

# Graphlet Kernels for Prediction of Functional Residues in Protein Structures

VLADIMIR VACIC,<sup>1</sup> LILIA M. IAKOUCHEVA,<sup>2</sup> STEFANO LONARDI,<sup>1</sup>  
AND PREDRAG RADIVOJAC<sup>3</sup>

## ABSTRACT

We introduce a novel graph-based kernel method for annotating functional residues in protein structures. A structure is first modeled as a protein contact graph, where nodes correspond to residues and edges connect spatially neighboring residues. Each vertex in the graph is then represented as a vector of counts of labeled non-isomorphic subgraphs (graphlets), centered on the vertex of interest. A similarity measure between two vertices is expressed as the inner product of their respective count vectors and is used in a supervised learning framework to classify protein residues. We evaluated our method on two function prediction problems: identification of catalytic residues in proteins, which is a well-studied problem suitable for benchmarking, and a much less explored problem of predicting phosphorylation sites in protein structures. The performance of the graphlet kernel approach was then compared against two alternative methods, a sequence-based predictor and our implementation of the FEATURE framework. On both tasks, the graphlet kernel performed favorably; however, the margin of difference was considerably higher on the problem of phosphorylation site prediction. While there is data that phosphorylation sites are preferentially positioned in intrinsically disordered regions, we provide evidence that for the sites that are located in structured regions, neither the surface accessibility alone nor the averaged measures calculated from the residue microenvironments utilized by FEATURE were sufficient to achieve high accuracy. The key benefit of the graphlet representation is its ability to capture neighborhood similarities in protein structures via enumerating the patterns of local connectivity in the corresponding labeled graphs.

**Key words:** algorithms, graphs, kernel methods, machine learning, protein structure, protein function.

## 1. INTRODUCTION

WITH OVER 50,000 STRUCTURES deposited in the Protein Data Bank (PDB) (Berman et al., 2000) and high-throughput efforts under way (Burley et al., 1999), functional characterization of proteins with

---

<sup>1</sup>Department of Computer Science and Engineering, University of California, Riverside, California.

<sup>2</sup>Laboratory of Statistical Genetics, The Rockefeller University, New York, New York.

<sup>3</sup>School of Informatics and Computing, Indiana University, Bloomington, Indiana.

known three-dimensional (3D) structure is gaining importance in the global effort to understand structure-to-function determinants (Laskowski and Thornton, 2008). Experimental assays for functional characterization are expensive and time-consuming; thus, the development of accurate computational approaches for function prediction is essential to the functional annotation process (Lee et al., 2007; Watson et al., 2005). Typically, the problem of protein function prediction reduces to one or more of the following questions: (1) prediction of the molecular and biological function of the molecule, (2) prediction of ligands, cofactors, or macromolecular interaction partners, and (3) prediction of the residues involved in or essential for function, for example, interface sites, hot spots, metal binding sites, catalytic sites, or post-translationally modified residues (Rost et al., 2003). At a higher level, computational methods can be used to establish connections between proteins and disease, typically via simulating protein folding pathways (Dobson, 2001) or by using statistical inference techniques to predict gene-disease associations (Dalkilic et al., 2008) or the effects of mutations (Mooney, 2005).

Prediction of protein function from 3D structure emerged in the late 1980s and early 1990s when the accumulation of solved structures in PDB made systematic studies feasible. There are four basic approaches used in this field, starting from residue-level function and building toward higher level annotation: (1) residue microenvironment-based methods, (2) template-based methods, (3) docking-based methods, and (4) graph-theoretic approaches. In addition to these bottom-up strategies, another group of methods tackle the problem top-down to directly predict protein function on a whole-molecule level, without necessarily finding functional residues, and then investigate the residues most critical in the classification process.

In residue microenvironment-based approaches, one defines a neighborhood around a residue of interest and counts the occurrences of different atoms, residues, groups of residues, or derived/predicted residue properties within this neighborhood. Zvelebil and Sternberg (1988) used spherical neighborhoods to distinguish between metal binding and catalytic residues, while Gregory et al. (1993) and Bagley and Altman (1995) used concentric spheres to generate a score (Gregory et al., 1993) or create a set of features (Bagley and Altman, 1995) that can be used to predict various functional properties. The residue microenvironment strategy has also been extended to the unsupervised framework, for instance, to gain insights into structural conservation and its relationship with function (Mooney et al., 2005) and has been combined with localized molecular dynamics simulations to predict function (Glazer et al., 2008). Template-based methods, introduced in the 1990s (Fetrow and Skolnick, 1998; Kleywegt, 1999; Russell, 1998; Wallace et al., 1996), encode spatial relationships between residues known or assumed to be functionally important in order to scan query protein structures for the existence of similar patterns. A classical example of a template is the catalytic triad in serine proteases, where Ser, His, and Asp residues are required to occur on the surface of the protein, within a predefined set of distances of one another. Another template-based strategy, adopted from computer vision, is geometric hashing, in which a database of atom or residue patterns is searched for similarities with the new structures (Nussinov and Wolfson, 1991; Wallace et al., 1997; Wolfson and Rigoutsos, 1997). Recently, strategies based on small molecule docking to a protein structure have also been used, where identification of a common ligand, preferably in its high-energy state, may indicate similar molecular function of the protein substrates, for example, catalysis of the same reaction (Hermann et al., 2007; Song et al., 2007). Finally, graph-theoretic approaches have been proposed in the context of computational chemistry and data mining. The idea common to all these approaches is to transform protein structures into graphs, where vertices encode residues, atoms, or secondary structure elements, and edges reflect proximity or physicochemical interactions. In one of the earliest approaches, protein structures were scanned for isomorphic subgraphs (Artymiuk et al., 1994; Grindley et al., 1993). Other authors addressed the problem via the framework of frequent subgraph mining (Bandyopadhyay et al., 2006; Huan et al., 2005; Wangikar et al., 2003), typically starting with a set of proteins known to have the same or similar function. Several residue-level functional predictors have been implemented as public web services dedicated to predicting both functionally important residues as well as the global function of the protein (Laskowski, et al., 2005a,b; Liang et al., 2003; Pal and Eisenberg, 2005).

Methods that predict function directly at the whole-molecule level (Borgwardt et al., 2005; Pazos and Sternberg, 2004) can in principle be combined with approaches that identify functional residues in the general sense (Elcock, 2001; Glaser et al., 2006; Kalinina et al., 2004; Ondrechen et al., 2001; Reva et al., 2007) to achieve similar results. Finally, we note that structural-alignment algorithms—for example, DALI (Holm and Sander, 1993)—can also be used to carry out function prediction. However, a small number of protein folds (~1000) compared to the large number of protein functions (~6000 leaf nodes for molecular function or biological process in the Gene Ontology) combined with the fact that different protein folds can

be associated with identical or similar functions limit the usability of structural alignment algorithms for this task.

