A kernel framework for protein residue annotation

Huzefa Rangwala[1], Christopher Kauffman[2] and George Karypis[2]

[1] George Mason University, Fairfax, VA 22030, USA (rangwala@cs.gmu.edu)
[2] University of Minnesota, Minneapolis, MN 55414, USA

Abstract. Over the last decade several prediction methods have been developed for determining structural and functional properties of individual protein residues using sequence and sequence-derived information. Most of these methods are based on support vector machines as they provide accurate and generalizable prediction models. We developed a general purpose protein residue annotation toolkit (*Pro*SAT) to allow biologists to formulate residue-wise prediction problems. *Pro*SAT formulates annotation problem as a classification or regression problem using support vector machines. For every residue *Pro*SAT captures local information (any sequence-derived information) around the reside to create fixed length feature vectors. *Pro*SAT implements accurate and fast kernel functions, and also introduces a flexible window-based encoding scheme that allows better capture of signals for certain prediction problems. In this work we evaluate the performance of *Pro*SAT on the disorder prediction and contact order estimation problems, studying the effect of the different kernels introduced here. *Pro*SAT shows better or at least comparable performance to state-of-the-art prediction systems. In particular *Pro*SAT has proven to be the best performing transmembrane-helix predictor on an independent blind benchmark. **Availability:** `http://bio.dtc.umn.edu/prosat`

1 Introduction

Experimental methods to determine the structure and function of proteins have been out-paced with the abundance of available sequence data. As such, over the past decade several computational methods have been developed to characterize the structural and functional aspects of proteins from sequence information [26].

Support vector machines (SVMs) [28] along with other machine learning tools have been extensively used to successfully predict the residue-wise structural or functional properties of proteins [4, 15, 20]. The task of assigning every residue with a discrete class label or continuous value is defined as a *residue annotation* problem. Examples of structural annotation problems include the secondary structure prediction [11, 15, 22], local structure prediction [5, 14], and contact order prediction [18, 27]. Examples of function property annotation include prediction of interacting residues [20] (e.g., DNA-binding residues, and ligand-binding residues), solvent accessible surface area estimation [21, 25], and disorder prediction [4, 9].

We have developed a general purpose protein residue annotation toolkit called *Pro*SAT. This toolkit uses a support vector machine framework and is capable of predicting both a discrete label or a continuous value. *Pro*SAT allows use of any type of sequence information with residues for annotation. For every

residue, *Pro*SAT encodes the input information from the residue and its neighbors. We introduce a new flexible encoding scheme that differentially weighs information extracted from neighboring residues, based on the distance to the central residue. *Pro*SAT also uses an exponential second-order kernel function shown to be effective in capturing pairwise interactions between residues, and hence improve the classification and regression performance for the annotation problems [15].

To the best of our knowledge, *Pro*SAT is the first tool that is designed to allow life science researchers to quickly and efficiently train SVM-based models for annotating protein residues with any desired property. The kernel functions implemented are also optimized for speed, by utilizing fast vector-based operation routines within the CBLAS library [29]. *Pro*SAT [3] is made available as a pre-compiled binary on several different architectures and environments.

In this paper we report our evaluation studies highlighting the different features of *Pro*SAT on the disorder prediction [4] and contact order estimation [27] problem. *Pro*SAT shows a statistically significant improvement on both the disorder prediction (1%) and contact order estimation problems (20%) in comparison to previously established methods. We have also tested *Pro*SAT on the DNA-binding [20], and local structure prediction problem (results not reported here). *Pro*SAT improves over state-of-the-art transmembrane helix prediction methods [12], as evaluated by an independent benchmark [17]. Recently, *Pro*SAT was used to develop the best performing transmembrane-helix segment identification and orientation system called TOPTMH [1], and improve the comparative modeling ligand-binding regions of proteins [16]. The models trained by *Pro*SAT are also used to generate predictions for a webserver developed by us called MONSTER (Minnesota prOteiN Sequence annoTation servER) available at `http://bio.dtc.umn.edu/monster`.

2 Problem Definition and Notations

In this paper, we will refer to protein sequences by X and Y, and an arbitrary residue by x. Given a sequence X of length n, with it are associated derived features F, a $n \times d$ matrix where d is the dimensionality of the feature space. The features associated with the ith residue x_i are the ith row of the matrix F denoted as F_i. When multiple types of features are considered, the lth feature matrix is specified by F^l. In Figure 1 (a) we show the PSI-BLAST derived position specific scoring matrix of dimensions $n \times 20$ (discussed in Section 3.2).

In order to encode information for a residue *Pro*SAT uses the information from neighboring residues as well. *Pro*SAT uses a $w$mer-based encoding to capture sequence information for residue x_i to perform the residue-wise prediction. *Pro*SAT uses the $(2w + 1)$ rows of the matrix F, $F_{i-w} \ldots F_{i+w}$ to encapsulate the feature information associated with the $w$mer centered at residue x_i. This submatrix is denoted by $w\texttt{mer}(F_i)$ and is linearized to generate a vector of length $(2w + 1)d$, where d is the dimension of the matrix F.

