Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 10-015

Evaluation of Protein Backbone Alphabets: Using Predicted Local Structure for Fold Recognition

Kyong Jin Shim

July 14, 2010

Evaluation of Protein Backbone Alphabets: Using Predicted Local Structure for Fold Recognition

Kyong Jin Shim*
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
kjshim@cs.umn.edu

ABSTRACT

Optimally combining available information is one of the key challenges in knowledge-driven prediction techniques. In this study, we evaluate six Phi and Psi-based backbone alphabets. We show that the addition of predicted backbone conformations to SVM classifiers can improve fold recognition. Our experimental results show that the inclusion of predicted backbone conformations in our feature representation leads to higher overall accuracy compared to when using amino acid residues alone.

1. INTRODUCTION

Local protein structures describe an individual amino acid residue's environment as well as its relationship to its neighboring amino acid residues in a three-dimensional space [2]. There are many different types of local structures such as Phi, Psi, and Omega angles, hydrogen bonds between amino acid residues, lengths of bonds between atoms, number of water molecules on the surface of an amino acid residue, and number of neighboring amino acid residues and their locations in certain proximity.

We can encode these different properties into discrete categories by grouping amino acid residues with similar properties into the same category. Such discrete encoding is called a local structure alphabet [2]. Local structure alphabets are designed to not allow an overlap between categories so that any given amino acid residue can be assigned to a single category unambiguously. Such discretization allows for complex information in a three-dimensional space to be represented in a one-dimensional space. A backbone alphabet is one example of a local structure alphabet. Each letter in a backbone alphabet represents a backbone conformation which defines a set of Phi (Φ) and Psi (Ψ) angle ranges.

Many earlier studies focused on a three-state (alpha helix, beta sheet, and coil) classification of secondary structure. However, the three-state classification is known to provide little information about coils, which accounts for 45% of protein structure [34]. Over the years, many studies reported fine-grained local structure alphabets [3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17]. A recent study [2] reported a systematic analysis of various local structure alphabets. The study broadly evaluated seven different local structure alphabets and they were DSSP [3], STRIDE [10], Protein Blocks [17], HMMSTR [16], STR [2], ALPHA [2], and TCO [2].

In this study, we evaluate six Phi and Psi-based backbone alphabets; Kang [18], HMMSTR [16], Topham [19], Sun and Jiang [20], Oliva [21], and Zimmerman [22]. Each backbone alphabet consists of a certain number of letters or backbone conformations. Each conformation represents a set of Phi and Psi angle ranges. Section 3 discusses each of these backbone alphabets in details.

Our evaluation protocol tests whether a backbone alphabet is predictable from amino acid residues. Then, it tests whether the backbone alphabets are useful for fold recognition. We compute the baseline performance of SVM classifiers that do not use backbone information. Then, we compute the performance of SVM classifiers that do use backbone information. We then perform comparative analyses between the two. We use Support Vector Machines [23; 24] and different feature representation schemes.

2. BACKGROUND

An amino acid is a molecule that contains an amino group (NH2) and a carboxyl group (COOH). Attached to its alpha carbon (C_α) are a hydrogen atom and a side chain. Two or more amino acids can be linked by a bond called peptide bond, which involves amino and carboxyl groups, and a linear succession of peptide bonds is called a peptide. When many amino acids form such a linear succession, it is called a polypeptide. All participating amino acids in a linear succession of peptide bonds are referred to as amino acid residues. The peptide backbone consists of all the atoms that are involved in the peptide bond.

A polypeptide chain has rotational freedom about two bonds formed by the alpha carbon, and they are the Phi (Φ) and Psi (Ψ) angles as shown in Figure 1. The Φ angle forms between C_α and N and the Ψ angle forms between C_α and C. Due to steric interference between the peptide backbone and the side chains, a given polypeptide chain can take a limited number of possible conformations.

In the late 1960's, G. N. Ramachandran used computer models of polypeptides to systematically vary the Φ and Ψ angles with the objective of finding stable conformations [26]. For each conformation, the structure was examined for close contacts between atoms. Atoms were treated as hard spheres with dimensions corresponding to their van der Waals radii. Therefore, the Φ and Ψ angles which cause spheres to col-

*Corresponding author. Work submitted as a Plan B project in June 2006.

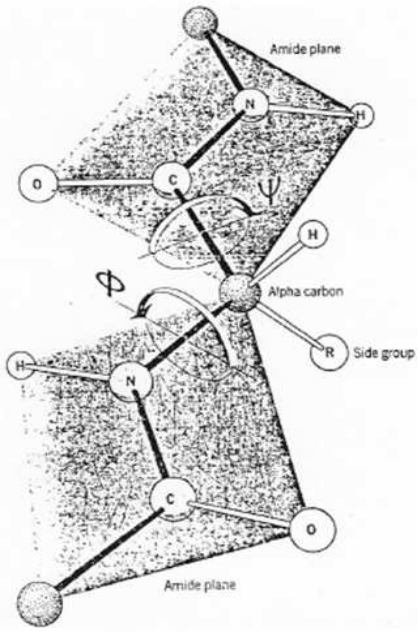

Figure 1: Rotation of the peptide backbone about C_α atom is only possible about the Φ and Ψ angles (Copyright Irving Geis).

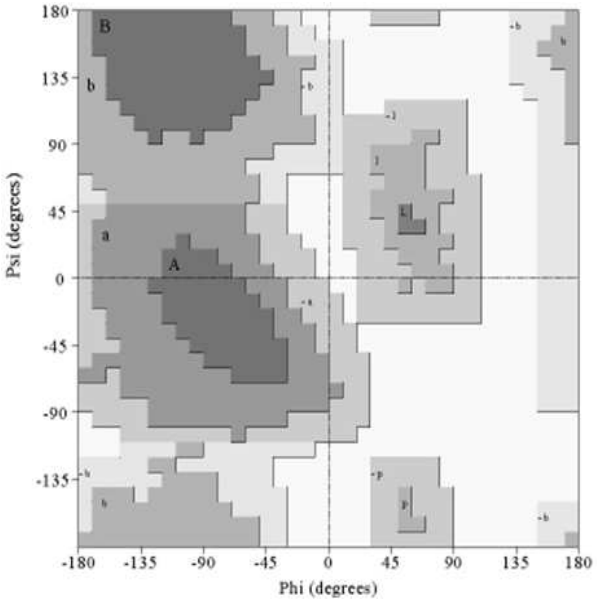

Figure 2: A Ramachandran plot (a plot of Φ angle versus Ψ angle). Dark areas are low energy or favored regions for particular combinations of torsion angles.

