

PROSAT: A Generalized Framework for Protein Sequence Annotation

Huzefa Rangwala, Christopher Kauffman and George Karypis

Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455
rangwala@cs.umn.edu, kauffman@cs.umn.edu, karypis@cs.umn.edu

Abstract

Motivation: *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. These protein residue annotation problems are often formulated as either classification or regression problems and solved using a common set of techniques.*

Methods: *We developed a generalized protein sequence annotation toolkit (PROSAT) for solving classification or regression problems using support vector machines. The key characteristic of our method is its effective use of window-based information to capture the local environment of a protein sequence residue. This window information is used with several kernel functions available within our framework. We show the effectiveness of using the previously developed normalized second order exponential kernel function and experiment with local window-based information at different levels of granularity.*

Results: *We report empirical results on a diverse set of classification and regression problems: prediction of solvent accessibility, secondary structure, local structure alphabet, transmembrane helices, DNA-protein interaction sites, contact order, and regions of disorder are all explored. Our methods show either comparable or superior results to several state-of-the-art application tuned prediction methods for these problems. PROSAT provides 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.*

Availability: <http://bio.dtc.umn.edu/monster>

Contact: rangwala@cs.umn.edu

1 Introduction

Protein structure and function prediction are grand challenges in computational biology [35, 36]. Residue-wise prediction problems are frequently more tractable than full prediction of three-dimensional structure and fill a supporting role for full prediction efforts.

Residue properties mostly take the form of either discrete label or continuous values and the task of assigning these properties is called protein sequence annotation. Each type of property has inspired a variety of computational methods [22, 33] to annotate protein sequences with the correct labels or values. Familiar examples include secondary structure prediction [30, 18, 13] and solvent accessibility prediction [24, 29, 33]. Though specifics vary, the crux of most

methods is to use sequence conservation signals to generate predictions. From problem to problem, the amount of sequence information required to generate an accurate and general model may vary substantially.

Our work develops a generalized protein sequence annotation framework using kernel-based techniques. The framework accepts any sequence information in the form of feature matrices and is capable of generating either discrete or continuous valued annotations. Kernels which compare variable-width windows around a sequence positions are used to determine what level of local information is necessary for accurate predictions. In some cases, only rough information about distant sequence neighbors may be required for accurate predictions. We explore this issue by examining the performance trade-off between fine-grained near-neighbor and coarse-grained distant-neighbor information.

As part of this work, we developed a protein sequence annotation toolkit, called PROSAT, that is applicable to any general annotation problem. We report empirical results on a wide range of prediction problems including annotation for solvent accessibility [29, 33], local structure alphabet [17, 5], transmembrane helices [14], DNA-protein interaction sites [26], contact order [39], and disordered regions [4, 10]. We use previously established datasets and that have pairwise sequence identity ranging from 25% to 40% for our evaluation.

Our results show the improvement in classification and estimation performance on the disordered and residue-wise contact order prediction problems by allowing the flexible representation introduced by us to capture the coarse-grained distant-neighbor information. We also report better than current state-of-the-art prediction results on an independently tested static benchmark for transmembrane helices prediction. Our generalized prediction framework shows an improvement compared to the well-established prediction methods on the residue-wise contact order, solvent accessibility, transmembrane helices, and local structure alphabet prediction problems. The results reported in this study are also shown to be statistically significant.

2 Problem Definitions

2.1 Notations

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 associate with the i th residue x_i are the i th row of the matrix F

denoted as F_i . When multiple types of features are considered, the l th feature matrix is specified by F^l .

Frequently we wish to capture local information around the i th residue, x_i by defining a subsequence from x_{i-w} to x_{i+w} . This $(2w + 1)$ -length subsequence is referred to as a *wmer* and is denoted by $wmer(x_i)$. The features associated with $wmer(x_i)$ are the rows of F , F_{i-w} to F_{i+w} and are denoted as $wmer(F_i)$.

2.2 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]. Accurate prediction of disordered regions can relieve some of the bottlenecks caused during high-throughput proteome analysis.

Several studies [32, 41] have shown the differences in sequences for ordered and disordered regions. As such, a large number of computational approaches have been developed to predict the disordered segments using sequence information. Predicting disordered regions forms part of the biennial protein structure prediction experiment CASP¹. Disorder region prediction methods mainly use physiochemical properties of the amino acids or evolutionary information. In particular IUPred [6] uses a pairwise energy function derived from amino acid composition, Poodle [10] employs a combination of different physiochemical properties as features for a SVM-based learning and prediction approach. Another disordered prediction tool, DISPro [4] utilizes a combination of evolutionary and sequence-derived features within a recurrent neural network.

2.3 Protein-DNA Interaction Site Prediction

When it is known that the function of a protein is to bind to DNA, it is highly desirable from an experimental point of view to know which parts of the protein are involved in the binding process. These interaction sites usually involve protein residues which come into contact with DNA and stabilize the complex due to favorable interactions with DNA. Sequence-based methods are to identifying the most likely binding residues as the full structure of the protein is rarely known. Accurate methods that do so would allow an experimentalist to alter the protein behavior by mutating only a few residues.