As seen in Figure 1(b) for the circled residue, three residues above and below are also selected and the corresponding information from the feature matrix is extracted. Further Figure 1(c) represents the linearized submatrix as a vector which encodes the information for the problem.

[3] http://bio.dtc.umn.edu/prosat

3 Methods

We approach the protein residue annotation problem by utilizing local sequence information around each residue in a supervised machine learning framework. We use support vector machines (SVM) [28] in both classification and regression formulations to address the problem of annotating residues with discrete labels and continuous values respectively. We use the publicly available SVM^{light} program [10] for the discriminatory learning.

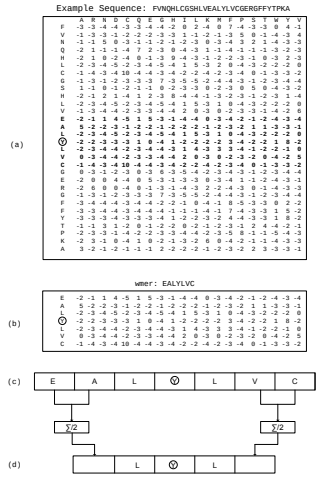

Fig. 1. (a) Input example sequence along with PSI-BLAST profile matrix of dimensions $n \times 20$, with a residue circled to show the encoding steps. (b) Example $w$mer of $w = 3$ and length seven, with extracted submatrix from the original PSI-BLAST matrix. (c) Encoded vector of length 7×20 formed by linearizing the submatrix (d) Flexible encoding showing three residues in the center using the finer representation, and two residues flanking the central residues on both sides using a coarser representation as an averaging statistic. Length of this vector equals 5×20.

3.1 Support Vector Classification and Regression

The task of assigning a label to the residue x from one of the K possible annotation labels is a typical multiclass classification problem. The general strategy is to build K one-versus-rest binary SVM classification models that assign a residue to be in a particular class or not. Let $\mathcal{A}^+$ refer to the residues with on particular label, the positive class, and $\mathcal{A}^-$ refer to the remaining residues, the negative class. In its dual formulation, a support vector machine learns a classification function $f(x)$ of the form

$$f(x) = \sum_{x_i \in \mathcal{A}^+} \lambda_i^+ \mathcal{K}(x, x_i) - \sum_{x_i \in \mathcal{A}^-} \lambda_i^- \mathcal{K}(x, x_i), \tag{1}$$

where λ_i^+ and λ_i^- are non-negative weights that are computed during training to provide the best possible prediction, and $\mathcal{K}(.,.)$ is a *kernel* function designed to capture the similarity between pairs of residues. Having learned the function $f(x)$, a new residue x is predicted to be positive or negative depending on whether $f(x)$ is positive or negative. The value of $f(x)$ also signifies the tendency of x to be a member of the positive or negative class and can be used to obtain a meaningful ranking of a set of the residues.

We use the error insensitive support vector regression ϵ-SVR [28] for learning a function $f(x)$ for estimation in case of determining a quantity, as in the case of solvent accessibility prediction problem. Given a set of training instances (x_i, y_i), where y_i is the continuous value to be estimated for residue x_i, the ϵ-SVR aims to learns a function of the form

$$f(x) = \sum_{x_i \in \Delta^+} \alpha_i^+ \mathcal{K}(x, x_i) - \sum_{x_i \in \Delta^-} \alpha_i^- \mathcal{K}(x, x_i), \tag{2}$$

where Δ^+ contains the residues for which $y_i - f(x_i) > \epsilon$, Δ^- contains the residues for which $y_i - f(x_i) < -\epsilon$, and α_i^+ and α_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function. The objective of the maximization is to determine the flattest $f(x)$ in the feature space and minimize the estimation errors for instances in $\Delta^+ \cup \Delta^-$. Hence, instances that have an estimation error satisfying $|f(x_i) - y_i| < \epsilon$ are neglected. The parameter ϵ controls the width of the regression deviation or tube.

3.2 Sequence-based Information

Pro`SAT` can use any general user-supplied features. In our empirical evaluation for a given protein X of length n we encode the sequence information using PSI-BLAST position specific scoring matrices, predicted secondary structure, and position independent scoring matrices like BLOSUM62. These feature matrices are referred to as $\mathcal{P}$, $\mathcal{S}$, and $\mathcal{B}$, respectively and are described below.