In this study, we introduce a method, referred to as the *graphlet kernel*, for identifying functional sites in protein structures. We first represent a protein structure as a contact graph where nodes are residues and edges connect vertices that correspond to the neighboring residues in space. The method then combines the graphlet representation of every vertex in a graph (Przulj et al., 2004) and kernel-based statistical inference (Scholkopf et al., 2004). We extend the concept of graphlets to labeled graphlets and use the counts of labeled graphlets to compute a kernel function as a measure of similarity between the vertices. We show that the graphlet kernel generalizes some previous methods such as FEATURE (Bagley and Altman, 1995; Wei and Altman, 1998) and S-BLEST (Mooney et al., 2005), and can also be readily extended to other problems involving graphs, in either a supervised or an unsupervised learning scenario. Finally, we provide evidence that the performance of this algorithm compares favorably to standard sequence and structure-based methods in the tasks of predicting phosphorylation sites and catalytic residues from protein structures.

## 2. METHODS

The problem addressed here can be generally defined as follows: given a protein structure, probabilistically assign function to each amino acid. Functional assignments are based on similarities of the structural neighborhoods of residues under consideration and measured in terms of local patterns of inter-residue connectivity. We start by modeling a protein structure as a *protein contact graph*, where each amino acid is represented by a vertex in the graph and two vertices are connected by an undirected edge if the corresponding amino acids are closer than some predetermined distance. We then introduce the *graphlet kernel*, an efficient method for computing similarities of vertex neighborhoods, and show how a maximum margin classifier (e.g., support vector machine [SVM]) can be used for binary classification of vertices.

More formally, let us assume that the input data to our algorithm consists of a set of protein structures, represented by a single disconnected graph  $G = (V, E)$ , where  $V$  is the set of labeled vertices and  $E \subset V \times V$  is the set of edges. We consider two labeling functions  $f: V \rightarrow \mathcal{A}$ , where  $\mathcal{A}$  is a finite alphabet, and  $g: V \rightarrow \{+1, -1\}$ , where  $g(v) = +1$  indicates that the residue is functional and  $g(v) = -1$  indicates that the residue is not functional or that the functional information is unknown. The task of the kernel-based classifier is to assign a posterior probability of positive class to every vertex of an unseen protein contact graph.

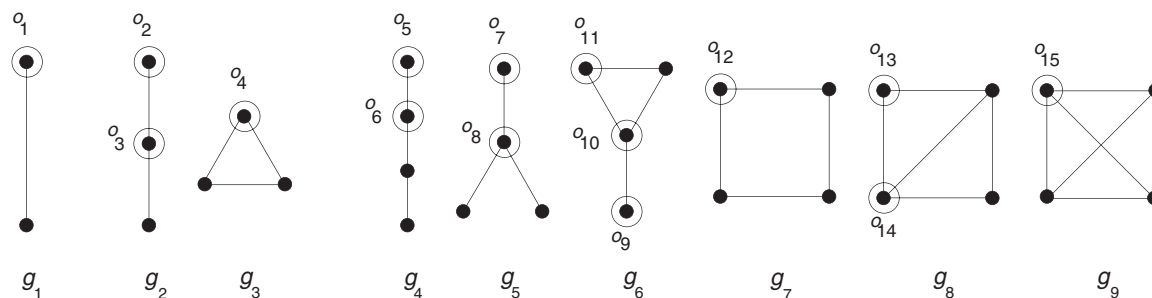
In the remainder of this section, we briefly review the notion of *graphlets* and *automorphism orbits* (Przulj, 2007; Przulj et al., 2004). We extend the concept of automorphism orbits to labeled graphs, present an efficient method for enumerating labeled automorphism orbits, and introduce a novel topological similarity measure that is based on the counts of labeled automorphism orbits. We show how a kernel function is computed by comparing the local graph neighborhoods between pairs of vertices.

### 2.1. Graphlets and automorphism orbits

Graphlets are small non-isomorphic connected subgraphs (Przulj, 2007; Przulj et al., 2004) that can be used to capture local graph or network topology. Because a graph can be thought of as being composed of a collection of interdependent graphlets, the counts of graphlets (up to a given size) provide a characterization of the graph properties in a constructive, bottom-up fashion. We refer to a graphlet with  $k$  vertices as a  $k$ -graphlet. Figure 1 illustrates graphlets of size up to 4.

The graph-theoretic concept of automorphism (of graphlets) allows one to explicitly model relationships between a graphlet and its component vertices. For example, in the case of graphlet  $g_2$ , the vertex of interest may be at the periphery or in the center of the graph (Fig. 1). Different positions of this pivot vertex with respect to the graphlet correspond to automorphism orbits, or *orbits* for short. Accordingly, the two orbits corresponding to graphlet  $g_2$  are labeled as  $o_2$  and  $o_3$  (Fig. 1). In the following sections, we show an efficient way to enumerate the orbits which surround the vertex of interest. For a more formal treatment of graphlets and automorphism orbits, we refer the reader to a study by Przulj (2007).

We extend the concepts of graphlets/orbits to labeled graphlets/orbits by associating each vertex in a graph with a symbol from a finite alphabet  $\mathcal{A}$ . In the case of protein contact graphs, these labels can



**FIG. 1.** Nine types of 2-, 3-, and 4-graphlets and the corresponding 15 automorphism orbits. Graphlet  $g_0$  and orbit  $o_0$  represent a single vertex in a graph and are not shown.

represent either the amino acids or one of the reduced alphabets incorporating information on various physicochemical properties of amino acids. The alphabet can also be an extended set of amino acids where higher level residue properties (e.g., secondary structure assignment) are incorporated. In an alternative version of the protein contact graph, vertices may correspond to the elements of secondary structure (Borgwardt et al., 2005), or if a contact graph is constructed on the atom level, nodes may be labeled with the symbols of chemical elements (Ralaivola et al., 2005).

## 2.2. Combinatorial enumeration of graphlets and orbits

We limit further discussion to graphlets and orbits with up to four vertices. For protein contact graphs constructed using the most common parameters, this level of detail is likely to be sufficient, because short characteristic paths follow from the small world properties of such networks (Atilgan et al., 2004). It is not difficult to extend this approach to graphlets of sizes five and above to be used in the analysis of protein-protein interaction networks, for example. The computational cost involved in counting, however, can become prohibitive as the number of different graphlets grows exponentially with the number of vertices.

