Advanced Tree-based Kernels for Protein Classification

Elisa Cilia[1] and Alessandro Moschitti[2]

[1] Department of Information and Communication Technology
University of Trento,
Via Sommarive 14, 38100, Povo (Trento), ITALY
cilia@dit.unitn.it
[2] Department of Computer Science, System and Production
University of Rome, Tor Vergata,
Via Della Ricerca Scientifica s.n.c., 00133, Roma, ITALY
moschitti@info.uniroma2.it

Abstract. One of the aims of modern Bioinformatics is to discover the molecular mechanisms that rule the protein operation. This would allow us to understand the complex processes involved in living systems and possibly correct dysfunctions. The first step in this direction is the identification of the functional sites of proteins.

In this paper, we propose new kernels for the automatic protein active site classification. In particular, we devise innovative attribute-value and tree substructure representations to model biological and spatial information of proteins in Support Vector Machines. We experimented with such models and the Protein Data Bank adequately pre-processed to make explicit the active site information. Our results show that structural kernels used in combination with polynomial kernels can be effectively applied to discriminate an active site from other regions of a protein. Such finding is very important since it firstly shows a successful identification of catalytic sites for a very large family of proteins belonging to a broad class of enzymes.

1 Introduction

Recent research in Bioinformatics has been devoted to the production and understanding of genomic data. The availability of the human genome sequence has shown how small our knowledge about the relation between molecular structures and their functions is. There are about 10,000 genes encoding approximately the same number of well characterized proteins, but apparently, the number of protein functions seems to be higher. Current methods rarely allow us to recognize all the functions that a protein can carry out.

From a biological point of view, it is clear the importance of understanding the molecular mechanisms which rule the correct operation of a protein or, in case of pathologies, which lead to an altered or null protein function. Through the knowledge of these mechanisms it is possible to eventually correct dysfunctions.

Such research is quite complex to carry out as a protein function is the result of a combination of several factors. One important step in this direction is the study of the relation between molecular structures and their functions, which in turn depends on the discovering of the protein active sites. As there is a large number of synthesized proteins which have no associated function yet, automatic approaches for active site detection are critical.

Currently, the general strategy used to identify a protein active site involves the expertise of researchers and biologists accumulated in years of study on the target protein. This manual approach is conducted essentially using homology based strategies, i.e. inferring the function of a new protein based on a close similarity to already annotated proteins [1]. Sometimes proteins with the same overall tertiary structure can have different active sites, i.e. different functions and proteins with different overall tertiary structure can show the same function and similar active sites. In these cases homology based approaches are inadequate. In general, there is no automated approach to protein active site detection, although it is evident its usefulness to restrict the number of candidate sites and also to automatically learn rules characterizing an active site [2].

In this paper we define the problem of determining protein active sites in terms of a classification problem. We modeled protein active site based on both attribute/value and structural representations [3]. The former representation is a set of standard linear features whereas the latter is constituted by tree structures extracted from graphs associated with proteins or their candidate sites. The graph nodes (or vertices) represent amino acids (or better residues) and edges represent distances in the three-dimensional space between these residues.

We applied these representations to SVMs using polynomial kernels, tree kernels and some combinations of them. To experimentally evaluate our approach, we created a data set, using the protein structures retrieved from the *Protein Data Bank* (PDB) [4] maintained by the *Research Collaboratory for Structural Bioinformatics* (RCBS) at http://www.rcbs.org. The combined kernels show the highest F_1 measure, i.e. 68%, in the detection of active sites. This is an important and promising result considering that the baseline based on a random selection of active sites has an upperbound of only 2%.

In the remainder of this article Section 2 describes the faced problem. Section 3 describes the proposed linear and structural features. Section 4 describes the experimental evaluation and reports the results of the classification experiments. Finally, in Section 5, we summarize the results of the previous sections and propose other interesting future research lines.