Position Specific Scoring Matrices The profile of a protein is derived by computing a multiple sequence alignment of it with a set of sequences that have a statistically significant sequence similarity, i.e., they are sequence homologs as ascertained by PSI-BLAST [2]. In Figure 1 (a) we show the PSI-BLAST derived position specific scoring matrix for a sequence of length n. The dimensions of this matrix $n \times 20$. For every residue the PSI-BLAST matrix captures evolutionary conservation information by providing a score for each of the twenty amino acids.

The profiles in this study were generated using the latest version of the PSI-BLAST [2] (available in NCBI's blast release 2.2.10 using `blastpgp -j 5 -e 0.01 -h 0.01`) searched against NCBI's NR database that was downloaded in November of 2004 and contains 2,171,938 sequences.

Predicted Secondary Structure Information We use YASSPP [15] to predict secondary structure and generate a position-specific secondary structure matrices. For a length n sequence, the result is $\mathcal{S}$, a $n \times 3$ feature matrix. The (i, j)th entry of this matrix represents the propensity for residue i to be in state j, where $j \in \{1, 2, 3\}$ corresponds to the three secondary structure elements: alpha helices, beta sheets, and coil regions.

Position Independent Scoring Matrices A less computationally expensive feature of protein sequences may be obtained from a position independent scoring matrix such as the BLOSUM62 substitution matrix. The primary motivation for using BLOSUM62-derived feature vectors is to improve the classification accuracy in cases where a sequence does not have a sufficiently large number of homologous sequences in NR. In these cases PSI-BLAST fails to compute a correct alignment for some segments of the sequence giving a misleading PSSM [9, 15]. To make effective use of *Pro*`SAT`'s capabilities we create a $n \times 20$ feature matrix, referred to as $\mathcal{B}$, where each row of the matrix is a copy of the BLOSUM62 row corresponding to the amino acid at that position in the sequence.

By using both PSSM- and BLOSUM62-based information, the SVM learner can construct a model that is partially based on non-position specific information. Such a model will remain valid in cases where PSI-BLAST could not generate correct alignments due to lack of homology to sequences in the nr database [15].

3.3 Kernel Functions

A kernel function computes a similarity between two objects and selection of an appropriate kernel function for a problem is key to the effectiveness of support vector machine learning. We consider several individual kernels of interest and then proceed to describe combinations of kernels used in this study. Throughout this section we use F and G be the feature matrix for sequences X and Y respectively. A specific residue of X is denoted x_i and its associated vector of features is F_i.

Window Kernel Our contribution in this work is a two-parameter linear window-kernel, denoted by $\mathcal{W}_{w,f}$ which computes the similarity between two w`mers`, w`mer`(x_i) and w`mer`(y_j) according to their features w`mer`(F_i) and w`mer`(G_j), respectively. The kernel function is defined as

$$\begin{aligned}\mathcal{W}_{w,f}(x_i, y_j) = &\sum_{k=-f}^{f} \langle F_{i+k}, G_{j+k} \rangle + \\ &\langle \sum_{k=f+1}^{w} F_{i+k}, \sum_{k=f+1}^{w} G_{j+k} \rangle + \\ &\langle \sum_{k=f+1}^{w} F_{i-k}, \sum_{k=f+1}^{w} G_{i-k} \rangle .\end{aligned} \tag{3}$$

The parameter w governs the size of the w`mer` considered in computing the kernel while f offers control over the fine-grained versus coarse-grained sections of the window. Rows within $\pm f$ contribute an individual dot product to the total similarity while rows outside this range are first summed and then their dot product is taken. In all cases $f \leq w$ and as f approaches w, the window kernel becomes simply a sum of the dot products, the most fine-grained similarity measure considered. This window encoding is shown in Figure 1(d) where the positions away from the central residue are averaged to provide a coarser representation, whereas the positions closer to the central residue provide a finer representation. The rationale behind this kernel design is that some problems may require only approximate information for sequence neighbors which are far away from the central residue while nearby sequence neighbors are more important. Specifying $f \ll w$ merges these distant neighbors into only a coarse contribution to the overall similarity, as it only accounts for compositional information and not the specific positions where these features occur. The window kernel is defined as a dot-product, which makes it equivalent to linear kernel with a feature encoding scheme that takes into account the two variable parameters, w and f. Hence, we can embed the dot-product based $\mathcal{W}$ within other complex kernel functions.

Exponential Kernels Another individual kernel we use extensively is the second order exponential kernel, $\mathcal{K}^{soe}$, developed in our earlier works for secondary structure and local structure information prediction [15, 23]. Given any base kernel function $\mathcal{K}$, we define $\mathcal{K}^2$ as

$$\mathcal{K}^2(x, y) = \mathcal{K}(x, y) + (\mathcal{K}(x, y))^2. \tag{4}$$

which is a second-order kernel in that it computes pairwise interactions between the elements x and y. We then define $\mathcal{K}^{soe}$ as

$$\mathcal{K}^{soe}(x, y) = \exp\left(1 + \frac{\mathcal{K}^2(x, y)}{\sqrt{\mathcal{K}^2(x, x)\,\mathcal{K}^2(y, y)}}\right) \tag{5}$$

which normalizes $\mathcal{K}^2$ and embeds it into an exponential space.

