Protein Solvent Accessibility Prediction Using Support Vector Machines and Sequence Conservations

Hasan Oğul[1] and Erkan Ü. Mumcuoğlu[2]

[1] Department of Computer Engineering, Başkent University, 06530, Ankara, Turkey
hogul@baskent.edu.tr
[2] Information Systems and Health Informatics, Informatics Institute, Middle East Technical University, 06531, Ankara, Turkey
mumcuoglu@ii.metu.edu.tr

Abstract. A two-stage method is developed for the single sequence prediction of protein solvent accessibility from solely its amino acid sequence. The first stage classifies each residue in a protein sequence as exposed or buried using support vector machine (SVM). The features used in the SVM are physico-chemical properties of the amino acid to be predicted as well as the information coming from its neighboring residues. The SVM-based predictions are refined using pairwise conservative patterns, called maximal unique matches (MUMs). The MUMs are identified by an efficient data structure called suffix tree. The baseline predictions, SVM-based predictions and MUM-based refinements are tested on a nonredundant protein data set and ~73% prediction accuracy is achieved for a solvent accessibility threshold that provides an evenly distribution between buried and exposed classes. The results demonstrate that the new method achieves slightly better accuracy than recent methods using single sequence prediction.

1 Introduction

The difficulties in the determination of protein structure and function have been led to an increase on the demand for computational tools for protein analysis. Since there are many parameters which play role in forming protein conformations, high-resolution analysis tools are required to identify different kind of features of proteins. One of those features is the solvent accessibility of the residues in a protein. A protein is composed from a chain of amino acid residues and the solvent accessibility of a residue is described in terms of the degree of its interaction with the water molecules. This interaction degree is inferred from the accessible surface area of the residue in the protein polypeptide chain. The residues with an interaction level of lower than a specified threshold are called as *buried* and the others are called as *exposed*. Thus, the solvent accessibility prediction problem turns out to be a binary classification problem in which each residue of a protein is categorized as buried (negative class) or exposed (positive class).

Various methods have been proposed for the prediction of solvent accessibility from the primary sequence of proteins [3]. One group of methods is based on the

F.A. Savacı (Ed.): TAINN 2005, LNAI 3949, pp. 141 – 148, 2006.

single sequence prediction of the solvent accessibility from local amino acid compositions. Single sequence methods identify local statistics from amino acid sequences and predict the solvent accessibility using different classification schemes, such as neural networks [1,11], Bayesian statistics [13], multiple linear regression [8] or support vector machines [17]. Richardson and Barlow has provided a baseline method which uses only the statistics inferred from the tendency of each amino acid to be buried or exposed [12]. The single sequence prediction accuracy is about 71% and this can be increased using multiple sequence information in the data set. Multiple sequence predictions use evolutionary information inferred from the profiles constructed by multiple sequence alignments. Multiple sequence methods increase the prediction accuracy up to about 79% [3]. However, using multiple alignments is computationally inefficient and it is not always guaranteed that informative profiles could be constructed in the given dataset.

In this work, a two-stage method is developed for the single sequence prediction of solvent accessibility and ~73% accuracy is achieved with an accessibility threshold of 22.2% on a nonredundant data set of 420 proteins. The first stage uses support vector machines to predict the two-class solvent accessibility using the residue features such as hydropathy scale and residue mass, as well as the neighborhood information from the left and right side of an amino acid. The second stage searches the maximal and unique amino acid sequence matches between the target protein and the other proteins in the data set. The SVM-based predictions are refined using the conservations over the maximal unique matches (MUMs).

2 Methods

Baseline, SVM-based and MUM-based methods are developed for the prediction of solvent accessibility. The methods are applied individually and as an ensemble to test their performance over the protein data set.