The usual approach for a machine learning approach is to define a cutoff distance from DNA. If parts of a protein residue are within this cutoff, it is considered an interacting residue and is otherwise considered non-interacting, a binary classification problem. DISIS [26] uses support vector machines and a radial basis function kernel with PSSMs, predicted secondary structure, and predicted solvent accessibility as input features. This framework is directly comparable

to our own along with neural network method of Ahmad and Sarai [2] which employs only PSSMs. Researchers have also utilized structure information such as the structural neighbors in DISPLAR [40] and the solvent accessibility using in the earlier work of Ahmad et al. [1].

2.4 Transmembrane Helix Prediction

Proteins which span the cell membrane have proven to be quite 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 an inter-cellular, extra-cellular, and membrane portions where the membrane portion contains primarily hydrophobic residues in helices. Accurately predicting these helix segments allows them to be excluded from function studies as they are usually not involved in the activity of the protein.

MEMSAT [14] 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 helix, interior or exterior loop, or interior or exterior helix caps. Kernysky 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 [19].

2.5 Local Structure Alphabets

The notion of local, recurring substructure in proteins has existed for many years primarily in the form of the secondary structure classifications. With the advent of fragment assembly methods for tertiary structure prediction [37], there has been increased interest in methods for predicting the backbone conformation of a fixed length section of protein. This extended local structure, usually a superset of traditional 3-state secondary structure, can be a significant first step towards full tertiary structure.

Many local structure alphabets have been generated by careful manual analysis of structures such as the alphabet of DSSP [15] while others have been derived through purely computational means. One such example are the Protein Blocks of de Brevern et al. [5] which were constructed through the use of self-organizing maps, a clustering technique. The method uses residue dihedral angles during clustering and attempts to account for order dependence between local structure elements which should improve predictability. Karchin et al. used neural nets to predict local structure for a variety of alphabets [17]. They found Protein Blocks to be the best choice according to their 'bits saved per position,' a measure of how much prediction improvement there is for the alphabet over simply predicting the most frequent character.

2.6 Relative Solvent Accessibility Prediction

Solvent accessibility determines the degree to which a residue in a protein structure can interact with a solvent molecule. This is important, as it can ascertain the local shape of protein

¹<http://predictioncenter.org>

based on whether the residue is buried/exposed. The residue-wise notion of solvent accessibility is defined by DSSP [15] by determining the accessible surface area relative to the maximum possible surface area obtainable for the specific amino acid residue.

Predicting solvent accessibility can be formulated as a classification problem by defining buried or exposed classes by thresholding on the relative solvent accessibility value (normally 16% or 25%), and can also be a regression or density estimation problem of attempting to determine the percentage value using sequence information only. There are many methods available for solvent accessibility prediction that deploy a wide range of learning methods including neural networks [33], bi-recurrent neural networks [29], information theory statistics [24] and support vector machines [25] using the set of standard sequence derived features.

2.7 Residue-wise Contact Order Prediction

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 [21] is an average of the distance separation between contacting residues within a sphere of set threshold. It defines the extent to which a residue makes long-range contacts in native protein structure, and can be used to set up constraints in the overarching three-dimensional structure prediction problem, and explain protein-folding rates [28].

A support vector regression method [39] has used a combination of local sequence-derived information in the form of PSI-BLAST profiles [3] and predicted secondary structure information [13], 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 [22] use PSI-BLAST profiles as a global descriptor for this estimation problem.

3 Methods and Algorithms

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) [11, 42] 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 [12] for the discriminatory learning.

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.

For a particular class, positive residues \mathcal{A}^+ are defined as members of that class while the negative residues \mathcal{A}^- are members of other classes. The task of support vector clas-

sification is to learn a function $f(x)$ of the form

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

where λ_i^+ and λ_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function, and $\mathcal{K}(\cdot, \cdot)$ is the *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 [42, 38] 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 learn a function of the form

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

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.

In this work we focus on several key aspects related to the formulation and solution of the classification and regression problems. In particular we explore different types of sequence information associated with the residues, develop efficient ways to encode this information to form fixed length feature vectors, and design sensitive kernel functions to capture the similarity between residues in the feature spaces.

3.2 Sequence-based Information

PROSAT 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.

3.2.1 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 [3]. For a sequence of length n , PSI-BLAST generates a position-specific scoring matrix \mathcal{P} of dimensions $n \times 20$, where the 20 columns

of the matrix correspond to the twenty amino acids. The profiles in this study were generated using the latest version of the PSI-BLAST [3] (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.

3.2.2 Predicted Secondary Structure Information

We use YASSPP [18] 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.

Predicted secondary structure is an example of a local structure alphabet and plays a critical role in protein structure prediction. YASSPP [18] has an identical framework to PROSAT and is one of the best performing secondary structure prediction methods with a reported Q_3 accuracy of 80%.