We also use the standard radial basis kernel function (*rbf*), defined for some parameter γ by $\mathcal{K}^{rbf}(x, y) = \exp(-\gamma||x-y||^2)$. By setting a specific γ parameter and using normalized unit length vectors the standard *rbf* kernel can be shown equivalent (upto a scaling factor) to a first order exponential kernel obtained by removing the $\mathcal{K}^2(x, y)$ term in Equation 4, and plugging the modified kernel in Equation 5.

In this paper, we denote the *soe* to be the kernel $\mathcal{K}^{soe}$ using the $\mathcal{W}_{w,f}$ as the base, *rbf* to be the kernel $\mathcal{K}^{rbf}$ using the normalized form with $\mathcal{W}_{w,f}$ as the base, and *lin* to be the base linear kernel $\mathcal{W}_{w,f}$.

3.4 Integrating Information

To integrate the different information, we use a linear combination of the kernels derived for different feature matrices. Consider two sequences with features F^l and G^l for $l = 1, \ldots, k$, our fusion kernel using the is defined

$$\mathcal{K}^{fusion}(x_i, y_j) = \sum_{l=1}^{k} \omega_l \, \mathcal{K}^{soe}(F_i^l, G_j^l) \tag{6}$$

where the weights ω_l are supplied by the user. Note the *soe* kernel in Equation 6 can be replaced by the *lin*, and *rbf* kernels.

In the future we intend to explore the possibility of automatically learning the weights ω_l. This can be done by using some of the recent multiple kernel integration work that combines heterogeneous information using semidefinite programming [19], second order cone programming [3], and semi-infinite linear programming [24].

4 Case Studies

*Pro*SAT was tested on a wide variety of local structure and function prediction problems. Here we present a case study on the disorder prediction, contact order estimation and transmembrane-helix prediction problems. We review the methods used for solving the problems, and provide comparative results by using standard benchmarks which are described below.

*Pro*SAT was also tested on the DNA-binding prediction problem [20], ligand-binding prediction problem, solvent accessibility surface area estimation [21,25], and local structure alphabet prediction problem [5]. The results of these experiments are not reported here for sake of brevity. *Pro*SAT showed comparable to the state-of-the-art prediction systems for the different problems.

4.1 Experimental Protocol

The general protocol we used for evaluating the different parameters, and features, as well as comparing to previously established studies remained fairly consistent across the different problems. In particular we used a n-fold cross validation methodology, where $1/n^{th}$ of the database in consideration was used for testing and the remaining dataset was used for training, with the experiment being repeated n times.

Table 1. Classification Performance on the Disorder Dataset.

	w	$f=1$		$f=3$		$f=5$		$f=7$		$f=9$		$f=11$	
		ROC	F1	ROC	F1	ROC	F1	ROC	F1	ROC	F1	ROC	F1
$\mathcal{P}^{lin}$	3	0.775	0.312	**0.800**	0.350	-	-	-	-	-	-	-	-
	7	0.815	0.366	**0.817**	0.380	0.816	0.384	0.816	0.383	-	-	-	-
	11	0.821	0.378	0.826	0.391	**0.828**	0.396	0.826	0.400	0.824	0.404	0.823	0.403
	13	0.823	0.384	0.829	0.398	0.832*	0.405	0.830	0.404	0.828	0.407	0.826	0.409
$\mathcal{P}^{rbf}$	3	**0.811**	0.370	0.811	0.369	-	-	-	-	-	-	-	-
	7	0.845	0.442	**0.849**	0.450	0.848	0.445	0.845	0.442	-	-	-	-
	11	0.848	0.464	0.855	0.478	0.858	0.482	**0.858**	0.480	0.855	0.470	0.853	0.468
	13	0.848	0.473	0.855	0.484	0.859	0.490	0.861*	0.492	0.860	0.487	0.857	0.478
$\mathcal{P}^{soe}$	3	0.815	0.377	**0.816**	0.379	-	-	-	-	-	-	-	-
	7	0.847	0.446	**0.852**	0.461	0.852	0.454	0.851	0.454	-	-	-	-
	11	0.848	0.469	0.856	0.482	0.860	0.491	**0.862**	0.491	0.861	0.485	0.862	0.485
	13	0.847	0.473	0.856	0.485	0.861	0.491	0.864	0.495	0.865*	0.494	0.864	0.492
$\mathcal{P}\,\mathcal{S}^{soe}$	3	0.836	0.418	**0.838**	0.423	-	-	-	-	-	-	-	-
	7	0.860	0.472	**0.862**	0.476	0.860	0.473	0.859	0.468	-	-	-	-
	11	0.861	0.490	0.867	0.496	**0.868**	0.498	0.868	0.495	0.866	0.488	0.865	0.485
	13	0.860	0.497	0.867	0.503	0.870	0.503	0.871*	0.503	0.870	0.498	0.868	0.492
$\mathcal{P}\,\mathcal{S}\,\mathcal{B}^{soe}$	3	**0.842**	0.428	0.841	0.428	-	-	-	-	-	-	-	-
	7	0.869	0.497	**0.870**	0.499	0.869	0.494	0.867	0.489	-	-	-	-
	11	0.871	0.516	0.875	0.518	**0.877**	0.517	0.877	0.512	0.874	0.508	0.873	0.507
	13	0.869	0.519	0.875	0.522	0.878	0.521	0.879**	0.519	0.879	0.518	0.876	0.514