2.1 Baseline Predictions

The baseline predictions can be obtained using the solvent accessibility statistics of each amino acid in the selected data set. Solvent accessibility values are taken from DSSP database [7] and the statistics are extracted from the training set. DSSP gives a solvent accessibility value between 0 and 9 for each residue of the proteins such that the value of 0 refers to a completely buried (0%) residue, 1 refers to a solvent accessibility of (0-11.1]%, 2 refers to (11.1-22.2]% and so on. The tendency of an amino acid to be buried is determined simply by comparing the counts of buried and exposed occurrences of that amino acid in the training set. If the number of buried occurrences is higher than exposed ones, this amino acid is predicted as buried for all test cases. Otherwise, it is marked as exposed. According to the statistics, V, I, L, F, M, W, C are buried and G, A, P, S, T, N, Q, Y, H, D, E, K, R are exposed amino acids with an accessibility threshold of 22.2%. For 0% threshold, all are marked as buried, whereas only G is exposed for 55.5% threshold.

2.2 SVM-Based Prediction

Support vector machine (SVM) is a binary classifier which works based on the structural risk minimization principle [15]. An SVM non-linearly maps its n-dimensional input space into a high dimensional feature space. In this high dimensional feature space a linear classifier is constructed. Because of its power and robustness in performing binary classification and real valued function approximation tasks, it has been more popular in recent years and applied on many problems in computational biology [9,16,17]. In many applications, it has been shown that SVM is consistently superior to other supervised machine learning methods, such as Bayesian classifiers or neural networks.

To train the classifier which makes a separation between the buried and exposed classes (exposed used for positive and buried for negative classes), we used the feature vectors which represent the physicochemical properties of the amino acid, the binary code of the amino acid itself and the properties of left and right neighbors of the center amino acid to be predicted. In the previous work of Yuan et al [17], only amino acid codes were used for feature vectorization. The properties used in our vectorization step are listed in Table 1, where the hydropathy scales (free energy changes for transfer from oil to water for each amino acid) are taken from the Horton's book [6] and the relative residue mass values are given by Li and Pan [8]. For each 3-length amino acid string, a feature vector with a length of 66 (3x22) is used. 20 of the vector elements represent one of the 20 different amino acids and 2 elements are physicochemical properties explained before.

Table 1. Chemical and physical properties of amino acids used in the feature representations

Amino acid	Hydropathy Scale	Relative residue mass (W as 1.0)
G	0.67	0.00076
A	1.0	0.115
V	2.3	0.33
I	3.1	0.13
L	2.2	0.13
F	2.5	0.7
P	-0.29	0.323
M	1.1	0.577
W	0.0	1.0
S	-1.1	0.238
T	-0.75	0.346
N	-2.7	0.446
Q	-2.9	0.55
Y	0.08	0.82
H	-1.7	0.63
D	-3.0	0.446
E	-2.6	0.55
K	-4.6	0.48
R	-7.5	0.777
C	0.17	0.36

We used the SVM-*Gist* software implemented by Noble and Pavlidis (www.cs.columbia.edu/compbio/svm) in our tests. In the G*ist* software, a kernel function acts as the similarity score between the pairs of input vectors. The base kernel is normalized in order to make that each vector has a length of 1 in the feature space, that is,

$$K(X,Y) = \frac{X.Y}{\sqrt{(X.X)(Y.Y)}}$$

where *X* and *Y* are the input vectors, *K(.,.)* is the kernel function, and *"."* denotes the dot product.

To tune an SVM, the most significant parameter needed is the kernel function. We use radial basis function, which can be expressed with a modified kernel function *K'(X,Y)*, as follows:

$$K'(X,Y) = e^{-\frac{K(X,X)-2K(X,Y)+K(Y,Y)}{2\sigma^2}} + 1$$

where the width σ is the median Euclidean distance from any positive training example to the nearest negative example. Since the separating hyperplane of SVM is required to pass from the origin, the constant 1 is added to the kernel so that the data goes away from the origin.

