class SVM model as described in Section 3.1. The SVM model is trained using $SVM^{multiclass}$ [3] with linear kernel function, a program that solves multi-class SVM problem by applying cutting plane algorithm described in [46].

We applied 10-fold cross-validation on our training set to estimate the best $C$ (see Equation 4), a tradeoff between model parameter complexity and tolerable model training errors. A big $C$ indicates that a small training error is tolerated but a big model parameter complexity allowed. A model trained using such a $C$ may not generalize well to the test data. Hsu *et al.* showed in [24] that by testing on a sequence of exponentially growing $C$ values, a good model can be identified in practice. Thus, we tried $C = 2^{-5}, 2^{-4}, ..., 2^{20}$ for each training case, and determined its best $C$ value.

### 4.2 Training and Test Set

Selecting reasonable training and test sets is very important for fairly evaluating the performance of machine learning methods. We used PDB20 as our training set, in which any two proteins do not share more than 20% sequence identity. We also removed those proteins in this set with resolution worse than $2\mathring{A}$. This results in a data set of 3060 proteins. For test set, we used Dunbrack's benchmark set [16], which consists of 180 proteins. Since we also uses the rotamer library extracted from a set of 800 proteins [17], we examined the overlap among PDB20, the set of 800 proteins for rotamer library generation, and Dunbrack's benchmark set. It turns out that Dunbrack's benchmark set contains 87 proteins in PDB20 and 102 in the set of proteins for rotamer library generation. Thus, we removed all the overlapping proteins from Dunbrack's benchmark set and obtain a reduced benchmark set of 78 proteins. It can be seen from Fig.1 that both our PDB20 training set and the reduced test set are good samples of real world proteins.

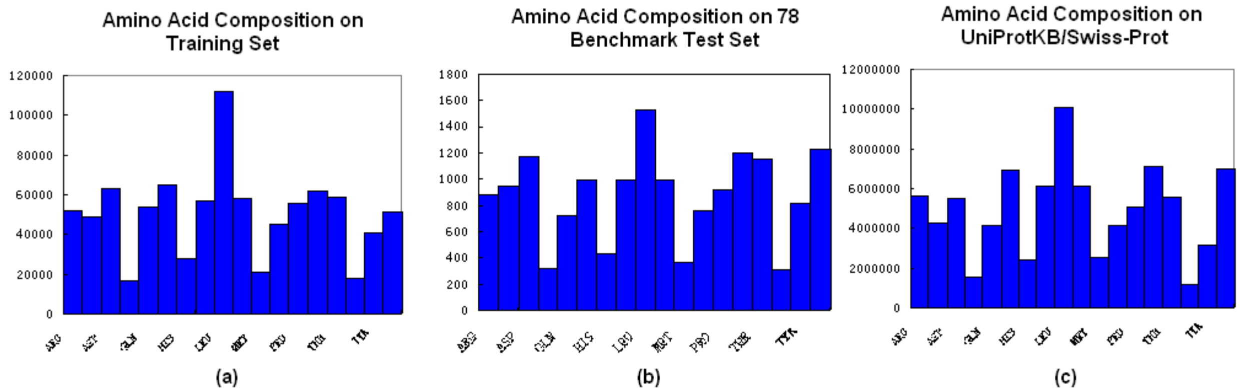


**Fig. 1.** The amino acid compositions on PDB20 training set(a), reduced 78 benchmark test set(b), and the UniProtKB/Swiss-Prot protein knowledgebase(c), respectively. UniProtKB/Swiss-Prot protein knowledgebase[1] is one of the largest protein sequence databases. The statistics of UniProtKB/Swiss-Prot was taken on 283454 protein sequences on Sep.11, 2007.

We evaluated the performance of our method on both this reduced benchmark set and Dunbrack's original benchmark set which has overlapping proteins to our training set. Not surprisingly, the accuracy of our method is approximately 8% higher on the Dunbrack's benchmark set than on the reduced set, while the accuracy of SCWRL3.0 is consistent on the two benchmark sets. Thus, in the following experimental studies, we will only evaluate our method on this reduced benchmark set.