DISPro [4] reports a *ROC* score of 0.878. The numbers in bold show the best models for a fixed w parameter, as measured by *ROC*. $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{S}$ represent the PSI-BLAST profile, BLOSUM62, and YASSPP scoring matrices, respectively. *soe*, *rbf*, and *lin* represent the three different kernels studied using the $\mathcal{W}_{w,f}$ as the base kernel. * denotes the best classification results in the sub-tables, and ** denotes the best classification results achieved on this dataset. For the best model we report a Q_2 accuracy of 84.60% with an *errsig* rate of 0.33.

4.2 Evaluation Metrics

We measure the quality of the methods using the standard receiver operating characteristic (*ROC*) scores. The *ROC* score is the normalized area under the curve that plots the true positives against the false positives for different thresholds for classification [8]. We also compute other standard statistics, and report the $F1$ score which takes into account both the precision and recall for the prediction problem.

The regression performance is assessed by computing the standard Pearson correlation coefficient (*CC*) between the predicted and observed true values for every protein in the datasets. We also compute the root mean square error *rmse* between the predicted and observed values for every proteins. The results reported are averaged across the different proteins and cross validation steps. For the *rmse* metric, a lower score implies a better quality prediction.

We also compute a statistical significance test, *errsig* to differentiate between the different methods. *errsig* is the significant difference margin for each score and is defined as the standard deviation divided by the square root of the number of proteins.

4.3 Disorder Prediction

Some proteins contain regions which are intrinsically disordered in that their backbone shape may vary greatly over time and external conditions. A disordered region of a protein may have multiple binding partners and hence can take part in multiple biochemical processes in the cell which make them critical in performing various functions [7]. Disorder region prediction methods like IUPred [6], Poodle [9], and DISPro [4] mainly use physiochemical properties of the amino acids or evolutionary information within a machine learning tool like bi-recurrent neural network or SVMs.

*Pro*SAT was evaluated on the disorder prediction problem by training binary classification model to discriminate between residues that belong to part of disordered region or not. For evaluating the disorder prediction problem we used the DisPro [4] dataset which consisted of 723 sequences (215612 residues), with the maximum sequence identity between sequence pairs being 30%.

We used the PSI-BLAST profile matrix denoted by $\mathcal{P}$, a BLOSUM62 derived scoring matrix denoted by $\mathcal{B}$, and predicted secondary structure matrix denoted by $\mathcal{S}$ feature matrices both independently, and in combinations . We varied the w, and f parameters for the $\mathcal{W}$, and also compared the *lin*, *rbf*, and *soe* kernels. Table 1 shows the binary classification performance measured using the *ROC* and F_1 scores achieved on the disorder dataset after a ten fold cross validation experiment, previously used to evaluate the DISPro prediction method.

Comparing the *ROC* performance of the $\mathcal{P}^{soe}$, $\mathcal{P}^{rbf}$, and $\mathcal{P}^{lin}$ models across different values of w and f used for parameterization of the base kernel ($\mathcal{W}$), we observe that the *soe* kernel shows superior performance to the *lin* kernel and slightly better performance compared to the normalized *rbf* kernel used in this study. This verifies results of our previous studies for predicting secondary structure [15] and predicting RMSD between subsequence pairs [23], where the *soe* kernel outperformed the *rbf* kernel.

The performance *Pro*SAT on the disorder prediction problem was shown to improve when using the $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{S}$ feature matrices in combination rather than individually. We show results for the $\mathcal{P}\,\mathcal{S}$ and $\mathcal{P}\,\mathcal{S}\,\mathcal{B}$ features in Table 1. The flexible encoding introduced by *Pro*SAT shows a slight merit for the disorder prediction problem. These improvements are statistically significant as evaluated by the *errsig* measure.

The best performing fusion kernel improves the accuracy by 1% in comparison to DisPro [4] that encapsulates profile, secondary structure and relative solvent accessibility information within a bi-recurrent neural network.