2.3 MUM-Based Refinement

In spite of the fact that the data set we used is composed of non-homolog or remote homolog proteins, they may still share some conservative patterns between them. If these kinds of sequence conservations refer also to the conservations in solvent accessibility, we can use the statistics inferred from them to refine the incorrectly identified residues.

We use the maximal unique match definition, which is originally described in MUMer [4] to accelerate the alignment of long DNA sequences, to define the local conservations between two proteins. A maximal unique match between two sequences can be defined as the substring that appears only once in both sequences and not contained in any longer such substrings. Because of their uniqueness, maximal unique matches are important local similarities and give important clues about the structural conservations between two proteins.

Since we have already trained 3-letter strings in SVM applications, here, we extract only the matches longer than 5 amino acids and calculate the solvent accessibility statistics obtained from the middle-point of each maximal unique match. Among all maximal unique matches extracted from the dataset, the percentage of correctly identified residues is 79.4%. For any homolog data set, this percentage promises good prediction accuracy for solvent accessibility. However, in our data set, containing less or no homology, the number of maximal unique matches is relatively low. Therefore, the information gathered from the maximal unique matches can only be used for the refinement of the predictions made by other methods.

Suffix Trees

To find the maximal unique matches, we used a special data structure called suffix tree. A suffix tree is a compacted tree that stores all suffixes of a given text string. (An example suffix tree is shown in Figure 1.). It is a powerful and versatile data structure which finds application in many string processing algorithms, such as string matching, text compression and analyzing genetic sequences [5].

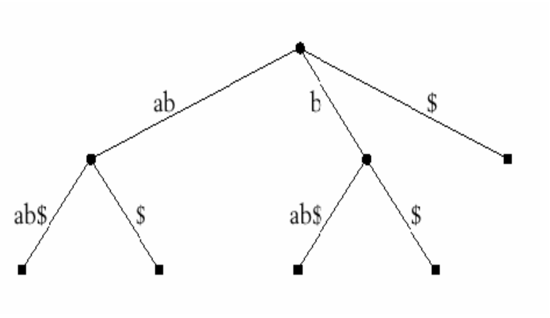

Fig. 1. Suffix tree of “abab$”

Let $\boldsymbol{A}$ be string of $\boldsymbol{n}$ characters, $\boldsymbol{A=s_1s_2...s_n}$, from an ordered alphabet $\boldsymbol{\Sigma}$ except $\boldsymbol{s_n}$. Let $\boldsymbol{\$}$ be a special character, matching no character in $\boldsymbol{\Sigma}$, and $\boldsymbol{s_n}$ be $\boldsymbol{\$}$. The suffix tree $\boldsymbol{T}$ of $\boldsymbol{A}$ is a tree with $\boldsymbol{n}$ leaves such that;

- Each path from the root to a leaf of $\boldsymbol{T}$ represents a different suffix of $\boldsymbol{A}$.
- Each edge of $\boldsymbol{T}$ represents a non-empty string of $\boldsymbol{A}$.
- Each non-leaf node of $\boldsymbol{T}$, except the root, must have at least two children.
- Substrings represented by two sibling edges must begin with different characters.

There are many algorithms for the construction of a suffix tree. We used Ukkonen’s linear-time construction algorithm [14] in our implementation.

Finding Maximal Unique Matches

To find the maximal unique matches between any two sequences, first, a generalized suffix tree is constructed for the sequences. This is simply done by concatenating two sequences with a dummy character (not contained in the alphabet) between them and constructing a suffix tree for newly created sequence. In our representation, a maximal unique match is a maximal pair in the concatenated sequence one of which appears before the dummy character and the other appears after that. The algorithm to find maximal pairs is given by Gusfield [5]. We used a variation of this algorithm considering the fact that each of the pair should appear in different sequences. The details of the algorithm can be found in the previous study of Oğul and Erciyes [10].

In MUM-based refinement stage, each protein in the test set is searched for the maximal unique matches with all other proteins in a pairwise fashion. The solvent accessibility of the residue appearing in the middle of a maximal unique match is determined by a simple voting scheme.