2 Definition of the Protein Active Site Classification Task

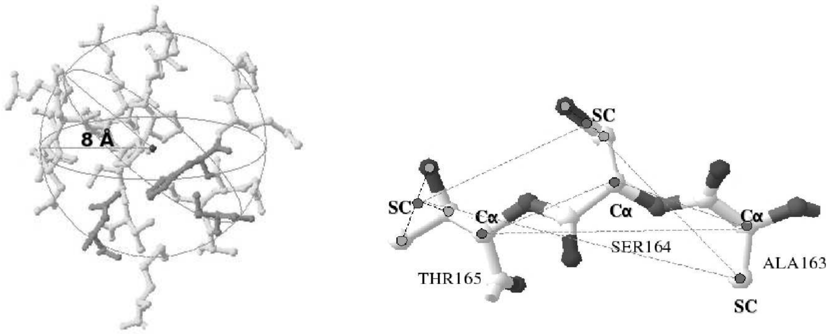
In this section we formally define the task of protein active site classification: first, we give the protein active site definition then we provide a formal description of our computational model.

2.1 Protein Active Site Definition

An active site in a protein is a topological region which defines the protein function, in other words it is a functional domain in the protein three-dimensional structure (see also [2]). In a cell there are many types of proteins which carry out different functions. The enzymes are those proteins able to accelerate chemical processes inside a cell. This type of proteins are distinguished from structural and supplying proteins for their catalytical action on the large part of molecules constituting the living world. We limit our research to a particular class of enzymes, the hydrolases.

Hydrolases are maybe the most studied and known type of enzymes. They catalyze hydrolysis reactions, generically consisting in the cleavage of a biochemical compound thanks to the addition of a water molecule (H_2O). The characteristic of some hydrolases to catalyze reactions in the presence of a water molecule motivates our model: as a hydrolase active site, we choose a sphere in a three-dimensional space centered in the coordinates of the oxygen atom of a water molecule. This sphere includes a portion of the protein within its volume, which is a number of amino acids which could reciprocally interact with other amino acids in the surrounding space, or with water molecules. In this first analysis, we consider a sphere with a radius of 8 Å, which is the maximum distance needed for the water-residue interaction.

Figure 1(a), shows the active site of 1A2O protein structure and its representation according to our model. The protein residues are colored in light gray whereas the particular catalytic residues are in dark gray. The center of the sphere is the black colored oxygen atom of a water molecule.



(a) (b)

Fig. 1. (a) A sphere (positive example). (b) Distances.

2.2 The Computational Model

We defined the functional site identification as a classification problem, where the objects we want to classify are protein active sites. We represent the portion of the protein contained in a spherical three-dimensional region with a completely connected graph. Each vertex of this graph is a residue and each edge represents the distance in the three-dimensional space between a pair of vertices.

Every amino acid is represented by two points in the three-dimensional space: the first represents an amino acid main-chain (the α-carbon atom of the amino acid, Cα) and the second represents an amino acid side-chain (the centroid between the coordinates of the atoms belonging to the amino acid side-chain, SC) (see figure 1(b)). The same kind of approximation has been described in [5] because it seems to provide a good balance between fuzziness and specificity in these kind of applications.

In figure 1(b) the three-dimensional SC-SC distances and Cα-Cα distances are indicated between the represented chain of three residues.

An object (modeled by a graph centered on a water molecule) can be classified as being an active site or not with a binary classifier. Thus, we consider as a positive example, a graph whose set of vertices includes all the catalytic amino acids and as a negative example a graph which contains no catalytic amino acid. Moreover, to reduce the task complexity, we extract, from the initial completely connected graph, some spanning substructures which preserve the edges within the maximum interaction distance of 5 Å between the side-chains of the residues.

The next section shows how the above representation model can be used along with Support Vector Machines to design an automatic active site classifier.

3 Automatic Classification of Active Sites

Previous section has shown that the active site representation is based on graphs. To design the computational model of these latter, we have two possibilities: (1) we extract scalar features able to capture the most important properties of the graph and (2) we can use graph based kernels [6] in kernel-based machines such as Support Vector Machines [7]. Point (2) often leads to high computational complexity. We approached such problem by extracting a tree forest from the target graph and applying efficient tree kernels [8].