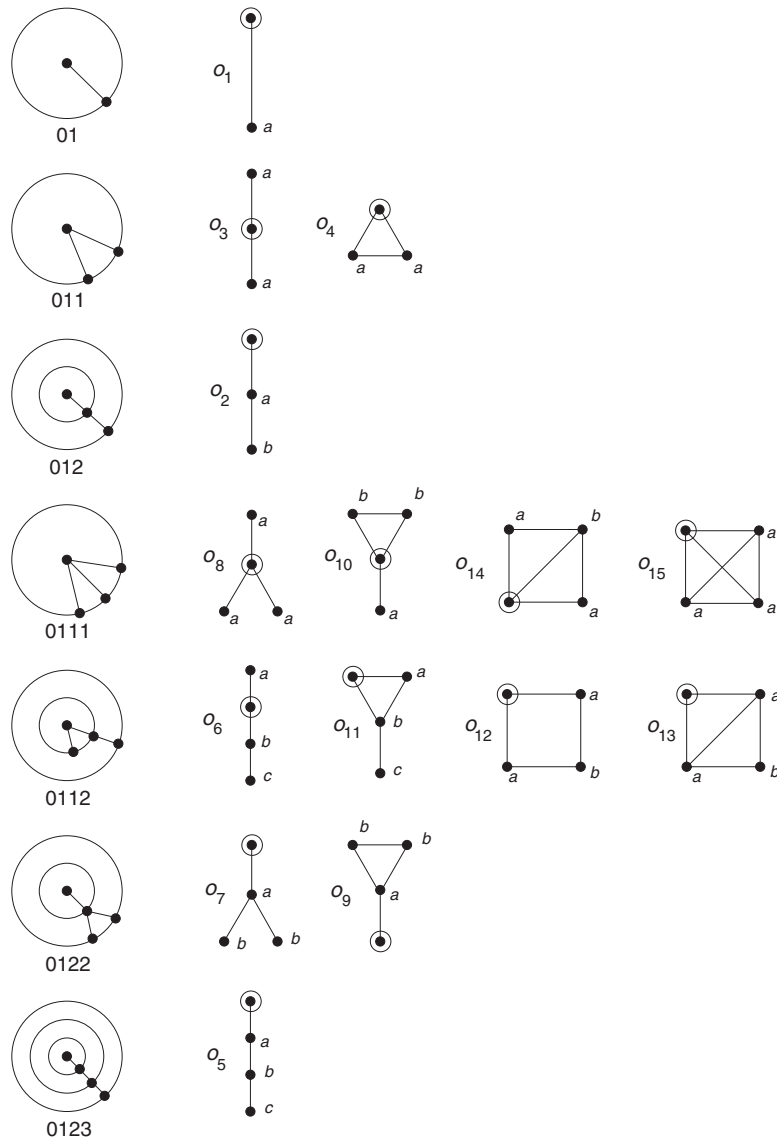
We start by computing the shortest-path distances between a vertex in  $V$  to the remaining vertices using breadth-first search. Given that we are only interested in graphlets of size up to 4, we can terminate the search after the third level. The resulting subgraph is then used to count labeled orbits.

**2.2.1. Counting 1-graphlets and 2-graphlets.** Counting 1-graphlets and 2-graphlets is straightforward. There is exactly one 1-graphlet per vertex in a graph. To count 2-graphlets, it suffices to examine the adjacency list of the pivot vertex  $p$ . Using the distances of vertices from the pivot as the naming convention, we name this case 01 (for the schematic representation of the counting algorithm, see Fig. 2). There is a total of  $\text{deg}(p)$  type  $g_1$  graphlets, that is, orbits  $o_1$ , where  $\text{deg}(\cdot)$  denotes the degree of a vertex.

**2.2.2. Counting 3-graphlets.** There are two cases for counting 3-graphlets: 011, where both non-pivot vertices are at distance 1 from the center, and 012, when one vertex is at distance 1 and the other is at distance 2 (Fig. 2). Case 011 yields 3-graphlets with orbits  $o_3$  or  $o_4$ , but in order to determine the exact orbit type, we need to determine whether there is an edge between the two vertices at distance 1. Case 012 yields only  $o_2$  orbits. Here we do not have to perform the additional edge check because distance 2 implies that there is no edge directly connecting the vertex with the center.

**2.2.3. Counting 4-graphlets.** There are four cases for counting 4-graphlets, namely 0111, 0112, 0122, and 0123 (Fig. 2). Case 0111 yields orbits  $o_8$ ,  $o_{10}$ ,  $o_{14}$ , or  $o_{15}$ , and requires to check connectivity between level-1 neighbors of the pivot vertex. Case 0112 yields  $o_6$ ,  $o_{11}$ ,  $o_{12}$ , or  $o_{13}$  orbits; case 0122 yields  $o_7$  or  $o_9$  orbits; and case 0123 yields  $o_5$  orbits. Similarly to the case 012 in the 3-graphlet counting, the edge checks between the center and vertices with distance 2 or 3 are not necessary.

Assuming that counts of labeled orbits are kept in a hash table that allows expected constant time access to the elements, the counting algorithm runs in  $O(|E_p|) + O(d^4)$  time, where  $|E_p|$  is the number of edges within the level-3 neighborhood of the pivot vertex  $p$  and  $d$  is the maximum degree of a vertex in that neighborhood. The first term in the sum is related to the breadth-first search, whereas the second reflects the



**FIG. 2.** Schematic of the orbit counting algorithm: case 01, orbit  $o_1$ ; case 011, orbits  $o_3$  and  $o_4$ ; case 012, orbit  $o_2$ ; case 0111, orbits  $o_8$ ,  $o_{10}$ ,  $o_{14}$ , and  $o_{15}$ ; case 0112, orbits  $o_6$ ,  $o_{11}$ ,  $o_{12}$ , and  $o_{13}$ ; case 0122, orbits  $o_7$  and  $o_9$ ; and case 0123, orbit  $o_5$ .

cost of counting over all cases, assuming that one can check the existence of edges in  $O(d)$  time using a space-efficient adjacency list representation. This time complexity analysis does not include the time needed to convert a protein structure into the contact graph.

The description of the algorithm has so far ignored vertex labels. In Figure 2, we demonstrate the relationship between the pivot of the graphlet and the remaining vertices. For example, in the 0111 case for orbits  $o_8$  and  $o_{15}$ , the positions of all three non-pivot points are symmetric with respect to the pivot. Hence, when we assign the label to orbits  $o_8$  or  $o_{15}$ , we lexicographically sort the labels of individual residues for consistency. In contrast, in the 0111 case for orbits  $o_{10}$  and  $o_{14}$ , there is a topological difference between vertices marked with  $a$  and  $b$ , but there is no difference between any two vertices each marked with  $a$  or  $b$ . In this case, when we assign the label to orbits  $o_{10}$  or  $o_{14}$ , we first sort the vertices according to their position with respect to the pivot and then lexicographically sort the vertices with the same position based on their labels. This labeling scheme guarantees that a group of vertices will always be labeled in a consistent way without introducing counting artifacts.