3 Experiment and Results

The baseline predictions, SVM-based predictions, and MUM-based refinements are applied into data set for a solvent accessibility threshold of 22.2%. The data set contains 420 proteins which have no pair with a sequence similarity above 25%. In SVM-based prediction stage, 15 proteins which are randomly selected from the dataset are used for training the SVM. Total number of training examples is 3067, where 1564 of them are exposed and 1503 of them are buried with 22.2% threshold. The remaining proteins are used for the tests. The proteins in the training and test sets are given in the Table 2 with their Protein Data Bank [2] identification numbers.

Table 2. Data set

Training Set	1acx, 1amp, 1aya, 1ctf, 1hmp, 1hmy, 1hnf, 1hor, 5lyz, 6cpa, 6dfr, 6tmn, 7rsa, 9api, 9wga,
Test Set	154l, 1aaz, 1add, 1ade, 1ahb, 1alk, 1amg, 1aor, 1aoz, 1asw, 1atp, 1avh, 1azu, 1bam, 1bbp, 1bcx, 1bdo, 1bds, 1bet, 1bfg, 1bmv, 1bnc, 1bov, 1bph, 1brs, 1bsd, 1cbg, 1cbh, 1cc5, 1cdl, 1cdt, 1cei, 1cel, 1cem, 1ceo, 1cew, 1cfb, 1cfr, 1cgu, 1chb, 1chd, 1chk, 1chm, 1cks, 1clc, 1cns, 1coi, 1col, 1com, 1cpc, 1cpn, 1cqa, 1crn, 1cse, 1csm, 1cth, 1ctn, 1ctm, 1ctn, 1ctu, 1cxs, 1cyx, 1daa, 1dar, 1del, 1dfj, 1dfn, 1dih, 1dik, 1din, 1dkz, 1dlc, 1dnp, 1dpg, 1dsb, 1dts, 1dup, 1dyn, 1eca, 1ece, 1ecl, 1ecp, 1edd, 1edm, 1edn, 1eft, 1efu, 1epb, 1ese, 1esl, 1etu, 1euu, 1fba, 1fbl, 1fc2, 1fdl, 1fdt, 1fdx, 1fin, 1fjm, 1fkf, 1fnd, 1fua, 1fuq, 1fxi, 1gal, 1gcb, 1gcm, 1gd1, 1gdj, 1gep, 1gfl, 1ghs, 1gky, 1gln, 1gmp, 1gnd, 1gog, 1gp1, 1gp2, 1gpc, 1gpm, 1grj, 1gtm, 1gtq, 1gym, 1han, 1hip, 1hcg, 1hcr, 1hiw, 1hjr, 1hpl, 1hsl, 1htr, 