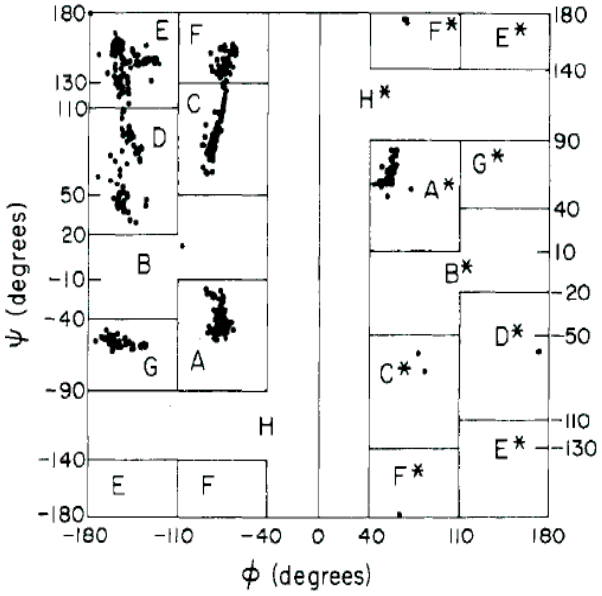

Figure 3: Zimmerman's backbone alphabet

lide correspond to sterically disallowed conformations of the polypeptide backbone. A Ramachandran plot is shown in Figure 2.

3. PHI AND PSI-BASED BACKBONE ALPHABETS

3.1 Zimmerman

In their 1977 study, Zimmerman et. al. attempted to locate all low-energy minima of amino acid residues in polypeptides. They split the Phi and Psi space into 1296 10-degree by 10-degree bins, and plotted the conformational energy of a given amino acid residue. After the plotting, they divided the Phi and Psi space into 16 regions in the Ramachandran plot and defined the letter code so that all related minima fall within the same region. The 16 regions are E, F, D, C, B, G, A, H, E*, F*, D*, C*, B*, G*, A*, and H*, and they are shown in Figure 3.

3.2 Kang

In their 1992 work, Kang et. al. attempted to set up probability tables for the Phi and Psi angles (Figure 4). Their objective was to improve the process of identifying possible conformations for a given protein by searching through a large number of available conformations. Rather than searching through a vast number of available conformations randomly, by using probability tables, a set of biases can be introduced when searching through conformations. In their experiment, they used 8,600 amino acid residues from 55 high-resolution protein structures. They divided the Phi and Psi space into 1,296 10-degree by 10-degree bins. When they calculate the probability of a given amino acid residue belonging to a particular bin, the residue's local neighbors and their Phi and Psi angles are considered in the calculation. Then, they averaged the Phi and Psi angle probabilities over all residue types, and split the Phi and Psi space

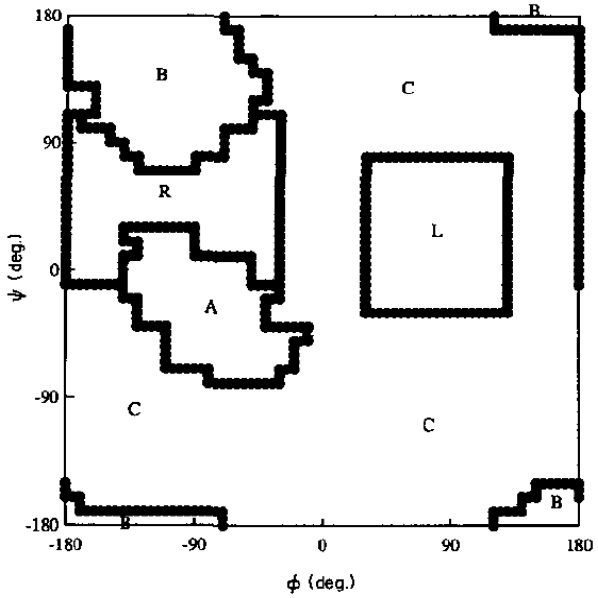

Figure 4: Kang's backbone alphabet

into five regions and they are A, B, C, L, and R.

3.3 Topham

In their 1993 study, Topham et. al. sought to construct substitution tables for amino acid residues that are conformationally constrained. They split the Phi and Psi space into 1,296 10-degree by 10-degree bins. Then, they took 83 high-resolution protein structures, calculated the Phi and Psi angles of their amino acid residues, and allocated them to corresponding bins in the Ramachandran plot. Later, they largely split the Phi and Psi space into seven regions in the Ramachandran plot as shown in Figure 5. The regions are a, b, e, g, l, p, and t. The plot indicates the number of observed amino acid residues at 10-degree intervals.

3.4 Sun and Jiang

In their 1996 study, Sun and Jiang set out to analyze supersecondary structures in 240 high-resolution proteins (Figure 6). They first followed Kabsch and Sander's method [3] of classifying an amino acid residue's conformation as helix, strand, or coil. However, they noticed that this three-state conformation was sensitive to even small changes of direction of amide and carbonyl groups. Then, they classified amino acid residues into Topham's seven conformational regions and simplified the classification by merging p with b and merging g with l, and they are labeled b' and l', respectively. They took 38,368 amino acid residues from 240 proteins for the analysis. The plot indicates the number of observed amino acid residues at 10-degree intervals.

3.5 Oliva

In their 1997 study, Oliva et. al. set out to automate protein loop classification. Extending an earlier work by Wilmot and Thornton [25], in which they split the Phi and Psi space into 10-degree by 10-degree bins, took beta-turn amino acid residues, and plotted them in the Ramachandran plot, Oliva et. al. split the Phi and Psi space into a 9-by-9 matrix and