### 4.3 Prediction Accuracy on Native Backbones

We compared the accuracy of our method to the most widely used method SCWRL3.0 in terms of $\chi_1$ and $\chi_{1+2}$. Other widely used programs, such as Modeller [40], SCAP [50], and TreePack [51, 52], performs no better than SCWRL3.0 on both the 180 benchmark set and the 78 benchmark set. Due to the page limitations, we only show the comparison between our method and SCWRL3.0 in Table 2. A prediction is considered to be correct if its value is within 40° from its experimental value. The prediction accuracy of one amino acid is calculated as the ratio of the number of correctly predicted side chains to the total number of side chains of this amino acid type.

As shown in Table 2, the overall accuracy of our method is very close to that of SCWRL3.0. In fact, the $\chi_1$ accuracy of our method is only 0.61% lower than that of SCWRL3.0, while the $\chi_{1+2}$ accuracy is 0.51% lower. Although our method is based on local backbone information only, it does not lose any accuracy while is much more computationally efficient, which we will show later. In fact, the $\chi_1$ accuracy of our method is higher than SCWRL3.0 for nine out of the eighteen amino acids, especially LYS, SER and THR. However, our method is much worse than SCWRL3.0 for CYS, LEU, PHE and TRP. Meanwhile, the $\chi_{1+2}$ accuracy of our method is higher than SCWRL3.0 for eight out of the eighteen amino acids. This means local backbone information can also determine $\chi_2$ conformation accurately. On the other hand, results shown in Table 2 also demonstrate that the accuracy of our method is not worse than any global optimization methods.

We further examined the eight amino acids on which our method did not perform well (with $\chi_1$ accuracy $\leq$ 82%). They are ARG, ASN, GLN, GLU, LEU, LYS, MET and SER. Except for

**Table 2.** Prediction accuracy of our method and SCWRL 3.0 on the 78 benchmark set. A prediction of a side chain is correct if its deviation from the experimental value is no more than 40°. $\chi_1$ accuracy of one amino acid is the ratio of the number of correctly predicted $\chi_1$ angles to the total number of this amino acid type, while $\chi_{1+2}$ accuracy of one amino acid is the ratio of the number of side chains with both $\chi_1$ and $\chi_2$ being predicted correctly to the total number of this amino acid type.

| | Our Results | | SCWRL 3.0 | |
|---|---|---|---|---|
| amino acid | $\chi_1$ accuracy | $\chi_{1+2}$ accuracy | $\chi_1$ accuracy | $\chi_{1+2}$ accuracy |
| ARG | 0.7701 | 0.6060 | 0.7558 | 0.6226 |
| ASN | 0.7888 | 0.7011 | 0.7956 | 0.6882 |
| ASP | 0.8322 | 0.7337 | 0.8218 | 0.6974 |
| CYS | 0.8497 | 0.8497 | 0.8915 | 0.8915 |
| GLN | 0.7493 | 0.5416 | 0.7449 | 0.5319 |
| GLU | 0.6841 | 0.5077 | 0.7084 | 0.5128 |
| HIS | 0.8226 | 0.7551 | 0.8382 | 0.7745 |
| ILE | 0.9172 | 0.7884 | 0.9114 | 0.8060 |
| LEU | 0.7851 | 0.7321 | 0.8996 | 0.8142 |
| LYS | 0.7678 | 0.5768 | 0.7199 | 0.5444 |
| MET | 0.8169 | 0.6097 | 0.8160 | 0.6720 |
| PHE | 0.8410 | 0.7740 | 0.9361 | 0.8774 |
| PRO | 0.8426 | 0.7701 | 0.8517 | 0.7879 |
| SER | 0.7556 | 0.7556 | 0.6883 | 0.6883 |
| THR | 0.9193 | 0.9193 | 0.8855 | 0.8855 |
| TRP | 0.8328 | 0.6851 | 0.8843 | 0.6688 |
| TYR | 0.9239 | 0.8616 | 0.9171 | 0.8615 |
| VAL | 0.8922 | 0.8922 | 0.9075 | 0.9075 |
| overall | 0.8205 | 0.7314 | 0.8266 | 0.7365 |