1hup, 1hvq, 1hxn, 1hyp, 1il8, 1ilk, 1inp, 1irk, 1isa, 1isu, 1jud, 1kin, 1knb, 1kpt, 1krc, 1kte, 1ktq, 1kuh, 1l58, 1lap, 1lat, 1lba, 1lbu, 1leh, 1lib, 1lis, 1lki, 1lpb, 1lpe, 1mai, 1mas, 1mct, 1mda, 1mdt, 1mjc, 1mla, 1mmo, 1mns, 1mof, 1mrr, 1mrt, 1msp, 1nal, 1nar, 1nba, 1ncg, 1ndh, 1nfp, 1nga, 1nlk, 1nol, 1nox, 1noz, 1oac, 1onr, 1otg, 1ovb, 1ovo, 1oxy, 1oyc, 1paz, 1pbp, 1pbw, 1pda, 1pdn, 1pdo, 1pga, 1pht, 1pii, 1pky, 1pmi, 1pnm, 1pnt, 1poc, 1pow, 1ppi, 1ppt, 1ptr, 1ptx, 1pyp, 1pyt, 1qbb, 1qrd, 1r09, 1rbp, 1rec, 1reg, 1req, 1rhd, 1rhg, 1rie, 1ris, 1rld, 1rlr, 1rpo, 1rsy, 1rvv, 1s01, 1scu, 1sei, 1ses, 1sfe, 1sft, 1sh1, 1smn, 1smp, 1spb, 1sra, 1srj, 1stf, 1stm, 1svb, 1tab, 1taq, 1tcb, 1tcr, 1tfr, 1tht, 1thx, 1tie, 1tif, 1tig, 1tii, 1tml, 1tnd, 1tnf, 1tpl, 1trb, 1trh, 1trk, 1tsp, 1tss, 1tul, 1tup, 1ubd, 1ubq, 1udh, 1umu, 1vca, 1vcc, 1vhh, 1vhr, 1vid, 1vjs, 1vmo, 1vnc, 1vok, 1vpt, 1wap, 1wfb, 1whi, 1wsy, 1xva, 1ypt, 1yrn, 1znb, 1zym, 256b, 2aai, 2aat, 2abk, 2adm, 2afn, 2ak3, 2alp, 2asr, 2bat, 2blt, 2bop, 2cab, 2ccy, 2cmd, 2cpo, 2cyp, 2dkb, 2dln, 2dnj, 2ebn, 2end, 2erl, 2fox, 2fxb, 2gbp, 2gcr, 2gls, 2gn5, 2gsq, 2hft, 2hhm, 2hip, 2hmz, 2hpr, 2i1b, 2ltn, 2mev, 2mhu, 2mlt, 2mta, 2nad, 2npx, 2olb, 2pab, 2pgd, 2phh, 2phy, 2pol, 2reb, 2rsl, 2rsp, 2scp, 2sil, 2sns, 2sod, 2spt, 2stv, 2tgi, 2tgp, 2tmd, 2tmv, 2trt, 2tsc, 2utg, 2wrp, 2yhx, 3ait, 3b5c, 3bcl, 3blm, 3cd4, 3chy, 3cla, 3cln, 3cox, 3eca, 3gap, 3hmg, 3icb, 3ink, 3mdd, 3pgk, 3pgm, 3pmg, 3rnt, 3tim, 4bp2, 4cpa, 4fis, 4gr1, 4pfk, 4rhv, 4rxn, 4sdh, 4sgb, 4ts1, 4xia, 5cyt, 5er2, 5ldh, 5sic, 6acn, 6cpp, 6cts, 6hir, 6rlx, 7cat, 7icd, 821p, 8adh, 9ins, 9pap