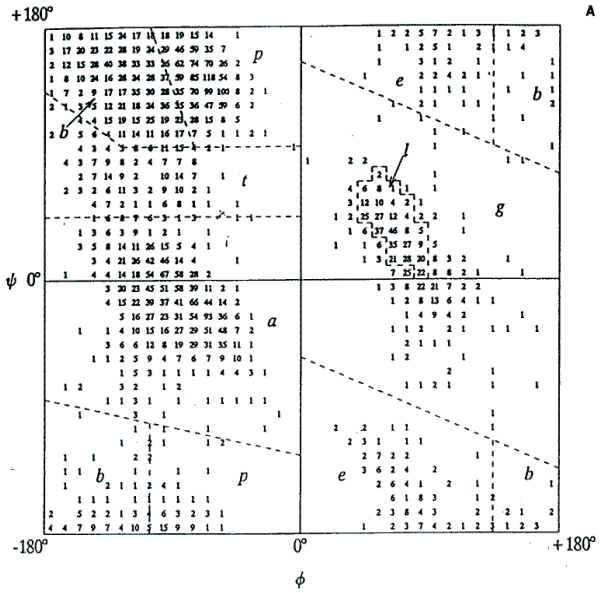

Figure 5: Topham's backbone alphabet

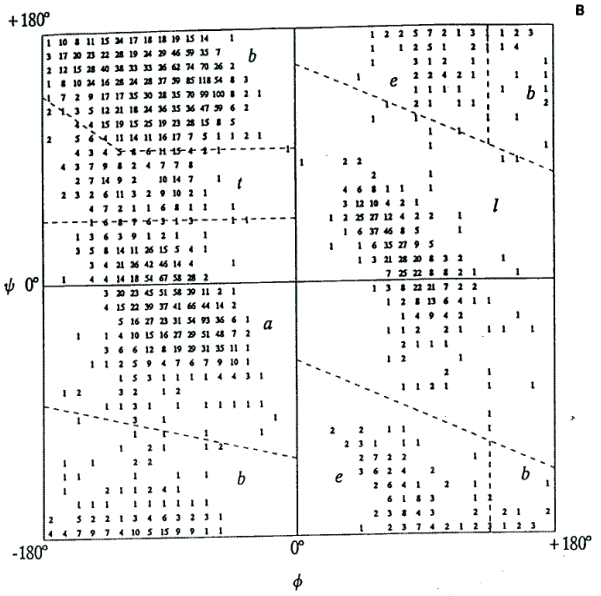

Figure 6: Sun and Jiang's backbone alphabet

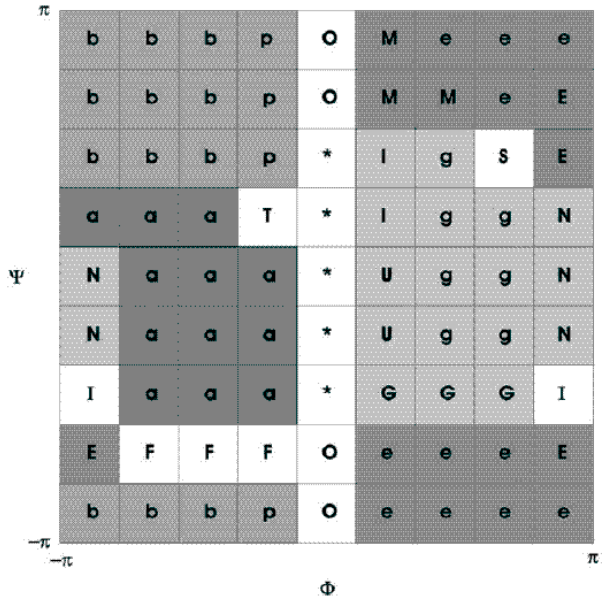

Figure 7: Oliva's backbone alphabet

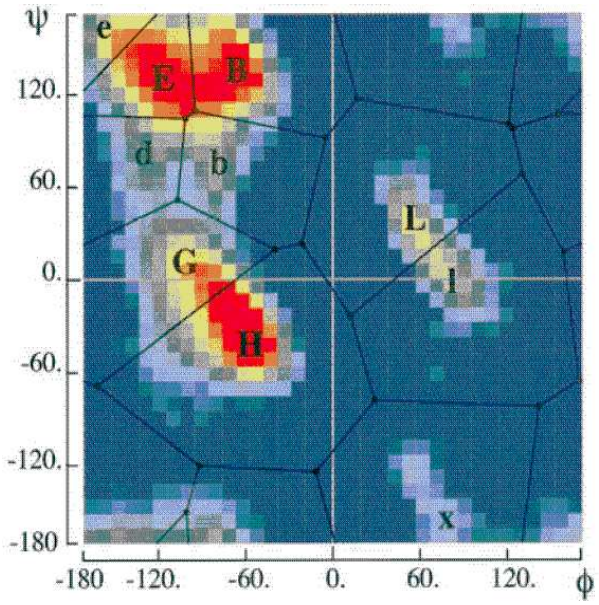

Figure 8: HMMSTR backbone alphabet

partitioned into seventeen regions. As shown in Figure 3.5, the regions are b, p, a, T, N, I, E, F, b, O, *, M, e, l, g, S, and G. In their analysis, they used 233 non-homologous and well-defined proteins.

3.6 HMMSTR

In their 2000 study, Bystroff et. al. attempted to build HMMSTR system. They partitioned the Phi and Psi space into eleven regions and they are e, E, B, d, b, G, H, L, l, x, and c. In their analysis, they used PDBselect: December 1998 [27], which is a non-redundant database of proteins whose structures are known. They took all trans amino acid residues from the proteins and plotted their Phi and Psi angles in the Ramachandran plot. Then, they used k-means clustering with k = 10 to partition the Phi and Psi space, and Voronoi method was used to draw boundaries between different clusters. As a result, they came up with ten regions for trans amino acid residues and one region for cis amino acid residues (region c) as shown in Figure 8.

4. BACKBONE PREDICTION

In backbone prediction, we want to predict the backbone conformation of each amino acid residue in a given protein sequence. We investigate the six Phi and Psi-based backbone alphabets. Each backbone alphabet consists of one or more letters, and each letter represents a certain backbone conformation.

4.1 Dataset

Of the 3,314 PDB chains in the latest Dunbrack-culled-PDB dataset (with sequence identity cut-off of 20%, resolution cut-off of 3.0 Angstrom, and R-factor cut-off of 1.0), 1,709 of them map to 2,109 chains in Astral SCOP version 1.69 [35], which contains 67,210 chains [28]. Also, those proteins whose sequence length is less than 20 amino acid residues are removed. The reason that the mapping is not one-to-one

Dataset	Number of SCOP chains	Number of Nine-mers*
1	516	93,326
2	579	103,304
3	479	86,824

Table 1: *Feature representation in backbone prediction

is because a single PDB chain can map to more than one SCOP chain.

Next, we take the 2,109 SCOP chains and apply certain filtering criteria [2]. The SCOP class e (multi-domain) is removed because folds of this class contain domains belonging to different classes. The SCOP class i (low-resolution) is removed because folds of this class have poor quality structures. The SCOP class j (peptide) is removed because folds of this class are short protein fragments. The SCOP folds a.137 (non-globular, all-alpha subunits of globular proteins) and d.184 (non-globular, alpha-beta subunits of globular proteins) are removed because of similar reasons of ambiguous fold membership. The SCOP superfamily f.2.1 (membrane all-alpha) is removed because it is a temporary classification of transmembrane alpha-helix [35].