3.1 Tree Kernels

Tree-kernel functions are viable alternative to attribute/value representations of tree structures. Such functions implicitly define a feature space based on all possible tree substructures. Given two trees T_1 and T_2, the kernel function evaluates the number of common fragments.

More formally, let $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ be a tree fragment space, the indicator function $I_i(n)$ is equal to 1 if the target f_i is rooted at node n and equal to 0 otherwise. A tree-kernel function over t_1 and t_2 is $K_t(t_1, t_2) = \sum_{n_1 \in N_{t_1}} \sum_{n_2 \in N_{t_2}} \Delta(n_1, n_2)$, where N_{t_1} and N_{t_2} are the sets of the t_1's and t_2's nodes, respectively. In turn $\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \lambda^{l(f_i)} I_i(n_1) I_i(n_2)$, where $0 \leq \lambda \leq 1$ and $l(f_i)$ is the height of the subtree f_i. Thus $\lambda^{l(f_i)}$ assigns a lower weight to larger fragments. When $\lambda = 1$, Δ is equal to the number of common fragments rooted at nodes n_1 and n_2.

The Δ function depends on the type fragments that we consider as a *basic* features. For example, in [9], the SubSet Trees (SSTs) are proposed. These are any portion p of the initial tree T subject to the constraint that for each node $n \in p$ either n has no children or it has all children described in T. The evaluation of Δ computing such fragment space can be carried out in $O(|N_{t_1}| \times |N_{t_2}|)$.

In figure 2 all the subtrees of a given tree are represented while in figure 3 there are some of the fragments in the SST structure space for the same given tree.

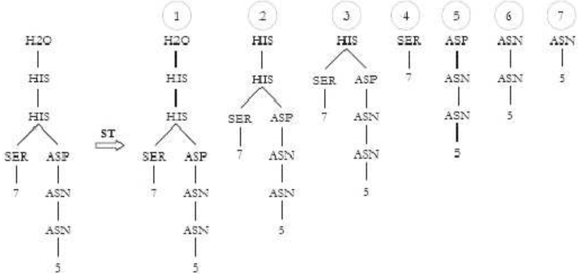


Fig. 2. A tree with its *SubTrees*.

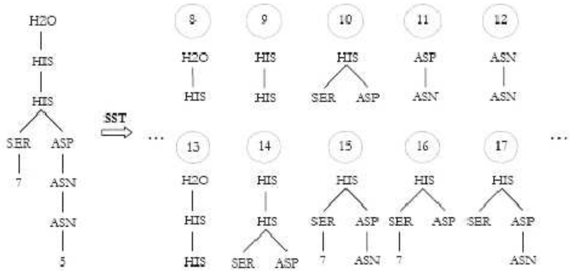


Fig. 3. A tree with some of its *SubSet Trees*.

A more general form of fragments has been given in [8]. In this case any portion p of T, namely Partial Trees (PT), is considered and an efficient evaluation is provided. Note that the SST space is less rich, i.e. it is a subspace of the PT set. Nevertheless, the SST-based kernel can provide higher accuracy since its parameterization is simpler and it usually contains less irrelevant fragments than the PT kernel. In figure 4 some of the partial trees of a given tree are repre-

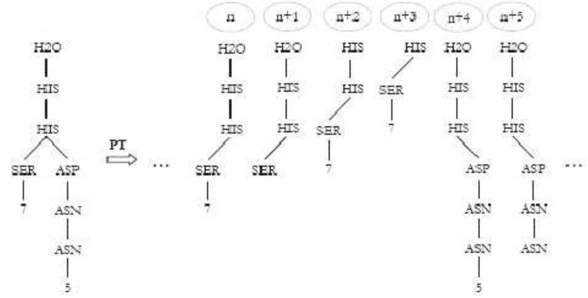


Fig. 4. A tree with some of its *Partial Trees.*

sented. Note that ST $\subset$ SST $\subset$ PT, if we consider ST, SST and PT respectively as the sets of all subtress, all subset trees and all the partial trees of a given tree.