All resulting predictions are compared with the actual values of solvent accessibilities obtained from DSSP database. The accuracy is defined as the percentage of number of correctly identified residues among all residues. The experimental results are given in Table 3 with varying threshold values.

As we can see from the table, SVM-based predictions give an improvement of 2.8% over baseline predictions for a 22.2% threshold, which is the case of evenly distribution of buried and exposed classes. When applied on the baseline predictions, MUM-based refinement improves the baseline accuracy by 0.5%. The MUM-based refinement improves the SVM-based predictions by 0.4%. Overall improvement achieved by the combination of SVM-based and MUM-based predictions over the baseline accuracy is 3.2%. The methods are also tested for 0% and 55.5% thresholds and the results are given in the Table 3. For those threshold values, same improvement can not be achieved with SVM. This is probably due to the fact that the positive and negative examples are not evenly distributed for those threshold values.

Table 3. Results showing the accuracies achieved with different solvent accessibility thresholds

Threshold Method	0%	22.2%	55.5%
Baseline	75.3%	69.5%	79.6%
Previous SVM method (Yuan et al, 2002)	70.9%	71.4%	78.7%
New SVM method with extended features	71.6%	72.3%	79.1%
MUM-refinement over baseline	**75.7%**	70.0%	**79.8%**
MUM-refinement over new SVM	72.3%	**72.7%**	79.3%

Since there is no common benchmarking set for the solvent accessibility prediction, a direct comparison with the previous methods that used different data sets is not valid. According to the recent review of Chen et al [3], which reports a baseline prediction accuracy of 69.6% in their data set, the accuracies achieved with the tested methods are 71.5% for decision tree model and 71.2% for Bayesian statistics with a 20% threshold. We could make a fair comparison only with the baseline method and the previous SVM method of Yuan et al [17] with the same threshold over the same experimental setup (Table 3). Comparing with these results, our methods achieve slightly better accuracy.

4 Conclusion

Protein solvent accessibility is an important property for the annotation of newly extracted protein sequences. We introduce a new computational method for the prediction of solvent accessibility using solely the sequence information and report the results of the tests performed on a non-redundant protein set. The new method uses an improved SVM approach with extended features for the prediction of the accessibilities and refines the SVM predictions along with the pairwise conservations, i.e. maximal unique matches, between the sequences. The main reason for the improvement in SVM predictions is the incorporation of new physicochemical features of the protein residues in the vectorization phase. Although the maximal unique match refinement does not make a significant improvement on the accuracy, it promises good results when the sufficient number of homologs is found. Whenever the larger datasets are made available, this scheme can be used to obtain better refinements.

References

1. Ahmad S., Gromiha M.M.: NETASA: neural network based prediction of solvent accessibility. Bioinformatics 18 (2002) 819-824.
2. Berman H.M., Westbrook J., Feng Z., Gilliland G., Bhat T.N., Weissig H., Shindyalov I.N., Bourne P.E.: The Protein Data Bank. Nucleic Acids Research 28 (2000) 235-242.
3. Chen H., Zhou H., Hu X., Yoo I.: Classification comparison of prediction of solvent accessibility from protein sequences. 2nd Asia-Pacific Bioinformatics Conference, Dunedin, New Zelland (2004).
4. Delcher A., Kasif S., Fleishmann R., Peterson J., White O., Salzberg S.: Alignment of whole genomes. Nucleic Acids Research 27 (1999) 2369-2376.
5. Gusfield D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, (1997).
6. Horton H.B., Moran L.A., Ochs R.S., Rawn J.D., Scrimgeour K.G.: Principles of Biochemistry. Prentice Hall, (2002).
7. Kabsch W., Sander C.: Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers 22 (1983) 2577-637.
8. Li X., Pan X-M.: New method for accurate prediction of solvent accessibility from protein sequence. Proteins 42 (2001) 1-5.
9. Liao L., Noble W.S.: Combining pairwise sequence similarity and support vector machines for remote homology detection. Proc. 6th. Int. Conf. on Computational Molecular Biology, (2002) 225-232 .
10. Oğul H., Erciyes K.: Identifying all local and global alignments between two DNA sequences. Proc. 17th Int. Sym. on Computer and Information Sciences, (2001) 468-475.
11. Rost B., Sander C.: Conservation and prediction of solvent accessibility in protein families. Proteins 20 (1994) 216-226.
12. Richardson C.J., Barlow D.J.: The bottom line for prediction of residue solvent accessibility. Protein Engineering 12 (1999) 1051-1054.
13. Thompson M.J., Goldstein R.A.: Predicting solvent accessibility: higher accuracy using Bayesian statistics and optimized residue substitution classes. Proteins 25 (1996) 38-47.
14. Ukkonen E.: On-line construction of suffix-trees. Algorithmica 14 (1995) 249-60.
15. Vapnik V.: The nature of statistical learning theory. Spring-Verlag, New York (1995).
16. Ward J., McGuffin L. C., Buxton B. F., Jones D. T.: Secondary structure prediction with support vector machines. Bioinformatics 19 (2003) 1650-55.
17. Yuan Z., Burrage K., Mattick J.: Prediction of protein solvent accessibility using support vector machines. Proteins 48 (2002) 566-570.