Later in the analysis, we use PSI-BLAST [29] log-odds score profiles of SCOP chains for feature representation. Hence, we apply one more filtering criterion by removing those SCOP chains whose PSI-BLAST log-odds profile consist of all zeros. A profile consisting of all zeros implies that there are no matches found in the target sequence database for a given SCOP chain. After this final filtering, the dataset contains 1,574 SCOP chains.

For backbone prediction and fold recognition experiments, we set out to create three distinct datasets. Specifically, for fold recognition purposes, we filter out those SCOP chains whose frequency is less than three. By doing so, we intend to make sure that at least one SCOP chain of a particular fold exists in all three datasets.

```
=== Kang's Backbone Alphabet: Round 1 ===
Total number of positive examples for each backbone alphabet letter
        B = 43109
        R = 2490
        A = 51865
        L = 4633
        C = 4504

        B       R       A       L       C
B       30592   30      12331   110     46
R       1258    3       1222    5       2
A       12039   26      39665   93      42
L       1689    60      2423    361     100
C       2556    64      1670    141     73
```

Figure 9: Confusion Matrix

Dataset 1 and Dataset 2 are used in backbone prediction experiments. Because we perform a two-fold cross validation, each of the two datasets becomes the training dataset or the testing dataset in each of the two rounds. In the first round, Dataset 1 is the training dataset and Dataset 2 is the testing dataset. In the second round, Dataset 2 is the training dataset and Dataset 1 is the testing dataset.

4.2 Classification and Evaluation

We use Support Vector Machines to predict the backbone conformation of the central amino acid residue in a nine-residue sequence fragment. We call this nine-residue sequence segment a nine-mer. We use the latest SVM Light package [30]. We perform a one-versus-all classification to make multi-class predictions from binary classifiers. For instance, if we are performing backbone prediction using Kang's backbone alphabet, we build a total of five binary classifiers, one for each backbone conformation (A, B, C, L, and R). Then, we assign the label (in this case, a backbone conformation) of the class that produces the maximum margin for each test example (i.e. a nine-mer). The label is then the predicted backbone conformation of the central amino acid residue in a nine-mer.

We perform a two-fold cross validation. At the end of the classification, each nine-mer will have for the central amino acid residue the predicted conformation. Then, we construct a confusion matrix where rows represent true labels and columns represent predicted labels as shown in Figure 9.

Figure 9 shows a confusion matrix constructed from Round 1 of the two-fold cross validation performed on Kang's backbone alphabet. The numbers highlighted in red are True Positives or correctly predicted backbone conformations. The figure shows that, for example, 30,592 out of 43,109 nine-mers whose central amino acid residue had "B" as their true backbone conformation were correctly predicted. 12,331 nine-mers whose central amino acid residue had "B" as their true backbone conformation were incorrectly predicted to have "A" as their backbone conformation. Once we have a confusion matrix constructed, we compute the accuracy of our backbone prediction by taking the summation of diagonal values and dividing it by the total number of nine-mers in the dataset. The accuracy indicates how many true positives and true negatives out of all the nine-mers in the dataset that our method correctly predicts.

4.3 Feature Representation

We use the PSI-BLAST log-odds profiles of a nine-mer for feature representation. Around the central amino acid residue, there are four locally neighboring amino acid residues.

```
A B C D E F G H I

Central residue:  E

Local neighbors:  A, B, C, D, F, G, H, I
```

Figure 10: Amino Acid Residues

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	C	-1	-4	-3	-4	10	-4	-4	-3	-4	-2	-2	-4	-2	-3	-3	-1	-1	-3	-3	-1
2	P	-2	-4	-3	-3	-4	-3	-2	-4	-4	-4	-4	-2	-4	-5	8	-2	-2	-5	-4	-4
3	V	-2	-4	-5	-5	-3	-4	-5	-5	-5	3	1	-4	-1	-2	-4	-4	-2	-5	-3	6
4	S	-2	-3	5	-2	5	1	1	0	-2	-4	-4	-2	0	-5	-4	2	0	1	-4	-4
5	S	1	-3	-2	-3	2	-3	-3	-3	-3	-1	-2	-3	-2	-3	-3	4	-1	-4	-3	4
6	Y	-3	-3	-3	-4	-4	-3	-1	-4	5	-4	-3	0	-3	2	-4	-1	1	8	6	-4
7	N	-1	-3	7	-1	-4	-2	-2	-3	1	-5	-5	-2	-4	-5	-4	0	2	-6	-4	-4
8	E	-3	-2	-2	2	-6	0	7	-4	-2	-6	-5	-1	-4	-6	-3	-2	-3	-5	-4	-5
9	W	-4	-4	-5	-6	-4	-4	-5	-5	1	-4	-3	-5	-3	3	-5	-4	1	11	5	-4
10	D	1	-4	-1	6	-5	-3	0	2	-3	-5	-5	-3	-5	-5	-4	1	-3	-6	-5	-5
11	P	-3	0	-4	-3	-5	-3	0	-4	-4	-3	-2	-1	-4	-5	7	0	0	-5	-5	0
12	L	-4	-4	-6	-6	-3	-4	-5	-6	-5	0	6	-5	0	-2	-5	-5	-3	-4	-3	-1
13	E	-3	4	-2	-2	-5	1	4	-4	1	-5	-4	4	-4	-5	-3	-2	-1	-5	-4	-4
14	E	0	-1	-2	1	-4	-1	5	-4	3	-4	-4	2	-4	-5	-3	-1	1	-5	-3	-4
15	V	-2	-4	-5	-5	-3	-4	-4	-5	-5	3	0	-4	1	-3	-4	-4	-2	-5	-3	6
16	I	-3	-5	-5	-5	-3	-4	-5	-5	-5	6	0	-4	-1	-2	-4	-4	-3	-5	-3	4
17	V	-2	-4	-4	-3	-3	-4	-4	-5	-4	1	3	-4	-1	-2	-4	-4	-2	-4	-3	6
18	G	1	-3	-2	-3	4	-3	-3	6	-3	-4	-4	-3	-3	-4	-3	-2	-3	-3	-4	-3
19	R	-2	5	-2	-3	-1	0	-1	-3	-3	1	-1	0	-2	-3	-3	1	2	-4	-3	-1
20	A	4	-3	-3	-3	1	-3	-3	-3	-4	-2	-1	-3	1	-4	5	0	-2	-4	-4	0

Figure 11: PSI-BLAST log-odds profile

Figure 11 shows a segment of the PSI-BLAST log-odds profile of Astral SCOP 1.69 chain d2jdxa_, and it shows the log-odds scores of the first 20 amino acid residues out of 359. Starting with the first amino acid residue until the last one in a given protein sequence, each amino acid residue becomes the central amino acid residue in a nine-mer. In the profile as shown in Figure 11, each amino acid residue has a string of 20 integers. For a given nine-mer, we take one such string of 20 integers at a time and concatenate one after another from left to right. Hence, the feature representation of a given nine-mer is a concatenation of nine 20-dimension vectors.