SER, all the other seven amino acids have large side chain groups as shown in Fig. 2. This result is consistent with the model on which our method is built. Our method assumes that local backbone information can determine side chain conformations. However, if a side chain group is large, its position will be more likely to be impacted by other side chain groups around it and thus cannot be completely determined using only local information. Thus, for such cases, we probably need more information to determine side chain conformations. Interestingly, the global optimization method, SCWRL3.0, which considers all side chain and backbone atoms around one side chain, did worse than our method on four out of these seven amino acids as shown in red boxes in Fig. 2.

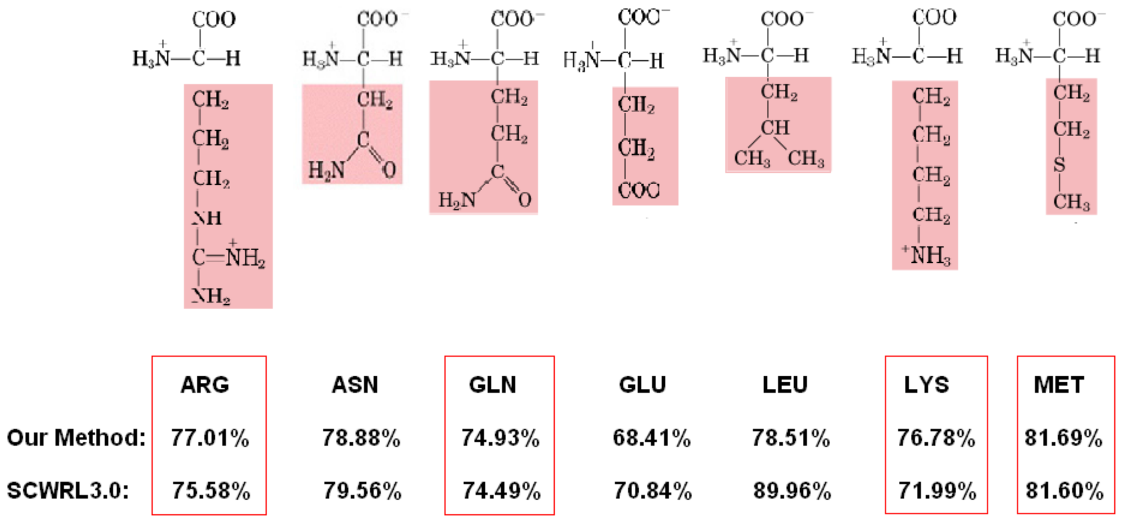
### 4.4 Feature Importance Analysis

A key step in feature based machine learning study is to evaluate the importance of each feature encoded. We evaluated the importance of each feature by removing it from the whole set of features, and testing the accuracy on the rest feature set. Table 3 shows the $\chi_1$ accuracy on different feature sets on amino acid arginine (ARG). The comparisons on other amino acids or on $\chi_{1+2}$ are similar. Due to the page limits, we only show the results on $\chi_1$ accuracy of ARG here.

**Table 3.** Feature importance analysis on ARG. The 1st column is the $\chi_1$ accuracy of our method with all 3 features. Starting from the 2nd column, the $\chi_1$ accuracy on feature sets without solvent accessibility, without secondary structure, and without contact number are listed, respectively.

| | with all 3 features | without solvent accessibility | without secondary structure | without contact number |
|---|---|---|---|---|
| $\chi_1$ Accuracy | 0.7701 | 0.7226 | 0.7352 | 0.7320 |

It can be seen from Table 3 that all of the three features are important to our method. More specifically, removing solvent accessibility feature will reduce the accuracy by 4.8%, while removing



|  | ARG | ASN | GLN | GLU | LEU | LYS | MET |
|---|---|---|---|---|---|---|---|
| Our Method: | 77.01% | 78.88% | 74.93% | 68.41% | 78.51% | 76.78% | 81.69% |
| SCWRL3.0: | 75.58% | 79.56% | 74.49% | 70.84% | 89.96% | 71.99% | 81.60% |