### 2.3. The graphlet kernel

We characterize graph vertices in terms of their local neighborhoods in the labeled contact graph. Specifically, for each vertex  $x \in V$ , we look at the distributions of labeled orbits where  $x$  is the pivot node. Given two vertices,  $x$  and  $y$  in the protein contact graph, we define the *kernel function*  $K$  as the following inner product:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

where  $\Phi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_m(x))$  and  $\Phi(y) = (\varphi_1(y), \varphi_2(y), \dots, \varphi_m(y))$  are vectors of counts of labeled orbits. Here,  $\varphi_i(x)$  denotes the number of times labeled orbit  $o_i$  occurs in the graphlet expansion of node  $x$ . Function  $K(x, y)$  is defined over all pairs of vertices  $x$  and  $y$ , and forms a symmetric and positive semi-definite kernel matrix  $K$ , because each element of  $K$  is an inner product of vectors of counts (Haussler, 1999). In addition to the kernel  $K(x, y)$ , we also consider the *normalized kernel*  $K'$  defined as

$$K'(x, y) = K(x, y) / \sqrt{K(x, x) - K(y, y)}.$$

### 2.4. Dimensionality of graphlet representation

Before addressing the computation of the kernel matrix, it is of interest to analyze the dimensionality of the count vector  $\Phi(x)$ . Clearly, the number of labeled orbits  $o_0$  is  $|\mathcal{A}|$  and the number of labeled orbits  $o_1$  is  $|\mathcal{A}|^2$ . Similarly, the number of orbits  $o_2$  equals  $|\mathcal{A}|^3$ , and the number of orbits  $o_6, o_{11}$ , and  $o_5$  equals  $|\mathcal{A}|^4$ . A characteristic of orbits  $o_0, o_1, o_2, o_5, o_6$ , and  $o_{11}$  is that there is no symmetry with respect to the pivot, and results in counts equal to the powers of  $|\mathcal{A}|$  during enumeration of all orbits. The remaining cases, on the other hand, are required to separately address every group of symmetric vertices. These vertices are labeled by the same letter ( $a, b$ , or  $c$ ) for the graphlets in Figure 2. Consider now orbits  $o_3$  and  $o_4$ . There are  $|\mathcal{A}|$  possibilities for the pivot position, while the number of possibilities for the two vertices labeled as  $a$  must begin by grouping of all  $|\mathcal{A}|^2$  cases based on the lexicographically sorted vertex labels. For example, for  $\mathcal{A} = \{0, 1\}$ , labels 01 and 10 are grouped together, while for  $\mathcal{A} = \{0, 1, 2\}$  labels 001, 010, and 100, or labels 122, 212, and 221, among others, are identical after the lexicographical sorting and thus belong to the same equivalence classes. The number of equivalence classes, in turn, corresponds to the number of terms in the multinomial expansion of  $(x_1 + x_2 + \dots + x_{|\mathcal{A}|})^2$ . In general, multinomial expansion of a  $k$ -nomial raised to the  $n^{\text{th}}$  power, that is,  $(x_1 + x_2 + \dots + x_k)^n$ , corresponds to  $n$  symmetric residues over an alphabet of size  $k$  and has  $C(n+k-1, k-1)$  multinomial coefficients, where  $C(n, m) = \binom{n}{m}$ . Therefore, the total number of labeled orbits  $o_3$  and  $o_4$  is  $|\mathcal{A}| \cdot C(|\mathcal{A}| + 1, |\mathcal{A}| - 1)$ . Extending this calculation to the remaining orbits, we obtain that the number of distinct labels of orbits  $o_8$  and  $o_{15}$  is  $|\mathcal{A}| \cdot C(|\mathcal{A}| + 2, |\mathcal{A}| - 1)$ , and the number of labels of orbits  $o_7, o_9, o_{10}, o_{12}, o_{13}$ , and  $o_{14}$  is  $|\mathcal{A}|^2 \cdot C(|\mathcal{A}| + 1, |\mathcal{A}| - 1)$ . In total, when  $|\mathcal{A}| = 20$ , the dimensionality of the encoding for a single vertex  $x$  adds up to  $\dim\{\Phi(x)\} = 1,062,420$ . We observe that for certain tasks, for example, prediction of phosphorylation sites, only a subset of residues can be phosphorylated (S, T, Y), and thus the number of choices for the pivot position is three instead of 20. Alternatively, a separate predictor can be trained on each residue, which effectively reduces the dimensionality of the representation to  $\dim\{\Phi(x)\} = 53,121$ .

### 2.5. Computing the kernel matrix

Two observations can be made about the vectors of counts. First, most of the entries in  $\Phi(x)$  will be zero due to the limited number of residues that can be placed in the volume of radius  $3r$  ( $r$  being the threshold distance in the construction of the contact graph) and the nature of grouping of amino acids (e.g., a clique of four positively charged residues is rare). Second, it is likely that a number of non-zero entries in each vector will have a zero as the corresponding entry in the other vector, and thus these counts will not contribute to the inner product. The first observation allows us to speed-up the computation of the inner product by using a sparse vector representation, that is, using (*key, value*) pairs, and either sort join or hash join to match the labeled orbit counts. In sort join, both vectors are sorted based on keys, and then joined in time linear in the sum of sizes of the vectors. In the hash join, a hash table is built based on the (*key, value*) pairs of one vector in linear time. The other vector is read sequentially and used to probe the hash table, for an expected linear time, which can degenerate to quadratic in the worst case. The second observation leads to even more significant speed-ups in practice. If the pivot residue is invariant, in a graphlet of size up to four, there are

up to three amino acid labels for every graphlet (denoted as  $a$ ,  $b$ , and  $c$  in Fig. 2). Since the ordering of labels has already been done during the label assignment step, we can construct a trie of labels of depth 3, with counts of  $o_1$  orbits in vertices at depth 1, counts of  $o_2$ ,  $o_3$ , and  $o_4$  orbits at vertices of depth 2, and so on. In this scenario, merging allows skipping subtrees for which the prefix leading to the subtree does not occur. The trie merge method could be combined with the graphlet counting step, where it would eliminate combinations of vertices based on their labels. For an overview of efficient strategies for string kernel computations, we direct the reader to an article by Rieck and Laskov (2008).

## 2.6. Classification

In the prediction step, the binary classification score of a query vertex  $q$  is computed as

$$\text{score}(q) = \sum_i \alpha_i \cdot d_i \cdot K(x_i, q)$$

where  $x_i$  is the  $i^{\text{th}}$  support vector coming from the training set,  $d_i \in \{+1, -1\}$  is the class label of  $x_i$ , and  $\alpha_i$  is the  $i^{\text{th}}$  Lagrange multiplier computed in the SVM learning step. This compact expression suggests a way to efficiently compute the prediction score, since all support vectors can be stored in a single data structure (hash table, trie, or a suffix tree) where the weight for each labeled orbit would correspond to a sum of coefficients  $\alpha_i \cdot d_i$  over all support vectors. The prediction score can be mapped into a probability using an approach by Platt (1999).