3.2.3 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 [10, 18]. To make effective use of PROSAT’s capabilities we create a $n \times 20$ feature matrix where each row of the matrix is a copy of the BLOSUM62 row corresponding to the amino acid at that position in the sequence. This feature matrix is referred to as \mathcal{B} .

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.

3.3 Kernel Functions

For a pair of sequences X and Y , let a specific set of derived features for the sequences be matrices F and G , respectively. To simplify notation we use F_i to indicate the i th row of matrix F , which corresponds to the features associated with the i th residue of X . 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.

Our first contribution is a two-parameter linear window-kernel, denoted by $\mathcal{W}_{w,f}$ which computes the similarity between two w mers, $wmer(x_i)$ and $wmer(y_j)$ according to their features $wmer(F_i)$ and $wmer(G_j)$, respectively. The

kernel function is defined as

$$\mathcal{W}_{w,f}(x_i, y_j) = \sum_{k=-f}^f \langle F_{i+k}, G_{j+k} \rangle + \left\langle \sum_{k=f+1}^w F_{i+k}, \sum_{k=f+1}^w G_{j+k} \right\rangle + \left\langle \sum_{k=-w}^{-f-1} F_{i+k}, \sum_{k=-w}^{-f-1} G_{j+k} \right\rangle. \quad (3)$$

The parameter w governs the size of the $wmer$ 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.

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.

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 [18, 31]. Given any base kernel function \mathcal{K} , we define \mathcal{K}^2 as

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

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

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

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 $\mathcal{K}^{rbf}(x, y) = \exp(-\gamma||x - y||^2)$. In our studies we notice that the classification and regression performance generally improves using unit length normalized vectors. By setting the γ parameter and using normalized unit length vectors it can be shown that the standard rbf kernel is equivalent (up to a scaling factor) to a first order exponential kernel which is obtained by replacing $\mathcal{K}^2(x, y)$ with only the first-order term as $\mathcal{K}(x, y)$ in Equation 4, and plugging this modified $\mathcal{K}^2(x, y)$ in the normalization framework of Equation 5.

In this paper, we mainly investigate the performance of \mathcal{K}^{soe} and \mathcal{K}^{rbf} kernels with $\mathcal{W}_{w,f}$ as the base kernel. From here forward, 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 $\mathcal{W}_{w,f}$.

We also investigate the use of fusion kernels which are generated via a linear combination of other kernels. In our case, we use a fusion of second-order exponential kernels on different features of a protein sequence. Considering two sequences with features F^l and G^l for $l = 1, \dots, k$, our fusion kernel is defined

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

where the weights ω_l are supplied by the user. In most cases, these weights are equal but they may be altered according to domain-specific information.

4 Experimental Design

4.1 Datasets

Our empirical evaluations are performed for different sequence annotation problems on previously defined datasets. Table 1 presents information regarding the source and key features of different datasets used in our cross validation and comparative studies. The datasets were used in previous studies, and we ensured that the pairwise sequence identities for the different datasets was less than 40%. Even though, the average sequence identity for the transmembrane-helix dataset was 80%, the dataset was only used to train a model for a blind independent evaluation on the static benchmark [19].

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. The number of cross validation was set based on the method that had used the same dataset previously for comparative purposes. In our experimental results we report results from previous studies.

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 [9]. The *ROC* score reported is averaged across the different classes and folds. We also compute other standard statistics, including precision as $TP/(TP + FP)$, and recall as $TP/(TP + FN)$. We also evaluate the accuracy of K -way multiclass classification by as $Q_K = (\sum_{i=1}^K TP_i)/(TotalResidues)$. Here, TP, FP, TN, FN denote the standard true positives, false positives, true negatives,

and false negatives, respectively. We also compute the F_1 score given as $2 * Precision * Recall / (Precision + Recall)$

The *ROC* score serves as a good quality measure in case of unbalanced classes, where measuring the accuracy or Q_K , especially in case of binary classification model may be skewed by predicting a particular class with larger number of instances. In such cases, it is essential to observe the precision and recall values, which penalize the classifiers for under-prediction as well as over-prediction. The F_1 score is a weighted average of precision and recall lying between 0 and 1, and also is a good measure for different classification problems.

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.

For the best performing models, we also report the *errsig* rate as the significant difference margin for Q_K and *CC* scores (to distinguish between two methods). *errsig* is defined as the standard deviation divided by the square root of the number of proteins (σ/\sqrt{N}), and can help us assess how significant the differences between the best performing models and the other models, as well as serves a reference for future studies.

5 Discussion and Results

For all the problems, we perform a comprehensive set of experiments encompassing a range of parameters, to determine the kernel type, features, and \mathcal{W} parameters (i.e., w and f).

5.1 Disorder Prediction Performance

Table 2 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.

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.

Comparing the characteristics of the different features keeping the kernel fixed to *soe*, we can notice that use of \mathcal{P} gives better classification performance compared to the \mathcal{S} and \mathcal{B} features. However, integrating features i.e., use of fusion kernels with $\mathcal{P}\mathcal{B}$, $\mathcal{P}\mathcal{S}$, and $\mathcal{P}\mathcal{S}\mathcal{B}$ tends to improve the disorder prediction over the kernels that use only one set of features, with the best results achieved by a combination of all three features.