**Fig. 2.** The $\chi_1$ accuracy of our method on amino acid types ARG, ASN, GLN, GLU, LEU, LYS, and MET. The four amino acids on which the accuracy of our method is higher than that of SCWRL3.0 are marked in red boxes.

secondary structure and contact number will reduce the accuracy by 3.5% and 3.8%, respectively. This means that solvent accessibility is the most important feature in our method, while secondary structure is the least. This makes sense becuase the backbone-dependent rotamer library [17] we used has already partially encoded secondary structure information by considering backbone $\phi$, $\psi$ angles in their statistics.

### 4.5 Performance on Non-native Backbones

We further evaluated the accuracy of our method on nonnative backbones. We compared the $\chi_1$ accuracy of our method to four commonly used side chain prediction methods: MODELLER, TreePack, SCWRL3.0, and SCAP, on a nonnative backbone test set provide by Xu *et al.* in [52]. The test set contains prediction models generated by a protein threading program, RAPTOR [53], on 24 CASP6 test proteins [2]. RAPTOR generated good alignments for most of these targets. MODELLER [40] was called by RAPTOR to generate model backbones according to the alignments. Besides, MODELLER is also able to predict side chains based on a statistical method. SCAP was tested using the CHARMM force field with the heavy atom model and the largest rotamer library available to SCAP.

The overall $\chi_1$ accuracy is shown in Table 4. The prediction accuracy of our method is the same as TreePack, and slightly worse than SCWRL3.0, while much better than MODELLER and SCAP. This indicates that our method also works well on nonnative backbones.

**Table 4.** $\chi_1$ accuracy of our method, MODELLER, TreePack, SCWRL3.0, and SCAP on the 24 nonnative test proteins.

|  | MODELLER | TreePack | SCWRL3.0 | SCAP | our method |
|---|---|---|---|---|---|
| $\chi_1$ Accuracy | 0.428 | 0.520 | 0.530 | 0.488 | 0.520 |

### 4.6 Computational Efficiency

Since our method is based on only local backbone features, it can be expected that our method is much more computationally efficient. TreePack has been reported as one of the fastest methods for side chain prediction. Table 5 shows the total CPU time comparison of TreePack, SCWRL3.0, and our method on the 78 benchmark set. All three programs are tested on a Debian Linux box with a 1.7GHz CPU.

**Table 5.** CPU time comparison of TreePack, SCWRL3.0, and our method on the 78 protein benchmark set.

| | TreePack | SCWRL3.0 | Our Method |
|---|---|---|---|
| Time | 3min6sec | 10min57sec | 46sec |

From Table 5, it is clear that our method is much faster than both TreePack and SCWRL3.0. In fact, we are more than 14 times faster than SCWRL3.0, and more than 4 times faster than TreePack. The average CPU time of our method on one test protein is 0.58 seconds. We also tested the CPU time of our method on the original 180 benchmark set, the results are consistent with the 78 benchmark set.

## 5 Discussions

This paper formulated protein side chain packing as a machine learning problem and developed a multi-class SVM method for protein side chain prediction. As far as we know, this is the first attempt to apply multi-class SVM method for the side chain prediction problem. Our experimental results demonstrate that this new method works very well.

This paper demonstrated that protein side chain positions can be predicted using local backbone information to the same accuracy as those programs employing pairwise energy functions and computationally-intensive optimization algorithms, such as SCWRL and TreePack. We hope our discovery will change the way researchers look at this problem and lead to rapid and accurate protein side chain packing programs, which are indispensable in high-accuracy protein structure modeling.

One of the major bottlenecks in protein structure refinement is how to quickly generate a huge number of possible full-atom conformations so that a full-atom energy function can be used to pick up those with favorable energy. Our method enables us to generate a good side chain packing extremely fast after a change of backbone conformation. Since our method depends on local backbone information only, our method can be made even much more faster when only a local part of a protein structure is refined. This allows us to do side chain packing at each step of protein structure refinement and thus makes it feasible to apply an accurate full-atom energy function to each generated conformation.