## 2.7. Performance evaluation

A prototype implementation of the graphlet kernel was coded in C++, using SVM<sup>light</sup> (Joachims, 2002) as the prediction engine. It was compared against a sequence-based predictor and our implementation of the FEATURE method (Bagley and Altman, 1995; Wei and Altman, 1998).

**2.7.1. Sequence-based predictor.** The sequence-based predictor builds a model similar to DisPhos 1.3 (Iakoucheva et al., 2004), a state-of-the-art phosphorylation site predictor. Sequence attributes were constructed as amino acid compositions and various physicochemical and predicted properties in concentric windows of length up to 21 around a pivot residue. In addition, we used binary representation for each position around the pivot up to  $\pm 12$  positions. An SVM with a linear kernel was used as the learning model. We refer to the sequence-based predictor as SEQUENCE.

**2.7.2. FEATURE predictor.** We implemented a simplified version of the FEATURE method in which amino acids were counted in a sphere of radius  $r$  or in radial intervals  $(r_1, r_2]$ . The counts of the 20 individual amino acids and 12 groups of amino acids were used in a vector encoding for each site, while the counts of atoms were ignored. Amino acid were grouped according to their physicochemical properties into aliphatic (A, V, L, I), hydroxyl-containing (S, T, Y), amide-containing (N, Q), sulfur-containing (C, M), acidic (D, E), basic (K, R, H), charged (D, E, R, K, H), aromatic (F, Y, W), polar (R, N, D, C, E, Q, H, K, S, T, W, Y), hydrophobic (A, C, G, I, L, M, F, P, W, Y), hydrophilic (R, N, D, E, K, S, T, V), and small (A, G, C, S). The following vector representations were constructed: (1) FEATURE, based on the original radial intervals defined elsewhere (Bagley and Altman, 1995; Wei and Altman, 1998):  $(0, 1.875]$ ,  $(1.875, 3.75]$ ,  $(3.75, 5.625]$ , and  $(5.625, 7.5]\text{\AA}$ ; (2) FEATURE<sub>6-12-18</sub>, based on radial intervals  $(0, 6]$ ,  $(6, 12]$ , and  $(12, 18]\text{\AA}$ ; and (3) FEATURE<sub>18</sub>, based on a single sphere of radius of  $18\text{\AA}$ . FEATURE<sub>6-12-18</sub> representation was chosen to mimic our construction of the protein contact graph (with  $C_\alpha-C_\alpha$  distance of  $6\text{\AA}$ ) and the level-3 neighborhood considered by the graphlet kernel, while FEATURE<sub>18</sub> was selected to quantify the difference between the cases of one sphere of radius  $18\text{\AA}$  and three radial intervals covering the same physical space. After the encoding was performed, each predictor was trained using a linear kernel and the default capacity parameter ( $C$ ) in the SVM<sup>light</sup> package.

The two predictors were chosen in order to provide fair and useful comparisons between methods, for example, such that conclusions can be drawn regarding the performance increase from sequence-based to structure-based models. Observe that FEATURE with only one shell of radius  $r$  is a special case of the graphlet kernel in which the protein contact graph is constructed using threshold  $r$  and where only graphlet  $g_1$  is used. Thus, the comparisons between the graphlet kernel and FEATURE can also provide information on the value of modeling interdependencies between residues within one sphere or shell.

2.7.3. *Cross-validation.* We employed leave-one-chain-out performance evaluation, in which one PDB chain was held out at a time. A model was trained on the remaining chains and then tested on the chain that was excluded during the training. This type of evaluation most closely resembles the realistic scenario in which a user would provide one chain at a time for prediction. We estimated sensitivity ( $sn$ ), specificity ( $sp$ ), precision ( $pr$ ), and area under the receiver operating characteristic (ROC) curve ( $AUC$ ) for each set of parameters. Sensitivity is defined as the true positive rate, the specificity is defined as the true negative rate, while the precision is defined as the fraction of positively predicted residues that are correctly predicted. ROC curve plots  $sn$  as a function of  $(1 - sp)$  over all decision thresholds.

### 3. EXPERIMENTS AND RESULTS

We conducted a comprehensive set of experiments with the goal of characterizing the performance of the graphlet kernel with respect to the choice of parameters (size of alphabet and normalization of the kernel function). The principal difference between the three methods is in how they model the analyzed residue and its surroundings. To minimize the influence of other variables on the outcome, all methods were trained using the same prediction engine (SVM<sup>light</sup>) and similarity measure (i.e., the inner product of the vector representations). The two data sets, CSA and PHOS, were split into subsets based on the analyzed amino acid. Thus, 20 distinct models were built for the CSA data set and three models for PHOS.

#### 3.1. Data sets

3.1.1. *CSA.* We selected all catalytic residues from the Catalytic Site Atlas (CSA) v.2.2.9 (Porter et al., 2004) found in the ASTRAL 40 v.1.73 structures (Chandonia et al., 2004), as positive examples. Catalytic activity in CSA has been assigned either experimentally or via function transfer, using PSI-BLAST (Altschul et al., 1997). When different groups of residues were annotated based on function transfer from different proteins, we included the union of residues from all groups. All remaining residues in the respective chains were considered to be negative examples. For a description of the data sets, see Tables 1 and 2.

TABLE 1. SUMMARY OF THE CSA DATA SET

<i>Residue</i>	<i>Sites</i>	<i>Chains</i>	<i>Non-sites</i>	<i>NS/S Ratio</i>	<i>Total</i>
Ala	110	102	3,310	30.09	3,420
Arg	618	493	9,119	14.76	9,737
Asn	329	310	4,685	14.24	5,014
Asp	1,116	836	16,710	14.97	17,826
Cys	292	222	950	3.25	1,242
Gln	135	129	1,532	11.35	1,667
Glu	740	626	14,422	19.49	15,162
Gly	347	238	5,909	17.03	6,256
His	934	712	5,801	6.21	6,735
Ile	48	46	705	14.69	753
Leu	65	62	1,580	24.31	1,645
Lys	634	531	10,005	15.78	10,639
Met	27	27	226	8.37	253
Phe	128	105	1,408	11.00	1,536
Pro	36	36	554	15.39	590
Ser	468	366	7,506	16.04	7,974
Thr	248	212	3,779	15.24	4,027
Trp	92	76	634	6.89	726
Tyr	426	373	4,824	11.32	5,250
Val	37	37	974	26.32	1,011
<b>Total</b>	<b>6,830</b>	<b>2,025</b>	<b>94,633</b>	<b>13.86</b>	<b>101,463</b>