At the beginning and the end of a given protein sequence where a nine-mer does not consist entirely of amino acid residues but one or more blank positions, we simply fill 20 zeros in each such blank position.

4.3.1 *Neighboring Scheme 1*

In this neighboring scheme, we use nine consecutive amino acid residues and their PSI-BLAST profiles for feature representation. Around the central amino residue, there are four locally neighboring amino acid residues.

4.3.2 *Neighboring Scheme 2*

Our preliminary experiments using Neighboring Scheme 1

```
_ _ _ C P V S S S Y
```

Figure 12: Nine-mer

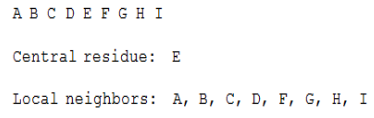

Figure 13: Neighboring Scheme 1

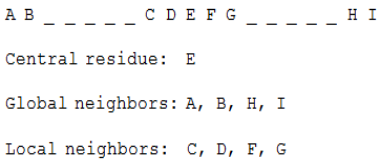

Figure 14: Neighboring Scheme 2

indicate that our classifiers tend to over-predict alpha and beta regions and under-predict coil regions. Throughout our experiments, when we refer to alpha, beta, and coil regions, we follow the secondary structure definitions of the DSSP secondary structure assignment program [3]. Taking into account the fact that coil regions rely on local as well as global interactions [31], we devise a different neighboring scheme specifically for coil regions.

We use the DSSP secondary structure assignments to divide our study into two cases: coil versus non-coil (alpha and beta). To simplify our experiments, instead of using the original eight-state secondary structures, we combine H, G, I into H (alpha), E and B into B (beta), and all others to C (coil).

For non-coil regions (alpha and beta), we use Neighboring Scheme 1 for feature representation. For coil regions, we use Neighboring Scheme 2. The scheme considers four globally neighboring amino acid residues and four locally neighboring amino acid residues around the central amino acid residue. The distance between the central amino acid residue and the closest global neighbor is "5 residues away".

4.3.3 Neighboring Scheme 3

This scheme is similar to Neighboring Scheme 2, but it considers different sets of neighbors. For coil regions, we use eight globally neighboring amino acid residues around the central amino acid residue. The distance between the central amino acid residue and the closest global neighbor is "5 residues away".

5. FOLD RECOGNITION

In fold recognition, we want to predict the SCOP fold of a given protein sequence. Our experiments are designed to evaluate which backbone alphabets are effective in improving fold recognition.

5.1 Dataset

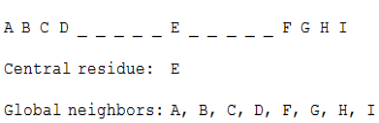

Figure 15: Neighboring Scheme 3

Dataset 3 is the testing dataset in fold recognition. Because we perform a two-fold cross validation in backbone prediction, we can have either Dataset 1 or Dataset 2 as the training dataset in fold recognition. If Dataset 1 is used as the training dataset in backbone prediction and Dataset 2 as the testing dataset, Dataset 2 is used as the training dataset in fold recognition. On the other hand, if Dataset 2 is used as the training dataset in backbone prediction and Dataset 1 as the testing dataset, Dataset 1 is used as the training dataset in fold recognition. Because we seek to incorporate predicted backbone conformations as part of feature representation in fold recognition, we perform backbone prediction of Dataset 3. Again, if Dataset 1 is used as the training dataset in backbone prediction of Dataset 2, we use Dataset 1 as the training dataset in backbone prediction of Dataset 3. On the other hand, if Dataset 2 is used as the training dataset in backbone prediction of Dataset 1, we use Dataset 2 as the training dataset in backbone prediction of Dataset 3.

5.2 Classification and Evaluation

We use Support Vector Machines to predict the SCOP fold of a given protein sequence. We use the latest SVM Light package. A total of 177 SCOP folds are used in our fold recognition experiments. Hence, we build a total of 177 binary classifiers. We compute ROC scores. The ROC score represents the area under the receiver operating characteristic curve. ROCN is a plot of true positives as a function of false positives up to the first N false positives [32]. On a scale of zero to one, a score of one means that there is perfect separation between positives and negatives. A score of zero means that among the samples selected by the method, none is positive. In cases where the ratio between true negatives and true positives is very large, a fixed ROC number is used and the area is calculated under a truncated ROC curve [32].

5.3 Feature Representation

The spectrum kernel [33] is used to generate SVM feature vectors. The spectrum kernel models a sequence in the space of all k-mers (subsequences of length k), and its features count the number of times each k-mer appears in the sequence. In our fold recognition experiments, we move a k-length sliding window across a protein sequence, look up the current k-mer subsequence in the look-up table, and increment the classifier value. For instance, using a window size of three, we can map each of the 20 amino acid residue to an 8000-dimension vector. And if we use Kang's backbone alphabet, we can map each of the five backbone conformations to a 125-dimension vector. We can use the above vectors individually as a feature representation, but we can also easily combine one or more vectors by concatenating them to give rise to a new feature representation.

For comparison purposes, our feature representation takes three different types of information and they are amino acid residues, backbone conformations, and secondary structures. Specifically, we can use each type of information individually or alternatively, we can use combinations of two or more different types of information. In our fold recognition experiments, we deal with five different combinations of available information and they are listed below.