To compute the PT space, we need to define a different Δ function as follows:

- if the node labels of n_1 and n_2 are different then $\Delta(n_1, n_2) = 0$;
- else

$$\Delta(n_1, n_2) = 1 + \sum_{\boldsymbol{J}_1, \boldsymbol{J}_2, l(\boldsymbol{J}_1)=l(\boldsymbol{J}_2)} \prod_{i=1}^{l(\boldsymbol{J}_1)} \Delta(c_{n_1}[\boldsymbol{J}_{1i}], c_{n_2}[\boldsymbol{J}_{2i}]) \qquad (1)$$

where $\boldsymbol{J}_1 = \langle J_{11}, J_{12}, J_{13}, ..\rangle$ and $\boldsymbol{J}_2 = \langle J_{21}, J_{22}, J_{23}, ..\rangle$ are index sequences associated with the ordered child sequences c_{n_1} of n_1 and c_{n_2} of n_2, respectively, $\boldsymbol{J}_{1i}$ and $\boldsymbol{J}_{2i}$ point to the i-th child in the corresponding sequence, and $l(\cdot)$ returns the sequence length.

Furthermore, we add two decay factors: μ for the height of the tree and λ for the length of the child sequences. It follows that

$$\Delta(n_1, n_2) = \mu\Big(\lambda^2 + \sum_{\boldsymbol{J}_1, \boldsymbol{J}_2, l(\boldsymbol{J}_1)=l(\boldsymbol{J}_2)} \lambda^{d(\boldsymbol{J}_1)+d(\boldsymbol{J}_2)} \prod_{i=1}^{l(\boldsymbol{J}_1)} \Delta(c_{n_1}[\boldsymbol{J}_{1i}], c_{n_2}[\boldsymbol{J}_{2i}])\Big)\,, \qquad (2)$$

where $d(\boldsymbol{J}_1) = \boldsymbol{J}_{1l(\boldsymbol{J}_1)} - \boldsymbol{J}_{11}$ and $d(\boldsymbol{J}_2) = \boldsymbol{J}_{2l(\boldsymbol{J}_2)} - \boldsymbol{J}_{21}$. In this way, we penalize both larger trees and subtrees built on child subsequences that contain gaps. Moreover, to have a similarity score between 0 and 1, we also apply a normalization in the kernel space, $K'(T_1, T_2) = \frac{K(T_1,T_2)}{\sqrt{K(T_1,T_1) \times K(T_2,T_2)}}$.

Equation 2 is a more general one, the kernel can be applied to PTs. Also note that if we only consider the contribution of the longest child sequence from node pairs that have the same children, we implement the SST kernel. For the ST computation we also need to remove the λ^2 term from Eq. 2.

3.2 Scalar Features

Scalar features refer to typical chemical values of the molecules described in the target graph. We defined 5 different types of such features (see Table 1):

The first class (C1) encodes chemical and physical properties of the graph. This class represents properties such as hydrophobicity, polarity, polarizability and Van der Waals volume of the amino acids composing the sphere. The encoding is the same used in [10] where the features were used to classify the function of proteins.

The second class (C2) encodes the amino acid composition of a spherical region. There is a feature associated with every labeled vertex (amino acid) in the graph, weighted with the inverse of the distance from the oxygen atom of the water molecule which is the center of the sphere. This group of features emphasizes the importance of the interaction distance of a residue with respect to a water molecule.

The third class (C3) represents charge or neutrality of a spherical region. This is measured by counting the number of positively or negatively charged amino acids.

Another group of linear features (C5) encodes the quantity of water in a sphere. This is measured by counting the number of water molecules within the sphere radius. This group of features is motivated by the fact that biologists observed that an active site is usually located in a hydrophobic core of the protein while on the surface the quantity of water is higher and the residues exposed to the solvent are not hydrophobic.

Finally, the last class of linear features (C6) is the one which measures the atomic density of the sphere calculated as the total number of atoms in the sphere.

Linear Features	**Description**
1st Class	Physical and chemical properties (amino acid attributes)
2nd Class	Amino acidic Composition
3rd Class	Charge/Neutrality
5th Class	Water molecule quantity
6th Class	Atomic density
Structural Features	**Description**
4th Class	Tree substructures from tertiary structure

Table 1. Representation: feature classes

It should be noted that (a) the last two classes of linear features are made discrete using a different number of value intervals. A feature is associated with an example if the measured value of a certain property falls in the correspondent range. (b) These features are often used to describe protein structures in similar tasks of Bioinformatics [10] and to develop software for protein structure prediction like Modeler 7v7.