TABLE 2. SUMMARY OF THE PHOS DATA SET

<i>Residue</i>	<i>Sites</i>	<i>Chains</i>	<i>Non-sites</i>	<i>NS/S Ratio</i>	<i>Total</i>
Ser	627	427	5,068	8.08	5,695
Thr	237	206	2,124	8.96	2,361
Tyr	293	235	1,526	5.21	1,819
<b>Total</b>	1,157	686	8,718	7.54	9,875

3.1.2. *PHOS*. It has previously been shown that protein phosphorylation sites are preferentially located in intrinsically disordered protein regions (Collins et al., 2008; Iakoucheva et al., 2004). There are, however, examples where phosphorylation sites can also be found in ordered regions (Iakoucheva et al., 2004; Johnson and Lewis, 2001). Furthermore, local or global conformational changes between folded conformations as well as disorder-to-order or order-to-disorder transitions could occur upon covalent attachment of the phosphate group to the side chain of a phosphorylated residue (Espinoza-Fonseca et al., 2007, 2008; Johnson and Lewis, 2001; Shen et al., 2005), leading to mischaracterization of disordered regions. In order to investigate the structural properties of phosphorylation sites in greater detail, we searched PDB for records that contain keywords “phosphoserine,” “phosphothreonine,” or “phosphotyrosine,” and which contain residue symbols Sep, Tpo, or Ptr in the HETATM lines. A non-redundant subset of proteins (<40% sequence identity between any pair of chains) is reported in Table S1 (see online Supplementary Material at www.liebertonline.com). We found a very limited number of annotated phosphorylation sites for which the structure has been determined: 48 phosphoserines in 35 non-redundant PDB chains, 20 (19) phosphothreonines, and 25 (20) phosphotyrosines.

The set of phosphorylation sites with solved structures was not large enough for systematic evaluation of our method. In order to expand the data set, we aligned a comprehensive set of sequences with experimentally annotated phosphorylation sites against the set of ASTRAL sequences using BLAST (Altschul et al., 1997). This set of sites was compiled from the proteins annotated in UniProt release 54.3 (Bairoch et al., 2005), Phospho.ELM (Diella et al., 2004), Phosida (Gnad et al., 2007), dbPTM (Lee et al., 2006), and through a survey of the literature (Ballif et al., 2004; Beausoleil et al., 2004; Ficarro et al., 2002; Fujii et al., 2004; Rush et al., 2005) (Table S2; see online Supplementary Material at www.liebertonline.com). We included only alignments longer than 50 consecutive residues, with at least 90% sequence identity, with Sep/Tpo/Ptr correctly aligned against Ser/Thr/Tyr, and without missing residues in the aligned segment. All other Ser/Thr/Tyr in the returned protein structures were added to the dataset as negatives. Aligning against ASTRAL40 still did not produce enough data points for training (291 S, 122 T, and 140 Y). Thus, we decided to map the phosphosites from known sequences to ASTRAL95 (627 S, 237 T, and 293 Y), which provided a compromise between data set redundancy and size. Compared to the number of phosphosites annotated within protein sequences, the fractions of these sites mapped onto ASTRAL40 (1.5% S, 2.7% T, 6.4% Y), ASTRAL95 (3.2% S, 5.1% T, 13.4% Y), or all structures in PDB (4.0% S, 6.8% T, 18.2% Y) were significantly smaller even though we allowed for inexact matches.

### 3.2. Construction of protein contact graphs

There is no universally agreed upon convention about when two residues are in contact and should be connected with an edge. A number of studies have looked at the distances between  $C_\alpha$  or  $C_\beta$  atoms, with appropriately chosen thresholds, for example, 8.5Å (Dokholyan et al., 2002) or in the 3–6Å range (Hu et al., 2007). An alternative is to look at distances between any two atoms and consider the residues to be in contact if the distance is below 5Å (Greene and Higman, 2003) or the sum of their van der Waals radii plus 0.5Å (Keskin and Nussinov, 2007). Brinda et al. (2002) proposed considering the strength of interaction between the amino acids, defined as the normalized number of atom-atom pairs below a cutoff distance.

Pollastri et al. (2002) analyzed protein coordination numbers, which correspond to degrees of vertices in our framework. They performed experiments with distances between  $C_\alpha$  atoms with thresholds set at 6, 8, 10, and 12Å and reported high similarity between residues in contact for thresholds 8Å or higher. With the goal of understanding how these choices influence the resulting protein contact graph, we extended their approach to four connection methods (distances between  $C_\alpha$ ,  $C_\beta$ , all atom pairs, and all atom pairs taking into consideration their van der Waals radii) and a number of appropriate thresholds. We generated graphs

based on all chains in the October 2007 version of PDBSelect25 (Hobohm and Sander, 1994). The similarity between sets of edges was quantified using the Jaccard similarity coefficient, defined for two sets  $A$  and  $B$  as  $J(A, B) = |A \cap B| / |A \cup B|$ , as shown in Table S5 (see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)).

The methods based on the distances between  $C_\alpha$  and  $C_\beta$  atoms are computationally more efficient because they perform a quadratic number of distance calculations in the number of residues, whereas atom level methods perform a quadratic number of computations in the number of atoms. Methods operating at the atom level have potentially higher sensitivity to the underlying biochemistry, in particular when the differences in the van der Waals radii are taken into consideration. However, the van der Waals method is the only one that cannot avoid an expensive square root operation. Based on our experiments,  $C_\alpha$ - and  $C_\beta$ -based methods display a relatively strong dependency on the threshold parameter, which is indicated by the low Jaccard coefficients between sets of edges (Table S5; see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)). Atom level methods are generally in good agreement with each other and are more robust to the choice of the threshold distance parameter. Also, they are in good agreement with the  $C_\alpha$  and  $C_\beta$  methods for 6Å threshold. These results have led us to choose the  $C_\alpha$ -based method with 6Å distance cut-off to build the protein contact graph, as a good compromise between speed and sensitivity.