We plan to further examine the features used in our method to see if more improvement can be achieved. For example, we only used a feature “contact number” to describe how many residues are in contact with a given residue. This feature does not capture the types of amino acids that are in contact with this given residue. We can extend this single “contact number” to a vector of twenty contact numbers, each of which is the number of residues, of the same amino acid type, in contact with this given residue. We only used three types of secondary structure in our model. This may be enriched by eight types of secondary structure.

## Acknowledgements

We are grateful to Dongbo Bu and Bo Jiang for the thought provoking discussion and comments. We also want to thank Shuai Cheng Li for helping us prepare the PDB20 data set.

## References

1. http://ca.expasy.org/sprot/relnotes/relstat.html.

2. http://predictioncenter.org/casp6/Casp6.html.
3. http://svmlight.joachims.org/svm_multiclass.html.
4. Hubbard,s.j. and thornton, j.m. (1993), 'naccess', computer program, department of biochemistry and molecular biology, university college london.
5. T. Akutsu. NP-hardness results for protein side-chain packing. *Genome Informatics 8*, pages 180–186, 1997.
6. E. Benedetti, G. Morelli, G. Nemethy, and H. Scheraga. Statistical and energetic analysis of sidechain conformations in oligopeptides. *Int. J. Peptide Protein Res.*, 22:1–15, 1983.
7. T.N. Bhat, V. Sasisekharan, and M. Vijayan. An analysis of side-chain conformation in proteins. *Int. J. Pept. Protein Res.*, 14:170–184, 1979.
8. A. Canutescu, A. Shelenkov, and R. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science*, 12:2001–2014, 2003.
9. R. Chandrasekaran and G. Ramachandran. Studies on the conformation of amino acids. xi. analysis of the observed side group conformations in proteins. *Int. J. Protein Research*, 2:223–233, 1994.
10. B. Chazelle, C. Kingsford, and M. Singh. A semidefinite programming approach to side chain positioning with new rounding strategies. *Informs Journal on Computing*, 16:380–392, 2004.
11. K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.
12. B. Dahiyat and S. Mayo. Protein design automation. *Protein Science*, 5:895–903, 1996.
13. B.V. Dasarathy. *Nearest neighbor (NN) norms: NN pattern classification techniques.* Los Alamitos: IEEE Computer Society Press, 1990, 1990.
14. J. Desjarlais and T. Handel. De novo design of the hydrophobic cores of proteins. *Protein Science*, 4:2006–2018, 1995.
15. J. Desmet, M. Maeyer, B. Hazes, and I. Laster. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992.
16. R. Dunbrack. Rotamer libraries in the 21st century. *Curr. Opin. Struct. Biol.*, 12:431–440, 2002.
17. R. Dunbrack and F. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6:1661–1681, 1997.
18. R. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins: Application to side-chain prediction. *J. Mol. Biol.*, 230:543–574, 1993.
19. R. Dunbrack and M. Karplus. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Struct. Biol.*, 1:334–340, 1994.
20. Rrendan J. Mcconkey Marvin Enelman Vlanimir Sobolev Eran Eyal, Rafeal Najmanovich. Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins. *J. Comput. Chem.*, 25:712–724, 2004.
21. O. Eriksson, Y. Zhou, and A. Elofsson. Side chain-positioning as an integer programming problem. In *WABI 2001*, pages 128–141, 2001.
22. J. Pothier G. Labesse, N. Colloc'h and J.P. Mornon. P-SEA, a new efficient assignment of secondary structure from $C_\alpha$ trace of proteins. *CABIOS*, 13:291–295, 1997.
23. L Holm and C. Sander. Fast and simple monte carlo algorithm for side chain optimization in proteins: Application to model building by homology. *Proteins: Structure, Function and Genetics*, 14:213–223, 1992.
24. C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2003.
25. J. Hwang and W. Liao. Side-chain prediction by neural networks and simulated annealing optimization. *Protein Eng.*, 8:363–370, 1995.
26. T. Jain, D. Cerutti, and J. McCammon. Configurational-bias sampling techinique for predicting side-chain conformations in proteins. *Protein Science*, 15:2029–2039, 2007.
27. J. Janin, S. Wodak, M. Levitt, and B. Maigret. The conformation of amino acid side chains in proteins. *J. Mol. Biol.*, 125:357–386, 1978.
28. C. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21:1028–1036, 2005.
29. H. Kono and J. Doi. A new method for side-chain conformation prediction using a hopfield network and reproduced rotamers. *J. Comp. Chem.*, 17:1667–1683, 1996.
30. C. Lee and S. Subbiah. Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.*, 217:373–388, 1991.
31. S. Liang and N. Grishin. Side-chain modeling with an optimized scoring function. *Protein Science*, 11:322–331, 2002.
32. M. Maeyer, J. Desmet, and I. Lasters. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Fold Des.*, 2:53–66, 1997.
33. M. McGregor, S. Islam, and M. Sternberg. Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *J. Mol. Biol.*, 198:295–310, 1987.