An interesting trend can be observed for the \mathcal{B}^{soe} and \mathcal{S}^{soe} results. As we increase the w parameter, keeping the f parameter fixed to a low value of one or three, the percentage

Table 1: Problem-specific Datasets.

Problem	Source	#C	#Seq	#Res	#CV	%
Disorder Prediction	DisPro [4]	2	723	215612	10	30
Protein-DNA Site	DISIS [26]	2	693	127240	3	20
Residue-wise Contact	SVM [39]	∞	680	120421	15	40
Solvent Accessibility	RS126 [34]	∞	126	23356	7	25
Solvent Accessibility	RS126 [34]	2	126	23356	7	25
Local Structure	Profnet [27]	16	1600	286238	3	40
Transmembrane Helix	Phobius [16]	4	247	95025	3	80*
Transmembrane Helix	Static Test [19]	2	2247	238084	-	-

#C, #Seq, #Res, #CV, and % denote the number of classes, sequences, residues, number of cross validation folds, and the maximum pairwise sequence identity between the sequences, respectively. ∞ represents the regression problem. *Even though the percent identity between sequence-pairs is high, the dataset is used only for training and an independent evaluation is performed on the static benchmark [19].

increase in the *ROC* value for the \mathcal{B} features is higher, which suggests that the \mathcal{B} features are more suited to be adopted in a coarse setting.

The best performing fusion kernel shows comparable performance to DisPro [4] that uses a bi-recurrent neural network to encapsulate profile, secondary structure and relative solvent accessibility information.

5.2 Contact Order Performance

In Table 3 we present the regression performance for estimating the residue wise contact order. These results are evaluated 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 after 15-fold cross-validation 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 [39]. 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. In our setting, using the default parameters for the γ , regression tube, and regularization parameters always lead to over-fitting of the data with the original *rbf* kernel. This trend has been noted in our previous studies [31].

We also tested the methods using only the \mathcal{P} features, and addition of \mathcal{B} features but did not see a significant improvement in the contact order estimation results and hence, do not report these results here.

5.3 Transmembrane Helix Performance

To predict transmembrane proteins helices with PROSAT, we set up a multi-class classification problem to differentiate be-

tween the helical and non-helical regions of transmembrane proteins. In particular, to determine the orientation and topological structure of the helices we used a dataset that annotated the intermediate helical regions as cytoplasmic/non-cytoplasmic resulting in four classes.

We performed a three-fold cross validation study for the four-way multi-class classification problem on a dataset consisting of only transmembrane proteins [16]. Note that our training set makes it difficult to develop models for differentiating between globular and transmembrane proteins, as well as signal-peptide proteins as highlighted by Phobius [16], which uses a combination of hidden Markov models and neural networks for discriminating between the different residues.

Table 4 (a) shows the classification performance evaluated using the Q_4 accuracy and *ROC* scores for the \mathcal{P}^{soe} kernel. Based on the classification performance metrics, we see that the better models have a finer representation of the w mers, i.e., where $w = f$.

To obtain the predictions for sequences in the static benchmark [19]² we used the \mathcal{P}^{soe} kernel with w and f parameters set to 7. We used all the 247 transmembrane proteins available in the Phobius dataset to build a four-way classification model, and annotated the 2247 sequences present in the static benchmark (which provides independent evaluation). For evaluation, we used a mapping from four to two classes rather than building a binary classification model. Residues marked as helices were mapped to helices while all others, including intermediate residues, were mapped to non-helices. One of the best membrane-helix prediction program, MEMSAT3 [14], was also evaluated for comparison purposes as these results were not reported in the static benchmark. MEMSAT3 also annotates residues into multiple classes: outside/inside loop, outside/inside helix cap, and internal helix. Loops were mapped to non-helices while other were mapped to helices.

Table 4 (b) shows some of the best performing schemes in comparison to our prediction method, denoted \mathcal{P}^{soe} , evaluated by an independent server on the static benchmark (We do not have the true predictions available for these sequences).

²Static Benchmark for testing Transmembrane helix prediction at http://cubic.bioc.columbia.edu/services/tmh_benchmark/index.html

Table 2: 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{B}^{soe}	3	0.753	0.314	0.752	0.312	-	-	-	-	-	-	-	-
	7	0.810	0.427	0.815	0.434	0.816	0.435	0.815	0.429	-	-	-	-
	11	0.820	0.459	0.827	0.465	0.831	0.468	0.832	0.472	0.832	0.472	0.831	0.471
	13	0.821	0.465	0.829	0.469	0.833	0.473	0.835	0.473	0.836	0.476	0.836*	0.478
\mathcal{S}^{soe}	3	0.786	0.332	0.791	0.342	-	-	-	-	-	-	-	-
	7	0.806	0.387	0.812	0.397	0.814	0.395	0.814	0.389	-	-	-	-
	11	0.811	0.417	0.818	0.424	0.821	0.426	0.823	0.427	0.822	0.420	0.821	0.415
	13	0.812	0.434	0.821	0.438	0.825	0.436	0.827*	0.437	0.827	0.434	0.826	0.428
$\mathcal{P} \mathcal{B}^{soe}$	3	0.825	0.399	0.824	0.395	-	-	-	-	-	-	-	-
	7	0.862	0.487	0.865	0.491	0.865	0.487	0.863	0.481	-	-	-	-
	11	0.864	0.502	0.869	0.509	0.872	0.513	0.873	0.514	0.873	0.513	0.873	0.510
	13	0.863	0.509	0.869	0.514	0.873	0.517	0.875	0.518	0.876*	0.518	0.876	0.519
$\mathcal{P} \mathcal{S}^{soe}$	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.