4.4 Contact Order Estimation

Pairs of residues are considered to be in contact if their C_β atoms are within a threshold radius, generally 12 Å. Residue-wise contact order [27] is an average of the distance separation between contacting residues within a sphere of set threshold. Previously, a support vector regression method [27] has used a combination of local sequence-derived information in the form of PSI-BLAST profiles [2] and predicted secondary structure information [11], and global information based on amino acid composition and molecular weight for good quality estimates of the residue-wise contact order value. Amongst other techniques, critical random networks [18] use PSI-BLAST profiles as a global descriptor for this estimation problem.

*Pro*SAT was used to train ϵ-SVR regression models for estimating the residue-wise contact order on a previously used dataset [27] using the fusion of $\mathcal{P}$ and $\mathcal{S}$ features, with a *soe* kernel. This dataset consisted of 680 sequences (120421 residues), and the maximum pairwise sequence identity for this dataset was 40%.

In Table 3 we present the regression performance for estimating the residue wise contact order by performing 15-fold cross validation. These results are eval-

uated by computing the correlation coefficient and rmse values averaged across the different proteins in the dataset.

Analyzing the effect of the w and f parameters for estimating the residue-wise contact order values, we observe that a model trained with $f < w$ generally shows better CC and $rmse$ values. The best models as measured by the CC scores are highlighted in Table 3. A model with equivalent CC values but having a lower f value is considered better because of the reduced dimensionality achieved by such models.

The best estimation performance achieved by our ϵ-SVR based learner uses a fusion of the $\mathcal{P}$ and $\mathcal{S}$ feature matrices and improves CC by 21%, and $rmse$ value by 17% over the ϵ-SVR technique of Song and Barrage [27]. Their method uses the standard *rbf* kernel with similar local sequence-derived amino acid and predicted secondary structure features. The major improvement of our method can be attributed to our fusion-based kernel setting with efficient encoding, and the normalization introduced in Equation 5.

4.5 Transmembrane-Helix Prediction

Proteins which span the cell membrane have proven difficult to crystallize in most cases and are generally too large for NMR studies. Computational methods to elucidate transmembrane protein structure are a quick means to obtain approximate topology. Many of these proteins are composed of a inter-cellular, extra-cellular, transition, and membrane portions where the membrane portion contains primarily hydrophobic residues in helices (a multi-class classification problem). Accurately labeling these four types of residues allows helix segments allows them to be excluded from function studies as they are usually not involved in the activity of the protein. MEMSAT [12] in its most recent incarnation uses profile inputs to a neural network to predict whether residues in a transmembrane protein are part of a transmembrane helical region or not.

Kernytsky and Rost have benchmarked a number of methods and maintain a server to compare the performance of new methods which we employ in our evaluation [17]. We evaluate *Pro*SAT using this independent static benchmark. Firstly, we perform model selection on a set of 247 sequences used previously by the Phobius algorithm [13]. We use the $\mathcal{P}^{soe}$ kernel with w and f parameters set to 7 to train a four-way classification model for predicting the residue to be in either the helical region, non-helical region, inter-cellular region, and extra-cellular region. Using the trained model we annotate each of the 2247 sequences in the static benchmark (no true labels known to us)[4]. The performance of *Pro*SAT is shown in Table 4, which is better in comparison to state-of-the-art methods. The predictions from *Pro*SAT were further smoother using a second-level model to build the best performing transmembrane helix identification system called TOPTMH [1]. The reader is encouraged to find more details about experimental results in the TOPTMH [1] study.

4.6 Runtime Performance of Optimized Kernels

We also benchmark the learning phase of *Pro*SAT on the disordered dataset comparing the runtime performance of the program compiled with and without the

[4] Static Benchmark for testing Transmembrane helix prediction at http://cubic.bioc.columbia.edu/services/tmh_benchmark

Table 2. Runtime Performance of *Pro*SAT on the Disorder Dataset (in seconds).

	w=f=11				w=f=13				w=f=15			
	#KER	NO	YES	SP	#KER	NO	YES	SP	#KER	NO	YES	SP
$\mathcal{P}^{lin}$	1.93e+10	83993	45025	1.86	1.92e+10	95098	53377	1.78	1.91e+10	106565	54994	1.93
$\mathcal{P}^{rbf}$	1.91e+10	79623	36933	2.15	1.88e+10	90715	39237	2.31	1.87e+10	91809	39368	2.33
$\mathcal{P}^{soe}$	2.01e+10	99501	56894	1.75	2.05e+10	112863	65035	1.73	2.04e+10	125563	69919	1.75

The runtime performance of *Pro*SAT was benchmarked for learning a classification model on a 64-bit Intel Xeon CPU 2.33 GHz processor. #KER denotes the number of kernel evaluations for training the SVM model. NO denotes runtime in seconds when the cblas library was not used, YES denotes the runtime in seconds when the cblas library was used, and SP denotes the speedup achieved using the cblas library.