3.3 Structural Features

We designed a class of structural features to encode the three-dimensional structure (tertiary structure) or better, the spatial configuration characterizing a spherical region, i.e. the set of amino acids composing it with their 3D distances. As previously mentioned this representation results in a completely connected graph since every vertex is connected to any other vertex in the sphere graph through an edge labeled with the 3D distance of the pair.

Starting from this completely connected graph, we extract some tree substructures using heuristics: we fix the maximum interaction distance to 5 Å between the side-chains of the residues and we use the minimum spanning tree algorithm to extract tree structures from the graph.

Such heuristics are motivated by the observation that to perform the catalytic function it is necessary that the side-chains of the catalytic residues can interact with each other and with the substrate. The maximum interaction distance between atoms in different residue side-chains is usually of about 3-4 Å. We chose a cut-off distance of 5 Å to take into consideration our approximation in the representation of residues (figure 1(b)).

The applied cut-off possibly leads to the separation in disconnected components of the initial graph. From each of these components, using the Prim algorithm [11], we extract the spanning tree which minimizes the interaction distances d_{xy} between the side-chains of the residues x and y. Note that as some graphs contain more than one connected component, the Prim algorithm is applied to each of them, therefore the final output is a tree forest.

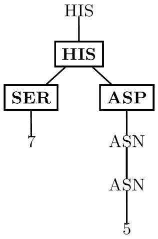


Fig. 5. Graphical representation of a tree of a sphere.

We add the water molecule (center of the sphere) as root node to the obtained spanning tree. In figure 5, we show a tree which can represent the spherical region in figure 1(a). In bold within the boxes, we highlight the nodes which represent catalytic amino acids.

The tree substructures generated for each example constitute the features analyzed by our tree kernel function. If two examples are described by two tree forests, we can use as a kernel function the summation of the tree kernels applied to all possible pairs coming from such forests.

4 Experiments

In the next subsections, we describe our classification experiments carried out on the data set that we generated from the Protein Data Bank.

(a)

Linear	Precision	Recall	F_1
C1	5.5%	66.7%	**10.2%**
C2	55.9%	63.3%	**59.4%**
C3	20%	3.3%	**5.7%**
C5	2.2%	30%	**4.1%**
C6	5.5%	13.3%	**7.8%**

(b)

	Precision	Recall	F_1 $\pm Std.Dev.$
L	62.3%	55.4%	**56.2%** ±6.8
SST_F	66.2%	31.8%	**39.9%** ±13.7
L+SST_F	82.9%	58.6%	**68.3%** ±14.5

Table 2. (a) Linear feature performance. (b) Combined kernel performance.

4.1 Experimental Set-up

The evaluations were carried out using the SVM-light-TK software [12] (available at http://ai-nlp.info.uniroma2.it/moschitti/) which encodes tree kernels in SVM-light [13]. We applied polynomial kernels to the linear features and tree kernels to the structural features.

More precisely, we used the SST and the PT kernels described in [8] on a simple tree, i.e. the main tertiary structure[3], or on a tree forest (see Section 3.3). The former kernels are indicated with SST_T and PT_T whereas the latter are called SST_F and PT_F. The kernel for tree forest is simply the summation of all possible pairs of trees contained in two examples.

We experimented with our models and the protein structures downloaded from the Protein Data Bank (PDB). We adequately pre-processed PDB files to obtain all the information of interest for this task. In particular, we created a data set of 14,688 examples from 48 hydrolases from the PDB structures. The data set is composed of 171 positive examples and 14,571 negative examples, which means a $\frac{1}{125}$ ratio between positive and negative examples.

The results were evaluated by applying a 5-fold cross validation[4] on this data set measuring the performance with the F_1 measure[5]. A noticeable attention was devoted to parameterization (cost factor, decay factor, etc.)