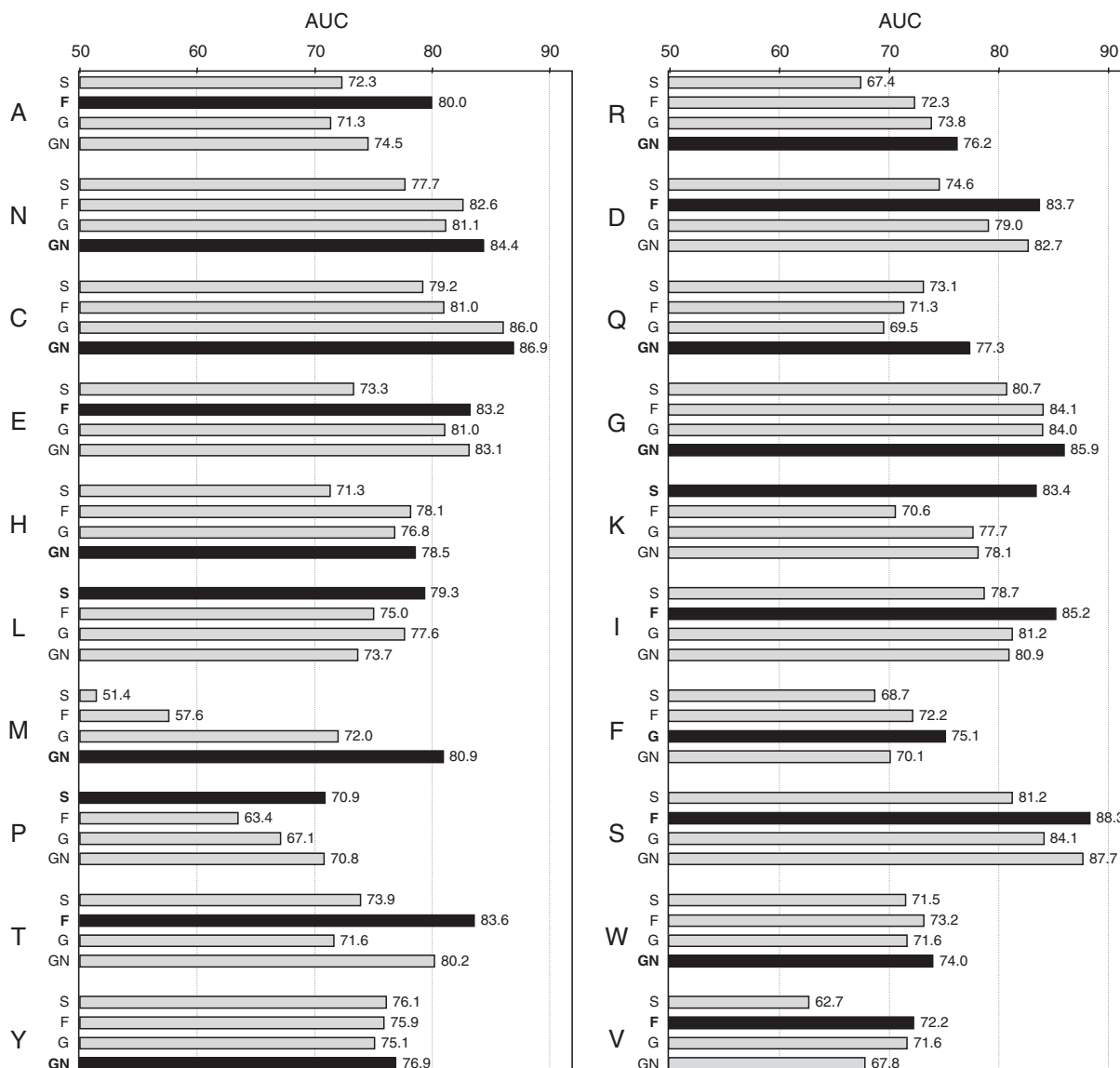
### 3.3. Parameter selection

On the PHOS data set, the non-normalized graphlet kernel resulted in slightly better AUC values than the normalized kernel, with 0.9, 2.6, and 0.1 percentage points advantage for Ser, Thr, and Tyr, respectively (Fig. 3). The best results were achieved using the full alphabet ( $|\mathcal{A}| = 20$ ), and the reduction in alphabet size was strongly correlated with the decrease in AUC. On the CSA data, the overall best classification model was the normalized graphlet kernel built using the full alphabet. Here, the dependence on the alphabet size was less clear. In some cases, AUC correlated with the alphabet size (e.g., C, G). In other cases, AUC was generally robust to the changes in alphabet size (e.g., A, K), while in the remaining cases there was no clear trend (e.g., P, T). Interestingly, the unlabeled graphlet kernels, i.e., when  $|\mathcal{A}| = 1$ , were consistently inferior to the remaining models. Perhaps not surprisingly, this suggests that in the analysis of protein structure graphs the connectivity information alone, which in turn is correlated with surface accessibility, is not useful for classification. We report all AUC values in Figures S2–S24, and Tables S2 and S3 (see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)).

To avoid overfitting, we evaluated the performance of all methods on one amino acid subset at a time, and the remaining amino acid subsets (19 for CSA and two for PHOS) were used to select the best performing predictor. The selected alphabet size for the normalized kernel was 20 both for CSA (best in 60% of the subsets) and PHOS (100%). For the non-normalized kernel, the selected alphabet size was 15 on CSA (27.5%) and 20 on PHOS (100%). The best overall performing variant of FEATURE was FEATURE<sub>6-12-18</sub>. The list of all results is provided in Tables S2 and S3, and Figures S2–S24 (see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)). The reduced alphabets were created by hierarchical clustering of amino acids using BLOSUM50 scoring matrix as a measure of similarity (Fig. S1; see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)).

### 3.4. Performance of the graphlet kernel model

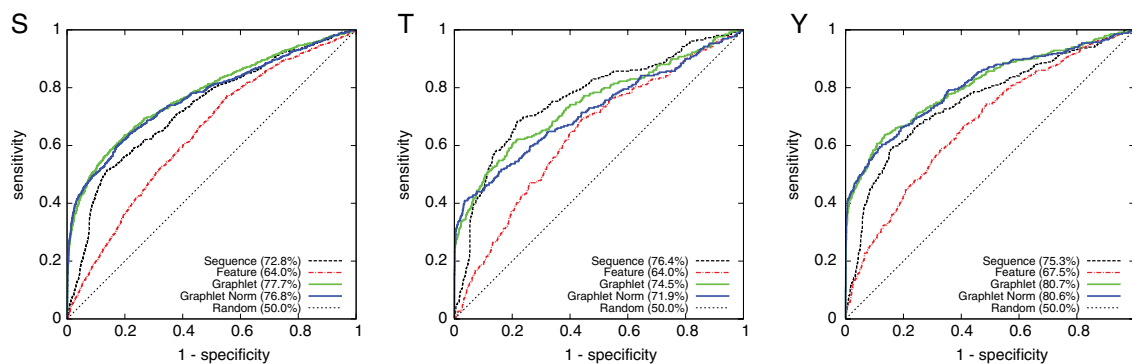
The normalized and the non-normalized versions of the graphlet kernel generally performed better than the alternative methods, both in terms of mean AUC values and the number of subsets on which each method outperformed all others. On the CSA data set, the mean AUC values were  $73.2 \pm 7.3\%$  (SEQUENCE),  $76.7 \pm 7.8\%$  (FEATURE),  $76.7 \pm 5.2\%$  (non-normalized graphlet kernel), and  $78.8 \pm 5.9\%$  (normalized graphlet kernel). On the PHOS data set, the mean AUC values were  $74.8 \pm 1.8\%$  (SEQUENCE),  $65.2 \pm 2.0\%$  (FEATURE),  $77.6 \pm 3.1\%$  (non-normalized graphlet kernel), and  $76.4 \pm 4.4\%$  (normalized graphlet kernel). Figure 3 shows a bar plot with the AUC values corresponding to the CSA data set. Figure 4 illustrates the ROC curves for the PHOS data set. On the CSA data set, the sequence-based predictor performed best on 15% of the subsets (I, L, P), FEATURE on 35% (A, D, E, S, T, K, V), the non-normalized graphlet kernel on 5% (F), and the normalized graphlet kernel on 45% (R, N, C, Q, G, H, W, M, Y). On the PHOS data set, the sequence-based predictor achieved the best performance for Thr, while the non-normalized graphlet kernel was the best model on Ser and Tyr.