Table 3. Residue-wise Contact Order Estimation Performance

	w	$f=1$		$f=3$		$f=5$		$f=7$		$f=9$		$f=11$	
		CC	rmse	CC	rmse	CC	rmse	CC	rmse	CC	rmse	CC	rmse
$\mathcal{P}\,\mathcal{S}^{soe}$	3	0.704	0.696	**0.708**	0.692	-	-	-	-	-	-	-	-
	7	0.712	0.683	0.719	0.677	**0.723**	0.672	.722	0.672	-	-	-	-
	11	0.711	0.681	0.720	0.673	**0.725**	0.667	0.725	0.666	0.724	0.666	0.722	0.667
	15	0.709	0.680	0.719	0.672	0.726**	0.665	0.726	0.664	0.725	0.664	0.723	0.664

CC and $rmse$ denotes the average correlation coefficient and rmse values. The numbers in bold show the best models as measured by CC for a fixed w parameter. $\mathcal{P}$, and $\mathcal{S}$ represent the PSI-BLAST profile and YASSPP scoring matrices, respectively. *soe*, *rbf*, and *lin* represent the three different kernels studied using the $\mathcal{W}_{w,f}$ as the base kernel. * denotes the best regression results in the sub-tables, and ** denotes the best regression results achieved on this dataset. For the best results the $errsig$ rate for the CC values is 0.003. The published results [27] uses the default rbf kernel to give $CC = 0.600$ and $rmse = 0.78$.

CBLAS subroutines. These results are reported in Table 2 and were computed on a 64-bit Intel Xeon CPU 2.33 GHz processor for the $\mathcal{P}^{lin}$, $\mathcal{P}^{rbf}$, and $\mathcal{P}^{soe}$ kernels varying the $w$mer size from 11 to 15. Table 2 also shows the number of kernel evaluations for the different models. We see speedups ranging from 1.7 to 2.3 with use of the CBLAS library. Similar experiments were performed on other environments and other prediction problems, and similar trends were seen.

5 Conclusions and Future Directions

In this work we have developed a general purpose support vector machine based toolkit for easily developing predictive models to annotate protein residue with structural and functional properties. *Pro*SAT was tested with different sets of features on several annotation problems. Besides the problems illustrated here *Pro*SAT was used for developing a webserver called MONSTER[5] that predicts several local structure and functional properties using PSI-BLAST profiles only. *Pro*SAT also showed success in predicting and modeling ligand-binding site regions from sequence information only [16].

The empirical results presented here showed the capability of *Pro*SAT to accept information in the form of PSI-BLAST profiles, BLOSUM62 profiles, and predicted secondary structure. *Pro*SAT was tested with the *soe*, *rbf*, and *lin* kernel function. In addition, the results showed that for some problems (contact order estimation), by incorporating local information at different levels of granularity with the flexible encoding, *Pro*SAT was able to achieve better performance when compared to the traditional fine-grain approach.

Presently we are studying different multiple kernel integration methods that would automatically weight the contribution of different information in Equation 6. An optimal set of weights can be learned using semi-definite programming [19], and semi-infinite linear programming [24]. Currently, *Pro*SAT automatically performs a grid search over the different parameters for selecting the

[5] http://bio.dtc.umn.edu/monster

Table 4. Performance of *Pro*SAT and TOPTMH on the trans-membrane helix prediction problem

Method	$\mathcal{P}^{soe}$	*TOPTMH*	MEMSAT3	TMHMM1	PHDpsihtm08	HMMTOP2	PHDhtm08
Q_2	84	84	83	80	80	80	78
REC	81	75	78	68	76	69	76
PRE	87	90	88	81	83	89	82

Q_2, REC, and PRE denote the per-residue accuracy, recall and precision respectively. Results for MEMSAT3 [12], TOPTMH [1] and $\mathcal{P}^{soe}$ were obtained by evaluating it on the TMH static benchmark [17] and submitting the results of prediction to the server. We use the $\mathcal{P}^{soe}$ kernel with $w = f = 7$. All the other results were obtained from the TMH static benchmark evaluation web-site. Note, TOPTMH [1] uses *Pro*SAT for performing per-residue annotation, and then uses a set of hidden markov models to improve the per-segment accuracy.

best model. The multiple kernel integration work can also be used to select the best model. This would allow the biologist to use *Pro*SAT effectively. Further like the TOPTMH [1] system, we would like to smooth the predictions obtained from the residue-level predictors. This can be done by training a second level model or incorporating domain specific rules. A second level SVM-based model [15] has been implemented in *Pro*SAT already, and preliminary results show good promise.

We believe that *Pro*SAT provides to the practitioners an efficient and easy-to-use tool for a wide variety of annotation problems. The results of some of these predictions can be used to assist in solving the overarching 3D structure prediction problem. In the future, we intend to use this annotation framework to predict various 1D features of a protein and effectively integrate them to provide valuable supplementary information for determining the 3D structure of proteins.