1) Amino acid residues

2) Amino acid residues AND "predicted" backbone confor-

Backbone Alphabet (# of backbone conformations)	Accuracy (Round 1)	Accuracy (Round 2)
Zimmerman (14*)	0.417	0.429
Kang (5)	0.663	0.658
Topham (7)	0.602	0.592
Sun and Jiang (5)	0.677	0.672
Oliva (17)	0.671	0.666
HMMSTR (11)	0.429	0.438

Table 2: *Backbone conformations 'c' and 'a' are omitted because we find no nine-mers in our datasets that match these conformations.

mations
- We use predicted backbone conformations in both training and testing.
3) Amino acid residues AND "true" backbone conformations
- We use true (known) backbone conformations in both training and testing. Ideally, this would never be the case in classification where backbone conformations are unknown for testing examples. This is done solely for evaluation purposes.
4) Amino acid residues AND PSI-PRED-predicted secondary structures
5) Amino acid residues AND DSSP-assigned secondary structures
- We use true (known) secondary structures in both training and testing. Ideally, this would never be the case in classification where secondary structures are unknown for testing examples. This is done solely for evaluation purposes.

5.3.1 *Neighboring Scheme 1*

In this feature representation, we use k = 3 for amino acid residues, backbone conformations, and secondary structures.

5.3.2 *Neighboring Scheme 2*

In this feature representation we use k = 3 for amino acid residues, and we use k = 5 for backbone conformations and secondary structures. Our goal is to see if including more neighbor information (in the cases of backbone conformations and secondary structures) would improve fold recognition.

6. EXPERIMENTS AND RESULTS

6.1 Backbone Prediction

6.1.1 *Neighboring Scheme 1*

In this first set of results, we investigate our backbone prediction using Neighboring Scheme 1 for feature representation. We evaluate the six backbone alphabets for each of which we construct a confusion matrix and compute accuracy. Because we perform a two-fold cross validation, for each backbone alphabet, we construct two confusion matrices one for each of the two rounds and compute two accuracy values. Table 2 comparatively shows the computed accuracy values when we use the linear kernel with default parameters. Using the polynomial and radial kernels with default parameters does not improve prediction accuracy.
It is worth studying the generated confusion matrices because they show which backbone conformations are over-predicted or under-predicted. Figure 16 and Figure 17 show

```
=== Kang's Backbone Alphabet: Round 1 ===
Total number of positive examples for each backbone alphabet letter
        B = 43109
        R = 2490
        A = 51865
        L = 4633
        C = 4504

        B       R       A       L       C
B       30592   30      12331   110     46
R       1258    3       1222    5       2
A       12039   26      39665   93      42
L       1689    60      2423    361     100
C       2556    64      1670    141     73
```

Figure 16: Confusion Matrix - Kang's Backbone Alphabet

```
=== Topham's Backbone Alphabet: Round 1 ===
Total number of positive examples for each backbone alphabet letter
        b = 28967
        p = 15248
        t = 2595
        a = 53604
        e = 1267
        g = 2172
        l = 2748

        b       p       t       a       e       g       l
b       16616   210     48      11674   157     145     117
p       5412    627     59      8635    230     164     121
t       948     46      6       1520    27      27      21
a       6602    314     40      46225   190     147     86
e       415     44      17      578     88      76      49
g       507     62      22      1293    105     103     80
l       480     30      16      1570    65      66      521
```

Figure 17: Confusion Matrix - Topham's Backbone Alphabet

two confusion matrices, one showing the second round of Kang and the other showing the second round of Topham.
In Figure 5.1, the numbers highlighted in blue represent the nine-mers that are predicted to have for its central amino acid residue the backbone conformation "B". The numbers highlighted in green represent the nine-mers that are predicted to have for its central amino acid residue the backbone conformation "A". In Dataset 2, over 89% of all the nine-mers in the dataset have these two backbone conformations for their central amino acid residues. The number of examples corresponding to the other three backbone conformations is relatively minimal. As a result, our classifiers tend to over-predict backbone conformations "B" and "A" and under-predict backbone conformations "R", "L", and "C".
Figure 5.2 shows the same trend in the results obtained from our experiment using Topham's backbone alphabet. In Dataset 2, over 77% of all the nine-mers in the dataset have backbone conformations "b" and "a" for their central amino acid residues. As in the case of Kang's backbone alphabet, the number of examples in the dataset that correspond to "b" and "a" is relatively large compared to those corresponding to "p", "t", "e", "g", and "l". As a result, our classifiers tend to over-predict largely-populated backbone conformations and under-predict the rest. All the confusion matrices generated in this experiment are available in Appendix A, and they show the same trend of over-predicting certain regions and under-predicting the rest.
An interesting observation is that the regions in the Ra-

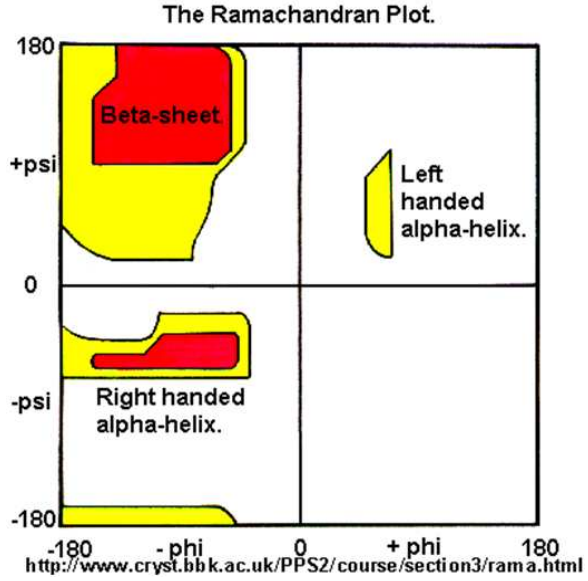

Figure 18: Ramachandran Plot

machandran plot that are over-predicted by our classifiers roughly match the regions that have been traditionally known as alpha and beta regions in the original Ramachandran plot.

Hence, it is observed that our classifiers tend to over-predict alpha and beta regions and under-predict coil regions. Taking into account the fact that coil regions rely on local as well as global interactions (Skolnick), we devise a different neighboring scheme specifically for coil regions. Our goal in this effort is to better represent coil regions by incorporating local and/or global neighbors in our nine-mer feature representation.

6.1.2 Neighboring Scheme 2

In this set of results, we investigate our backbone prediction using Neighboring Scheme 2 for feature representation. For non-coil regions (alpha and beta), we use Neighboring Scheme 1 for feature representation. For coil regions, we use Neighboring Scheme 2. The scheme considers four globally neighboring amino acid residues and four locally neighboring amino acid residues around the central amino acid residue. In this experiment, we only tested Kang's and HMMSTR backbone alphabets. More experiments that test all six backbone alphabets are planned in our future studies.