We obtain the best Q_2 accuracy evaluated on a per-residue basis, and have good precision and recall scores as well.

5.4 Solvent Accessibility Performance

We approached solvent accessibility as both a labelling problem and a regression problem. For labelling, we chose varying cutoff thresholds to define each residue as either buried if at or below the threshold or exposed if above the threshold. For regression, PROSAT was used to generate continuous valued estimates of each residues relative accessible surface area. We explored \mathcal{P}^{rbf} and also \mathcal{P}^{soe} but restricted the study to models where $w = f$. In each case, 7-fold cross-validation was performed on the full RS126 dataset.

The results for both classification and regression are shown in Table 5 along with a leading method for solvent accessibility prediction [20]. The general trend appears to be that prediction performance by \mathcal{P}^{rbf} is slightly exceeded by \mathcal{P}^{soe} . The best window size appears to be $w = 7$ for both our kernels for prediction. Both kernels exceed the performance of the previously published SVM method which uses an rbf kernel with window length of 15 and profiles with some predicted local structure as inputs. For regression, the \mathcal{P}^{rbf} outperforms \mathcal{P}^{soe} and increasing window size improves the per-

formance.

5.5 Protein-DNA Interaction Sites Performance

Analyzing the ROC and F_1 scores obtained on the protein-DNA interaction site prediction problem in Table 6, we observe that for the lin kernels the classification accuracy decreases with increasing w sizes but fixed f parameters. This suggests that for predicting protein-DNA interaction sites, finer order-specific information holds more value compared to the coarser information. This trend was reversed in the case of disorder prediction where coarser information did have some benefit over entirely using the fine information. This is likely due to the inherent nature of these properties.

Further, the lin kernel for a small $w = 3$ value shows better results than the soe and rbf kernel. The linear kernel with the coarse information can extract some of the pairwise information that is extracted by the rbf and soe kernels. The value of w plays an important part in reducing the size of feature vectors and hence, the computational complexity. As such, models with lower w values may be preferred over models with higher w values when the classification accuracy gap is not large.

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
$\mathcal{P} \mathcal{S}^{lin}$	3	0.683	0.720	0.686	0.718	-	-	-	-	-	-	-	-
	7	0.685	0.714	0.694	0.707	0.702	0.698	0.703	0.697	- 2	-	-	-
	11	0.683	0.713	0.695	0.703	0.704	0.694	0.705	0.692	0.704	0.691	0.704	0.692
	15	0.680	0.714	0.694	0.703	0.703	0.693	0.704	0.691	0.704	0.690	0.704	0.690
$\mathcal{P} \mathcal{S}^{rbf}$	3	0.703	0.699	0.707	0.696	-	-	-	-	-	-	-	-
	7	0.709	0.687	0.716	0.680	0.721	0.677	0.720	0.677	-	-	-	-
	11	0.707	0.686	0.718	0.676	0.723*	0.671	0.722	0.671	0.720	0.672	0.718	0.673
	15	0.704	0.686	0.716	0.675	0.723	0.669	0.723	0.669	0.721	0.669	0.719	0.670
$\mathcal{P} \mathcal{S}^{soe}$	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 [39] uses the default rbf kernel to give $CC = 0.600$ and $rmse = 0.78$.

Table 4: Classification Performance on the Transmembrane Helix Dataset.

(a)

	w	f=1	f=3	f=5	f=7	f=9
		Q_4 ROC	Q_4 ROC	Q_4 ROC	Q_4 ROC	Q_4 ROC
\mathcal{P}^{soe}	5	69.2 0.867	69.7 0.872	70.4 0.878	- -	- -
	7	70.5 0.876	71.1 0.882	71.7 0.887	71.8 0.888	- -
	9	71.4 0.884	71.8 0.888	72.4 0.892	72.7 0.894	72.8** 0.895

(b)

Method	Q_2	Recall	Precision
\mathcal{P}^{soe}	84	81	87
MEMSAT3	83	78	88
TMHMM1	80	68	81
PHDpsihtm08	80	76	83
HMMTOP2	80	69	89
PHDhtm08	78	76	82

The numbers in bold show the best models for a fixed w parameter, as measured by Q_4 accuracy score, and ** denotes the best classification results achieved on this dataset. Results for MEMSAT3 [14] and \mathcal{P}^{soe} were obtained by evaluating it on the TMH static benchmark [19] 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.