34. J. Mendes, H. Nagarajaram, C. Soares, T. Blundell, and M. Carrondo. Incorporating knowledge-based biases into an energy-based side-chain modeling method: Application to comparative modeling of protein structure. *Biopolymers*, 59:72–86, 2001.
35. R. Peterson, P. Dutton, and A. Wand. Improved side-chain prediction accuracy using an *ab initio* potential energy function and a very large rotamer library. *Protein Science*, 13:735–751, 2004.
36. N. Pierce and E. Winfree. Protein design is NP-hard. *Protein Eng.*, 15:779–782, 2002.
37. J. Ponder and F. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193:775–791, 1987.
38. C. Rohl, C. Strauss, D. Chivian, and D. Baker. Modeling structurally variable regions in homologous proteins with rosetta. *Proteins: Structure, Function, and Bioinformatics*, 55:656–677, 2004.
39. A. Roitberg and R. Elber. Modeling side chains in peptides and proteins: Application of the locally enhanced sampling and the simulated annealing methods to find minimum energy functions. *Chem. Phys.*, 95:9277–9287, 1991.
40. A. Sali and T.L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234:779–815, 1993.
41. H. Schrauber, F. Eisenhaber, and P. Argos. Rotamers: To be or not to be? an analysis of amino acid sidechain conformations in globular proteins. *J. Mol. Biol.*, 230:592–612, 1993.
42. G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.
43. A. Street and S. Mayo. Intrinsic beta-sheet propensities result from van der waals interactions between side chains and the local backbone. *PNAS*, 96:9074–9076, 1999.
44. N.L. Summers and M. Karplus. Construction of side-chains in homology modeling: Application to the c-terminal lobe of rhizopuspepsin. *J. Mol. Biol.*, 210:785–810, 1989.
45. B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. *NIPS 16*, 2004.
46. I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *The* $21^{st}$ *International Conference on Machine Learning*, volume 69, pages 104–111, 2004.
47. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453 C 1484, 2005.
48. P. Tuffery, C. Etchebest, S. Hazout, and R. Lavery. A new approach to the rapid determination of protein side chain conformations. *J. Biomol. Struct. Dyn.*, 8:1267–1289, 1991.
49. M. Vasquez. An evaluation of discrete and continuum search techniques for conformational analysis of side-chains in proteins. *Biopolymers*, 36:53–70, 1995.
50. Z. Xiang and B. Honig. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.*, 311:421–430, 2001.
51. J. Xu. Rapid protein side-chain packing via tree decomposition. In *RECOMB2005*, pages 423–439, 2005.
52. J. Xu and B. Berger. Fast and accurate algorithms for protein side-chain packing. *Journal of ACM*, 53:533–557, 2006.
53. J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1:95–117, 2003.
54. C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. In *RECOMB2007*, pages 381–395, 2007.