Using Neighboring Scheme 2 for coil regions and Neighboring Scheme 1 for alpha and beta regions has been shown to have no significant effects on improving backbone prediction. For instance, in the case of Kang's backbone alphabet, the prediction accuracy values in both rounds of the two-fold cross validation are 0.004 and 0.002 lower than when using Neighboring Scheme 1 for all nine-mers. In the case of HMMSTR, the prediction accuracy values in both rounds are 0.003 and 0.005 higher than when using Neighboring Scheme 1 for all nine-mers. But these improvements in prediction accuracy values are very minimal.

6.1.3 Neighboring Scheme 3

This scheme is similar to Neighboring Scheme 2, but it considers different sets of neighbors. For coil regions, we use eight globally neighboring amino acid residues around the central amino acid residue. In this experiment, we only tested Kang's and HMMSTR backbone alphabets. More experiments that test all six backbone alphabets are planned in our future studies.

Using Neighboring Scheme 3 for coil regions and Neighboring Scheme 1 for alpha and beta regions has been shown to have no significant effects but comparatively positive effects on improving backbone prediction in comparison to Neighboring Scheme 2.

For instance, in the case of Kang's backbone alphabet, the prediction accuracy values in both rounds of the two-fold cross validation are 0.007 and 0.009 higher than when using Neighboring Scheme 1 for all nine-mers. In the case of HMMSTR, the prediction accuracy values in both rounds are 0.035 and 0.024 higher than when using Neighboring Scheme 1 for all nine-mers. These improvements in prediction accuracy values achieved by Neighboring Scheme 3 are higher than that achieved by Neighboring Scheme 2.

6.2 Fold Recognition

6.2.1 Feature Representation 1

In this first set of results, we investigate our fold recognition using Feature Representation 1. Our feature representation takes three different types of information and they are amino acid residues, backbone conformations, and secondary structures. We test the following combinations of feature vectors for our feature representation.

1) Amino acid residues

2) Amino acid residues AND "predicted" backbone conformations

- We use predicted backbone conformations in both training and testing.

3) Amino acid residues AND "true" backbone conformations

- We use true (known) backbone conformations in both training and testing. Ideally, this would never be the case in classification where backbone conformations are unknown for testing examples. This is done solely for evaluation purposes.

4) Amino acid residues AND PSI-PRED-predicted secondary structures

5) Amino acid residues AND DSSP-assigned secondary structures

- We use true (known) secondary structures in both training and testing. Ideally, this would never be the case in classification where secondary structures are unknown for testing examples. This is done solely for evaluation purposes.

The fold recognition ROC24, ROC48, ROC240 and ROC 479 scores are shown in Figure 19. To compute these ROC scores, we estimate the area under the ROC curve with a trapezoidal method.

The differences between ROC scores across all six backbone alphabets are minimal. The combination AA + Predicted BB (Topham) has the highest accuracy by a small margin.

An important observation is that the inclusion of predicted backbone conformations in our feature representation leads to higher overall accuracy compared to when using amino acid residues alone.

Feature Combinations	ROC24	ROC48	ROC240	ROC479
AA	0.035	0.058	0.177	0.255
AA + Predicted BB (Kang)	0.053	0.092	0.246	0.302
AA + Predicted BB (HMMSTR)	0.056	0.089	0.221	0.285
AA + Predicted BB (Topham)	**0.063**	**0.098**	**0.241**	**0.300**
AA + Predicted BB (Sun and Jiang)	0.058	0.096	0.236	0.297
AA + Predicted BB (Oliva)	0.060	0.099	0.240	0.301
AA + Predicted BB (Zimmerman)	0.047	0.076	0.206	0.276
AA + True BB (Kang)	0.085	0.126	0.258	0.310
AA + True BB (HMMSTR)	0.098	0.136	0.257	0.307
AA + True BB (Topham)	0.096	0.136	0.262	0.312
AA + True BB (Sun and Jiang)	0.085	0.125	0.260	0.311
AA + True BB (Oliva)	0.083	0.125	0.259	0.311
AA + True BB (Zimmerman)	0.093	0.131	0.254	0.307
AA + PSI-PRED	0.068	0.103	0.225	0.288
AA + DSSP	0.080	0.116	0.239	0.298

Figure 19: Kang's Backbone Alphabet. ROC scores. AA denotes "amino acid residues". BB denotes "backbone conformations".

Feature Combinations	ROC24	ROC48	ROC240	ROC479
AA	0.035	0.058	0.177	0.255
AA + Predicted BB (Kang)	0.053	0.092	0.246	0.302
AA + Predicted BB (Sun and Jiang)	0.058	0.096	0.236	0.297
AA + Predicted BB (Kang)	0.057	0.091	0.244	0.300
AA + Predicted BB (Sun and Jiang)	0.059	0.095	0.236	0.296
AA + True BB (Kang)	0.085	0.126	0.258	0.310
AA + True BB (Sun and Jiang)	0.085	0.125	0.260	0.311
AA + True BB (Kang)	0.099	0.140	0.271	0.316
AA + True BB (Sun and Jiang)	0.103	0.146	0.275	0.319
AA + PSI-PRED	0.068	0.103	0.225	0.288
AA + DSSP	0.080	0.116	0.239	0.298
AA + PSI-PRED	0.073	0.109	0.231	0.292
AA + DSSP	0.094	0.131	0.247	0.303

Figure 20: ROC scores from Fold Recognition. AA denotes "amino acid residues". BB denotes "backbone conformations". Green color indicates the ROC scores when using k = 5 for backbone conformations and secondary structures.

6.2.2 *Feature Representation 2*

In this feature representation we use k = 3 for amino acid residues, and we use k = 5 for backbone conformations and secondary structures. Our goal is to see if including more neighbor information (in the cases of backbone conformations and secondary structures) would improve fold recognition.

Figure 20 shows the results of our experiments using k = 5 for backbone conformations and secondary structures. In the combinations AA + Predicted BB (Kang) and AA + Predicted BB (Sun and Jiang), including more neighbor information in our feature representation by expanding the sliding window from three to five only slightly improves ROC24 scores but decreases ROC48, ROC240, and ROC479 scores.

7. CONCLUSIONS

Optimally combining available information is one of the key challenges in knowledge-driven prediction techniques. In this study, the two types of information we use for feature representation are amino acid residues and predicted backbone conformations.