The best model is obtained by combining the \mathcal{P} and \mathcal{S} features which gives a raw Q_2 accuracy of 83%. The protein-DNA interaction site program DISIS uses a two-level approach to solve this problem [26]. The first level, which uses SVM learning with profile, predicted secondary structure, and predicted solvent accessibility as inputs, gives $Q_2 = 83\%$ to which our performance compares favorably. DISIS goes on to smooth this initial prediction using a rule-based approach which improves accuracy. We have not yet explored this type of multi-level approach.

Table 7: Classification Performance on the Local Structure Alphabet Dataset.

	$w = f = 5$		$w = f = 7$		$w = f = 9$	
	ROC	Q_{16}	ROC	Q_{16}	ROC	Q_{16}
\mathcal{P}^{rbf}	0.82	64.9	0.81	64.7	0.81	64.2
\mathcal{P}^{soe}	0.83	67.3	0.82	67.7	0.82	67.7
$\mathcal{P} \mathcal{S}^{rbf}$	0.84	66.4	0.84	66.9	0.83	67.2
$\mathcal{P} \mathcal{S}^{soe}$	0.85	68.0	0.84	68.5	0.83	68.9**

$w = f$ gave the best results on testing on few sample points, and hence due to the expensive nature of this problem, we did not test it on a wide set of parameters. ** denotes the best scoring model based on the Q_{16} scores. For this best model the *errsig* rate of 0.21.

5.6 Local Structure Alphabet Performance

We chose to use the Protein Blocks [5] as our target alphabet for local structure prediction. There are sixteen members

in this alphabet which significantly increases prediction difficulty over traditional three-letter secondary structure prediction.

We used a dataset consisting of 1600 proteins derived from the SCOP [23] version 1.57 database, classes a to e, and where no two protein domains have more than 75% sequence identity. This dataset was previously used for predicting of profile-profile scoring functions using neural networks [27]. We computed the local structure alphabets, Protein Blocks [5] using the 3D structure for the proteins.

Due to the high computational requirements associated with such a large training set, we evaluated our *soe* kernels on a wide set of parameters for w and f , but only on a small subset of the 1600 proteins present in the dataset. From this experiment, we observed that prediction was best when $w = f$ and used this to limit the choice of parameters for larger-scale evaluation. Once these promising models were determined, we carried out a 3-way cross validation experiment using all 1600 protein for each parameter set. Table 7 reports the classification accuracy in terms of the Q_{16} accuracy and average of the ROC scores for different members of the Protein Blocks.

From Table 7 we can draw the well-established conclusion of this paper that the *soe* kernel performs marginally better than the *rbf* kernel. The addition of predicted secondary structure information, \mathcal{S} features does improve the Q_{16} performance marginally as was expected for local structure prediction. Our Q_{16} results are very encouraging, since they

Table 5: Relative Solvent Accessibility Class Prediction and Regression Performance.

Cutoff % Method	0%			5%			16%			Regression	
	Q_2	ROC	F1	Q_2	ROC	F1	Q_2	ROC	F1	CC	rmse
$\mathcal{P}^{rbf}, w, f = 3$	87.0	0.845	0.486	79.9	0.855	0.664	78.0	0.855	0.755	0.648	0.211
$\mathcal{P}^{rbf}, w, f = 5$	87.1	0.845	0.491	80.4	0.857	0.670	78.3	0.857	0.758	0.654	0.209
$\mathcal{P}^{rbf}, w, f = 7$	87.1	0.844	0.491	80.2	0.856	0.668	78.4	0.856	0.758	0.653	0.209
$\mathcal{P}^{rbf}, w, f = 9$	86.9	0.843	0.487	80.3	0.855	0.667	78.3	0.855	0.756	0.654*	0.208
$\mathcal{P}^{rbf}, w, f = 11$	87.2	0.843	0.486	80.2	0.855	0.666	78.3	0.854	0.756	0.654	0.208
$\mathcal{P}^{soe}, w, f = 3$	87.5	0.845	0.491	80.2	0.857	0.669	78.5	0.858	0.758	0.641	0.211
$\mathcal{P}^{soe}, w, f = 5$	87.6	0.847	0.494	80.8	0.860	0.671	78.7*	0.861	0.762	0.647	0.209
$\mathcal{P}^{soe}, w, f = 7$	87.7*	0.846	0.491	81.0*	0.859	0.670	78.6	0.861	0.760	0.646	0.210
$\mathcal{P}^{soe}, w, f = 9$	87.7	0.846	0.493	80.9	0.859	0.670	78.5	0.860	0.760	0.648	0.209
$\mathcal{P}^{soe}, w, f = 11$	87.7	0.846	0.494	80.9	0.859	0.670	78.5	0.859	0.760	0.650	0.209
1-stage SVM	86.2	-	-	79.8	-	-	77.8	-	-	-	-

The cutoff % is in terms of relative accessible solvent area and determines which residues are exposed (above the cutoff) and buried (at or below the cutoff). The one-stage SVM is that of Kim and Park [20]. Q_2 measures are reported by these two methods but not ROC or F1 measures. CC and $rmse$ denotes the average correlation coefficient and rmse values. We observed the best classification and regression performance by setting $w = f$. The numbers in bold show the best models for a w, f parameter.

are above 67%, whereas the prediction accuracy for a random predictor would be only 6.25%. Competitive methods for local structure alphabet prediction have reported a Q_{16} accuracy of 40.7% [8]. However, these results cannot be directly compared with our method, as they were obtained on a different train/test dataset. We are in the process of comparing PROSAT’s performance to other datasets and methods [17, 5, 8].