Acknowledgements

This work was supported by NSF EIA-9986042, ACI-0133464, IIS-0431135, NIH RLM008713A, NIH T32GM008347, the Digital Technology Center, University of Minnesota and the Minnesota Supercomputing Institute. Huzefa Rangwala was supported by a start-up funding from George Mason University.

References

1. Rezwan Ahmed, Huzefa Rangwala, and George Karypis. Toptmh: Topology predictor for transmembrane alpha-helices. In *European Conference in Machine Learning*, Antwerp, Belgium, 2008. (in press). Available at www.cs.umn.edu/ karypis.
2. S. F. Altschul, L. T. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–402, 1997.
3. F. Bach, G. R. G. Lanckriet, and M . I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *Proceedings of the 2004 International Conference on Machine Learning*, 2004.
4. J. Cheng, M. J. Sweredoski, and P. Baldi. Accurate prediction of protein disordered regions by mining protein structure data. *Data Mining and Knowledge Discovery*, 11(3):213–222, 2005.
5. A. G. de Brevern, C. Etchebest, and S. Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins*, 41(3):271–287, Nov 2000.

6. Z. Dosztnyi, V. Csizmok, P. Tompa P, and I. Simon. Iupred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics*, 21(16):3433–3434, 2005.
7. A. K. Dunker, C. J. Brown, J. D. Lawson, L. M. Iakoucheva, and Z. Obradovic. Intrinsic disorder and protein function. *Biochemistry*, 41(21):6573–6582, 2002.
8. M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computational Chemistry*, 20:25–33, 1996.
9. S. Hirose, K. Shimizu, S. Kanai, Y. Kuroda, and T. Noguchi. Poodle-l: a two-level svm prediction system for reliably predicting long disordered regions. *Bioinformatics*, 23(16):2046–2053, 2007.
10. T. Joachims. *Advances in Kernel Methods: Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.
11. David T. Jones. Protein secondary structure prediction based on position-specific scoring matricies. *J. Mol. Biol.*, 292:195–202, 1999.
12. David T Jones. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics*, 23(5):538–544, Mar 2007.
13. L. Kall, A. Krogh, and E. L. L. Sonnhammer. A combined transmembrane topology and signal peptide prediction method. *Journal of Molecular Biology*, 338:1027–1036, 2004.
14. Rachel Karchin, Melissa Cline, Yael Mandel-Gutfreund, and Kevin Karplus. Hidden markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins*, 51(4):504–514, Jun 2003.
15. George Karypis. Yasspp: better kernels and coding schemes lead to improvements in protein secondary structure prediction. *Proteins*, 64(3):575–586, Aug 2006.
16. Chris Kauffman, Huzefa Rangwala, and George Karypis. Improving homology models for protein-ligand binding sites. In *LSS Comput Syst Bioinformatics Conference*, number 08-012, San Francisco, CA, 2008. (in press). Available at www.cs.umn.edu/ karypis.
17. Andrew Kernytsky and Burkhard Rost. Static benchmarking of membrane helix predictions. *Nucleic Acids Res*, 31(13):3642–3644, Jul 2003.
18. A. R. Kinjo and K. Nishikawa. Crnpred: highly accurate prediction of one-dimensional protein structures by large-scale critical random networks. *BMC Bioinformatics*, 7(401), 2006.
19. G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the 2004 Pacific Symposium on Biocomputing*, 2004.
20. Yanay Ofran, Venkatesh Mysore, and Burkhard Rost. Prediction of dna-binding residues from sequence. *Bioinformatics*, 23(13):i347–353, 2007.
21. G. Pollastri, P. Baldi, P. Farselli, and R. Casadio. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins: Structure, Function, and Genetics*, 47:142–153, 2002.
22. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural network and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47:228–235, 2002.
23. Huzefa Rangwala and George Karypis. frmsdpred: Predicting local rmsd between structural fragments using sequence information. *Proteins*, Feb 2008.
24. G. Ratsch, S. Sonnenburg, and C. Schafer. Learning interpretable svms for biological sequence classification. *BMC Bioinformatics*, 7(S9), 2006.
25. B. Rost. Phd: predicting 1d protein structure by profile based neural networks. *Meth. in Enzym.*, 266:525–539, 1996.
26. T. Schwede, J. Kopp, N. Guex, and M. C. Peltsch. Swiss-model: An automated protein homology-modeling server. *Nucleic Acids Research*, 31(13):3381–3385, 2003.

27. J. Song and K. Burrage. Predicting residue-wise contact orders in proteins by support vector regression. *BMC Bioinformatics*, 7(425), 2006.
28. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
29. R. Clint Whaley and Jack Dongarra. Automatically Tuned Linear Algebra Software. In *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, 1999. CD-ROM Proceedings.