This study shows that the addition of predicted backbone conformations to SVM classifiers can improve fold recognition. Our experimental results indicate that the inclusion of predicted backbone conformations in our feature representation leads to higher overall accuracy compared to when using amino acid residues alone. It also indicates that the differences between ROC scores across all six backbone alphabets are minimal even though the combination AA + Predicted BB (Topham) has the highest accuracy by a small margin.

One possible extension to our backbone prediction is to devise better neighboring schemes. Given a k-length sequence fragment, the process of selecting which local and/or global neighbors to consider for feature representation can be done more systematically. For example, we can take a large number of high-resolution protein sequences and statistically determine which combinations of local and/or global neighbors around a central amino acid residue occur frequently. Using this frequency chart or table, we can convert this into a score matrix which can be easily incorporated into feature representation when building SVM classifiers.

One possible extension to our fold recognition is to use other types of kernels for feature representation. In our experiments, we used the spectrum kernel. Mismatch kernel and profile-based kernel are other types of kernels that can be used for this purpose.

8. ACKNOWLEDGEMENTS

The author wishes to express sincere appreciation to Dr. George Karypis, Dr. Dan Boley, and Dr. Colin Campbell at the University of Minnesota.

9. REFERENCES

[1] P. E. Bourne and H. Weissig. Structural Bioinformatics. John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[2] R. Karchin, M. Cline, Y. Mandel-Gutfreund, and K. Karplus. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins, Jun 1;51(4):504-14, 2003.

[3] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers, 22(12):2577-2637, Dec 1983.

[4] R. Unger, D. Harel, Wherland S., and J. Sussman. A 3D building blocks approach to analyzing and predicting structure of proteins. Proteins, 5:355-373, 1989.

[5] R. Unger and J.L Sussman. The importance of short structural motifs in protein structure analysis. J Comput Aided Mol Des, 7(4):457-472, 1993.

[6] M. J. Rooman, J. Rodriguez, and S. J. Wodak. Automatic definition of recurrent local structure motifs in proteins. Journal of Molecular Biology, 213:327-336, 1990.

[7] M. J. Rooman, J. Rodriguez, and S. J. Wodak. Relations between protein sequence and structure and their significance. Journal of Molecular Biology, 213:337-350, 1990.

[8] X. Zhang, J. S. Fetrow, W. A. Rennie, D. L. Waltz, and G. Berg. Automatic derivation of substructures yields

novel structural building blocks in globular proteins. Proceedings, 1st International Conference on Intelligent Systems for Molecular Biology, pages 438-446, 1993.

[9] H. S. Kang, N. A. Kurochkina, and B. Lee. Estimation and use of protein backbone angle probabilities. Journal of Molecular Biology, 229:448-460, 1993.

[10] D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. Proteins: Structure, Function, and Genetics, 23:566-579, 1995.

[11] M. B. Swindells, M. W. MacArthur, and J. M. Thornton. Intrinsic phi,psi propensities of amino acids, derived from the coil regions of known structures. Nat. Struct. Biol., 2(7):596-603, Jul 1995.

[12] M. J. Thompson and R. A. Goldstein. Predicting protein secondary structure with probabilistic schemata of evolutionarily derived information. Protein Science, 6:1963-1975, 1997.

[13] C. Bystroff and D. Baker. Prediction of local structure in proteins using a library of sequence-structure motifs. Journal of Molecular Biology, 281:565-577, 1998.

[14] A. C. Camproux, P. Tuffery, J. P. Chevrolat, J. F. Boisvieux, and S. Hazout. Hidden Markov model approach for identifying the modular framework of the protein backbone. Protein Eng., 12(12):1063-1073, Dec 1999.

[15] S. M. King and W. C. Johnson. Assigning secondary structure from protein coordinate data. Proteins: Structure, Function, and Genetics, 35:313-320, 1999.

[16] C. Bystroff, V. Thorsson, and D. Baker. HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. Journal of Molecular Biology, 301(1):173-190, Aug 2000.

[17] A. G. de Brevern, C. Etchebest, and S. Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. Proteins: Structure, Function and Genetics, 41:271-287, 2000.

[18] H. S. Kang, N. A. Kurochkina, and B. Lee. Estimation and use of protein backbone angle probabilities. Journal of Molecular Biology, 229:448-460, 1993.

[19] C. M. Topham, A. MacLeod, F. Eisenmenger, J. P. Overington, M. S. Johnson, T. L. Blundell. Fragment ranking in modeling of protein structure, J. Mol. Biol. 229, 194-220, 1993.

[20] Z. Sun and B. Jiang. Patterns and conformations of commonly occurring supersecondary structures (basic motifs) in protein data bank. Journal of Prot.Chem., 15(7):675-690, 1996.

[21] B. Oliva, P. A. Bates, E. Querol, F. X. Aviles, M. J. Sternberg. An automated classification of the structure of protein loops. J. Mol. Biol. 266, 814-830, 1997.

[22] S. S. Zimmerman, M. S. Pottle, G. Nemethy, and H. A. Scheraga. Conformational analysis of the 20 naturally occurring amino acid residues using ECEPP. Macromolecules, 10(1):1-8, 1977.

[23] M. A. Shipp, et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine, 8:68-74, 2002.

[24] M. Bribskov and N. L. Robinson. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. Computers and Chemistry, 20(1):25-33, 1996.

[25] C. M. Wilmot andJ. M. Thornton. Beta-turns and their distortions: a proposed new nomenclature. Protein Eng. 3, 479-494, 1990.

[26] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. Journal of Molecular Biology, 7:95-99, 1963.

[27] U. Hobohm and C. Sander. Enlarged representative set of protein structures. Protein Sci. 3, 522-524, 1994.

[28] R. Dunbrack. Culling the PDB by resolution and sequence identity. 2001. http://www.fccc.edu/research/labs/dunbrack/culledpdb.html.

[29] S. F. Altshul, T. L. Madden, A. A. Schaffer, J. H. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nuc. Ac. Res., 25:3389-3402, 1997.

[30] T. Joachims. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[31] M. Betancourt and J. Skolnick. Local propensities and statistical potentials of backbone dihedral angles in proteins. Journal of Molecular Biology 2004: 342: 635-649.

[32] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. Computational Chemistry, 20:2533, 1996.

[33] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: a string kernel for SVM protein classification. Pac Symp Biocomput, 564-75, 2002.

[34] J. S. Fetrow, M. J. Palumbo, and G. Berg. Patterns, structures, and amino acid frequencies in structural building blocks, a protein secondary structure classification scheme. Proteins: Structure, Function, and Genetics, 27:249-271, 1997.

[35] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology, 247(4):536-540, 1995.