6 Conclusions

In this work we have developed a generic support vector machine based framework for producing predictive models to annotate protein sequences. We have tested our framework, PROSAT, with different sets of features on several annotation problems. We have evaluated multiclass classification and binary classification models for predicting local structure, solvent accessibility, transmembrane helical regions, disorder prediction, and protein-DNA interaction site prediction. We have also tested regression models for residue-wise contact order estimation and solvent accessibility prediction.

Our experimental evaluation showed that, in general, the *soe* kernel achieves better performance than the standard *rbf* kernels across a wide range of problems and datasets, even though for some problems, these improvements are rather small. In addition, our results showed that for some problems, by incorporating local information at different levels of granularity, we were able to achieve better performance when compared to the traditional fine-grain approach. Overall, PROSAT outperformed state-of-the-art, tuned prediction methods for residue-wise contact order, solvent accessibility, transmembrane helices, and local structure alphabet prediction problems. We also show comparable performance on the protein-DNA interaction and disordered prediction problems.

We believe that PROSAT 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.

Acknowledgments

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.

References

- [1] Shandar Ahmad, M. Michael Gromiha, and Akinori Sarai. Analysis and prediction of dna-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*, 20(4):477–486, Mar 2004.
- [2] Shandar Ahmad and Akinori Sarai. Pssm-based prediction of dna binding sites in proteins. *BMC Bioinformatics*, 6:33, 2005.
- [3] S. F. Altschul, L. T. Madden, A. A. Schiffer, 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.
- [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. Dosztanyi, 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. Jakoucheva, and Z. Obradovic. Intrinsic disorder and protein function. *Biochemistry*, 41(21):6573–6582, 2002.
- [8] C. Etchebest, C. Benros, S. Hazout, and A. de Brevern. A structural alphabet for local protein structures: Improved prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 59:810–827, 2005.
- [9] M. Gribskov and N. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computational Chemistry*, 20:25–33, 1996.
- [10] S. Hirose, K. Shimizu, S. Kanai, Y. Kuroda, and T. Noguchi. Poodle-I: a two-level svm prediction system for reliably predicting long disordered regions. *Bioinformatics*, 23(16):2046–2053, 2007.
- [11] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conference on Machine Learning*, 1998.
- [12] T. Joachims. *Advances in Kernel Methods: Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.
- [13] David T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, 292:195–202, 1999.
- [14] David T Jones. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics*, 23(5):538–544, Mar 2007.
- [15] W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.
- [16] 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.

Table 6: Classification Performance on the Protein-DNA Interaction Site Prediction.

	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	0.743	0.452	0.745	0.453	-	-	-	-	-	-	-	-
	7	0.738	0.446	0.743	0.449	0.748*	0.457	0.749	0.462	-	-	-	-
	11	0.712	0.421	0.728	0.438	0.740	0.452	0.743	0.451	0.745	0.452	0.744	0.453
	15	0.685	0.398	0.708	0.415	0.720	0.427	0.726	0.429	0.731	0.437	0.734	0.440
\mathcal{P}^{rbf}	3	0.735	0.441	0.741	0.456	-	-	-	-	-	-	-	-
	7	0.744	0.453	0.745	0.455	0.750	0.461	0.752	0.465	-	-	-	-
	11	0.731	0.443	0.743	0.457	0.749	0.461	0.755	0.464	0.761*	0.470	0.761	0.473
	15	0.708	0.415	0.729	0.440	0.740	0.449	0.745	0.456	0.751	0.462	0.753	0.462
\mathcal{P}^{soe}	3	0.734	0.449	0.741	0.453	-	-	-	-	-	-	-	-
	7	0.743	0.458	0.745	0.459	0.750	0.455	0.752	0.458	-	-	-	-
	11	0.728	0.440	0.743	0.459	0.751	0.463	0.758	0.467	0.763*	0.474	0.762	0.474
	15	0.700	0.411	0.727	0.435	0.740	0.451	0.746	0.459	0.752	0.460	0.755	0.470
$\mathcal{P} \mathcal{S}^{lin}$	3	0.754	0.463	0.756	0.463	-	-	-	-	-	-	-	-
	7	0.745	0.453	0.749	0.460	0.757	0.467	0.758*	0.469	-	-	-	-
	11	0.718	0.428	0.726	0.434	0.737	0.442	0.744	0.447	0.749	0.456	0.748	0.452
	15	0.696	0.403	0.706	0.412	0.719	0.420	0.727	0.428	0.734	0.438	0.739	0.444
$\mathcal{P} \mathcal{S}^{rbf}$	3	0.749	0.463	0.753	0.465	-	-	-	-	-	-	-	-
	7	0.749	0.455	0.750	0.455	0.753	0.460	0.754	0.462	-	-	-	-
	11	0.734	0.445	0.743	0.452	0.748	0.454	0.753	0.459	0.760*	0.464	0.759	0.466
	15	0.711	0.420	0.733	0.445	0.739	0.451	0.744	0.456	0.748	0.461	0.751	0.470
$\mathcal{P} \mathcal{S}^{soe}$	3	0.749	0.467	0.754	0.466	-	-	-	-	-	-	-	-
	7	0.750	0.464	0.749	0.462	0.754	0.464	0.756	0.468	-	-	-	-
	11	0.731	0.441	0.743	0.450	0.752	0.456	0.757	0.464	0.763**	0.470	0.763	0.468
	15	0.700	0.408	0.728	0.438	0.734	0.446	0.745	0.457	0.748	0.461	0.748	0.468