4.2 Experiment Results

Table 1(a) reports the results on the 5 types of linear features using the polynomial kernel (degree 3). These results are only indicative as we did not run a cross validation procedure. We note that most linear features cannot discriminate between active and non-active site. Only, the second class, which encodes the structural information, shows a meaningful F_1. The general low results of linear features is caused by the remarkable complexity of the task as suggested by the F_1 upperbound of the random selection, i.e. $\simeq 1.6\%$.

In order to boost the classification performance, we experimented with the structural kernels. Table 1(b) summarizes the cross validation results: Row 2

[3] The most relevant tree structure in the forest, that is, the tree which contains at least a catalytic amino acid and the two nearest residue side-chains of the sphere

[4] We separated the data set into five parts, each one composed of examples belonging to a set of nine or ten protein structures randomly assigned to this set.

[5] F_1 assigns equal importance to Precision P and Recall R i.e. $F_1 = \frac{2P \cdot R}{P+R}$

L+TK	**Precision**	**Recall**	F_1 $\pm Std.Dev.$
SST_F	82.9%	58.6%	**68.3%** ±14.5
SST_T	79.7%	51.7%	**62.3%** ±10.4
PT_T	80.4%	41.2%	**54.4%** ±9.1

Table 3. Tree kernel impact

reports the outcome with polynomial kernels on all the linear features (L), Row 3 shows the outcomes of the SST kernels on the tree forest (SST_F) and Row 4 illustrates the performance of the polynomial kernel summed to the SST kernel on the tree forest (L+SST_F). The ± sign precedes the standard deviation evaluated on the 5 folds.

It is worth to note that the F_1 obtained with the linear features (56.24%) improves by 12 absolute points if we use the combined model (L+SST_F), i.e. 68%.

We also experimented with the PT kernel. The results of the cross validation experiments are summarized in Table 3: Row 2 reports the results with polynomial kernel plus SST_F (applied to linear features and a forest structure), Row 3 reports the cross validation results of polynomial kernel plus SST_T (applied to linear features and a tree structure) and finally Row 4 illustrates the performance of the additive combination of polynomial with the PT kernel (PT_T) (on linear features and a tree structure).

The results show that the highest F_1 measure can be achieved with the SST_F but quite similar performance can be obtained representing examples with only a tree structure in the forest, i.e. SST_T.

In contrast to our expectations the PT kernel, which may be considered the most appropriate for this task, shows the lowest F_1. A plausible explanation of such low F_1 is the higher complex parameterization needed by the PT kernel.

Overall, the very good F_1 of our best model suggests that our classification system can be a useful tool to help biology researchers in the study of protein functions.

5 Related Work

Literature work demonstrates the importance of protein structures and theirs active sites for the discovering of protein functions. Hereafter we report some of the related work.

In [3], novel Recurrent Neural Networks for the detection of Protein secondary structures are proposed. These models exploit non-casual bidirectional dynamics to capture up-stream and down-stream information. In [5] we can see the first general approach to structure searches based on active site rather than exclusively on fold similarity. The authors showed that three-dimensional signature consisting of only a few functionally important residues can be diagnostic of membership in superfamily of enzymes which can represent a first step in the inference of some functional properties. This membership, results not only from fold similarity but also from the disposition of residues involved in a conserved function. It could be interesting to get a potential active site of a protein starting from its structure rather than get only an active site template from a known protein as the authors do in such work. In [14] a kernel-based approach is used to predict signal peptides and their cleavage sites from the primary sequence of proteins. In [15] the task of catalytic residue prediction is solved using 26 different classifiers from WEKA package. The authors show that the best performing algorithm is a Support Vector Machine. In this work structural properties are captured with linear features.

Many other approaches have been employed in different application areas of bioinformatics (functional genomics, protein bioinformatics, etc.) with success. Learning methods have been applied to face other biological tasks such as translation initiation site recognition in DNA genes [16], promoter region-based classification of genes [17], protein classification, protein-protein interaction [18], functional classification from microarray expression data [19], [20], [21], [22]. In such work SVMs and kernel methods (see [6] for a survey of kernels for structured data) are used as the main learning algorithm.