The numbers in bold show the best models for a fixed w parameter, as measured by ROC . \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 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 83.0% with an *errsig* rate of 0.34.

- [17] 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.
- [18] George Karypis. Yasspp: Better kernels and coding schemes lead to improvements in svm-based secondary structure prediction. *Proteins: Structure, Function and Bioinformatics*, 64(3):575–586, 2006.
- [19] Andrew Kernysky and Burkhard Rost. Static benchmarking of membrane helix predictions. *Nucleic Acids Res*, 31(13):3642–3644, Jul 2003.
- [20] Hyunsoo Kim and Haesun Park. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3d local descriptor. *Proteins: Structure, Function, and Bioinformatics*, 54:557–562, 2004.
- [21] A. R. Kinjo, K. Horimoto, and K. Nishikawa. Predicting absolute contact numbers of native protein structure from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 58(1):158–165, 2005.
- [22] 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.
- [23] 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*, 247:536–540, 1995.
- [24] Naderi-Manesh, H. Sadeghi, M. Arab, and M. Movahedi. Prediction of protein surface accessing with information theory. *Proteins: structure, function and genetics*, 42:452–459, 2001.
- [25] M. N. Nguyen and J. C. Rajapakse. Two-stage support vector machines to protein relative solvent accessibility prediction. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 67–72, 2004.
- [26] Yanay Ofra, Venkatesh Mysore, and Burkhard Rost. Prediction of dna-binding residues from sequence. *Bioinformatics*, 23(13):i347–353, 2007.
- [27] T. Ohlson and A. Elofsson. Profnet, a method to derive profile-profile alignment scoring functions that improves the alignments of distantly related proteins. *BMC Bioinformatics*, 6(253), 2005.
- [28] K. W. Plaxco, K. T. Simons, and D. Baker. Contact order, transition state placement and the refolding rates of single domain proteins. *Journal of molecular biology*, 277(4):985–994, 1998.
- [29] 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.
- [30] 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.
- [31] H. Rangwala and G. Karypis. frmsdpred: Predicting local rmsd between structural fragments using sequence information. In *Proceedings of the LSS Computational Systems Biology Conference*, volume 6, pages 311–322, 2007.
- [32] Pedro Romero, Zoran Obradovic, Xiaohong Li, Ethan C. Garner, Celeste J. Brown, and A. Keith Dunker. Sequence complexity of disordered protein. *Proteins: Structure, Function, and Genetics*, 42(1):38–48, 2001.
- [33] B. Rost. Phd: predicting 1d protein structure by profile based neural networks. *Meth. in Enzym.*, 266:525–539, 1996.
- [34] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.
- [35] R. Sanchez and A. Sali. Advances in comparative protein-structure modelling. *Current Opinion in Structural Biology*, 7(2):206–214, 1997.
- [36] T. Schwede, J. Kopp, N. Guex, and M. C. Peitsch. Swiss-model: An automated protein homology-modeling server. *Nucleic Acids Research*, 31(13):3381–3385, 2003.
- [37] K. T. Simons, C. Strauss, and D. Baker. Prospects for ab initio protein structural genomics. *J Mol Biol*, 306(5):1191–1199, Mar 2001.
- [38] A. Smola and B. Scholkopf. A tutorial on support vector regression. *Neuro-COLT2*, NC2-TR-1998-030, 1998.
- [39] J. Song and K. Burrage. Predicting residue-wise contact orders in proteins by support vector regression. *BMC Bioinformatics*, 7(425), 2006.
- [40] Harianto Tjong and Huan-Xiang Zhou. Displar: an accurate method for predicting dna-binding sites on protein surfaces. *Nucl. Acids Res.*, 35(5):1465–1477, 2007.
- [41] V. N. Uversky, J. R. Gillespie, and A. L. Fink. Why are “natively unfolded” proteins unstructured under physiologic conditions? *Proteins: Structure, Function, and Genetics*, 41(3):415–427, 2000.
- [42] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.