6 Conclusions

In this paper, we have studied the problem of the identification of protein functional sites. We have defined a novel computational representation based on biological and spatial considerations and several classes of linear and structural features.

The experiments with SVMs using polynomial and tree kernels and their combinations show that the highest F_1, i.e. 68%, is achieved with the combined model. Such finding is very important since it firstly shows the successful identification of catalytic sites of a very large family of catalytic proteins belonging to a broad class of enzymes. Moreover, our work highlights the importance of structural information in the detection of protein active sites. This result motivates the need of structural representations which we efficiently modeled by means of tree kernels.

Acknowledgements

The authors thank Dr. Sergio Ammendola for the valuable support regarding to the biological aspects of this work.

References

1. Cilia, E., Fabbri, A., Uriani, M., Scialdone, G.G., Ammendola, S.: The signature amidase from sulfolobus solfataricus belongs to the cx3c subgroup of enzymes cleaving both amides and nitriles: Ser195 and cys145 are predicted to be the active sites nucleophiles. The FEBS Journal **272** (2005) 4716–4724

2. Tramontano, A.: The ten most wanted solutions in Protein Bioinformatics. Chapman & Hall/CRC Mathematical Biology and Medicine Series (2005)
3. Brunak, S., Baldi, P., Frasconi, P., Pollastri, G., Soda, G.: Exploiting the past and the future in protein secondary structure prediction. Bioinformatics **15**(11) (1999)
4. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Res **28**(1) (2000) 235–242
5. Meng, E.C., Polacco, B.J., Babbitt, P.C.: Superfamily active site templates. PROTEINS: Structure, Function, and Bioinformatics **55** (2004) 962–976
6. Gärtner, T.: A survey of kernels for structured data. Multi Relational Data Mining (MRDM) **5** (2003) 49–58
7. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
8. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Proceedings of The 17th European Conference on Machine Learning, Berlin, Germany, 2006, Berlin, Germany (2006)
9. Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In: ACL02. (2002)
10. Borgwardt, K.: Graph-based Functional Classification of Proteins using Kernel Methods. Ludwig Maximilians University of Monaco (2004)
11. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. Journal **36** (1957) 1389–1401
12. Moschitti, A.: A study on convolution kernel for shallow semantic parsing. In: Proceedings of the 42th Conference on Association for Computational Linguistic (ACL-2004), Barcelona, Spain (2004)
13. Joachims, T.: Making large-scale svm learning practical. In: Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges and A. Smola editors (1999)
14. Yang, Z.R.: Orthogonal kernels machines for the prediction of functional sites in proteins. IEEE Trans. on Systems, Man and Cybernetics **35**(1) (2005) 100–106
15. Petrova, N.V., Wu, C.H.: Prediction of catalytic residues using support vector machine with selected protein sequence and structural properties. BMC Bionformatics 2006 (7) (2006) 312–324
16. Zien, A., Rätsch, G., S. Mika, B.S., Lengauer, T., Muller, K.: Engineering support vector machine kernel that recognize translation initiation sites. Bioinformatics **16**(9) (2000) 799–807
17. Pavlidis, P., Furey, T.S., Liberto, M., Grundy, W.N.: Promoter region-based classification of genes. In: Proceedings of the Pacific Symposium on Biocomputing. (2001) 151–163
18. Bock, J.R., Gough, D.A.: Predicting protein-protein interactions from primary structure. Bionformatics **17**(5) (2001) 455–460
19. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C., Ares, M., Haussler, D.: Support vector machine classification of microarray expression data. In: UCSC-URL. (1999)
20. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C., Furey, T.S., Ares, M., Haussler, D.: Knowledge-based analysis of microarray gene expression data by using support vector machine. PNAS **97**(1) (2000) 262–267
21. Furey, T.S., Duffy, N., Cristianini, N., Bednarski, D., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics **16**(10) (2000) 906–914
22. Pavlidis, P., Furey, T., Liberto, M., Grundy, W.N.: Learning gene functional classification from multiple data types. Journal of Computational Biology (2002)