**FIG. 3.** Area under the ROC curve for the CSA data sets. Method compared: S, SEQUENCE; F, FEATURE<sub>6-12-18</sub>; G, graphlet kernel with full alphabet; GN, normalized graphlet kernel with full alphabet. Bold bars indicate the method with the best performance on an individual data set. ROC, receiver operating characteristic; AUC, area under the ROC curve; CSA, Catalytic Site Atlas.

Though ROC curves are a useful way of comparing classification models, it can be observed in Figure 4 that the steepest slope of the ROC curve was consistently observed for the graphlet kernel. The lower left part of the ROC curve corresponds to the predictions with the highest scores. Thus, we evaluated the sensitivity of these predictors for a given precision of 95% (i.e., false discovery rate of 5%). On the PHOS-SER data set, the recall for the SEQUENCE, FEATURE, and graphlet kernel methods were 19.6%, 11.2%, 42.7% (non-normalized), and 43.8% (normalized), respectively. In the case of the PHOS-THR data set, these values were 20.5%, 11.6%, 37.8%, and 41.7%, while in the case of PHOS-TYR data set, the sensitivities at 5% false discovery rate were 23.9%, 17.4%, 48.4%, and 49.9%, respectively.

It is worth noting that the performance of the graphlet kernel was consistently good on both classes of problems. FEATURE performed well on the CSA data, but its accuracy dropped on the PHOS data. On the other hand, the sequence-based predictor performed well on PHOS, but less well on the CSA data set.

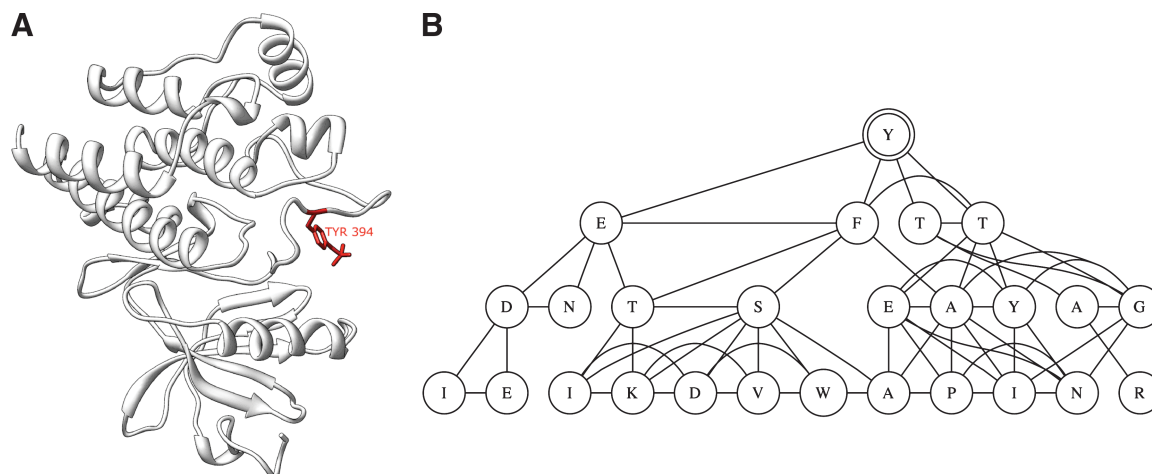


**FIG. 4.** ROC plots for the PHOS data sets: (S) phosphoserine, (T) phosphothreonine, and (Y) phosphotyrosine. Red curve, FEATURE<sub>6-12-18</sub>; black curve, SEQUENCE; green curve, graphlet kernel with full alphabet; blue curve, normalized graphlet kernel with full alphabet. ROC, receiver operating characteristic; PHOS, phosphorylation.

Although experimental testing of our method was far from exhaustive, the results presented here suggest that the graphlet count methodology might be more general than the other methods evaluated in this study. As an illustration of the graphlet patterns centered on a phosphorylation site, we show a 3D structure of the human lymphocyte kinase (Lck) with highlighted phosphosite Tyr394 (Fig. 5A) along with its level-3 neighborhood in the protein contact graph (Fig. 5B).

### 3.5. Structure of ordered protein phosphorylation sites

It was previously proposed that phosphorylation sites preferentially, although not exclusively, appear in intrinsically disordered protein regions (Iakoucheva et al., 2004). A recent mass spectrometry study of 162 cytosolic phosphoproteins provided an experimental confirmation: out of 512 phosphorylation sites, 97% occurred outside of structured domains, and 86% occurred in regions of protein disorder (Collins et al., 2008). Nonetheless, there are examples of ordered phosphorylation sites (Johnson and Lewis, 2001; Keane et al., 1994; Quirk et al., 1996; Tholey et al., 2001), and a number of structures containing phosphorylated sites have been deposited in PDB (in the Data Sets section). In addition, several studies addressed conformational changes in proteins following the addition of the phosphate group (Groban et al. 2006; Latzer et al., 2008; Shen et al., 2005). In Table S1 (see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)), we report a non-redundant subset of phosphorylated sites mined from PDB and results of the search for the structures of proteins that can be found both in the phosphorylated and the unphosphorylated states. The search returned 34 structures with 49 previously phosphorylated residues. Interestingly, in most cases, the structural change between the two proteins was minimal, suggesting that in these cases phosphorylation



**FIG. 5.** Structure of human lymphocyte kinase (Lck) with highlighted phosphorylation site Tyr394 (A) and the corresponding level-3 protein contact graph centered at Tyr394 (B).

TABLE 3. SECONDARY STRUCTURE ASSIGNMENTS OF THE PHOSPHORYLATION SITES FOUND IN PDB: NUMBER OF SITES AND PERCENTAGE

	<i>Helix</i>	<i>Sheet</i>	<i>Loop</i>	<i>Turn</i>
Serine	13 (28.9%)	3 (0.7%)	17 (37.8%)	12 (26.7%)
Threonine	2 (10%)	0	18 (90%)	0
Tyrosine	0	6 (24%)	11 (44%)	8 (32%)
<b>Total</b>	15 (16.7%)	9 (10%)	46 (51.1%)	20 (22.2%)

may not be affecting protein function via an allosteric effect, but rather via introducing a binding site. We also analyzed secondary structure of the annotated phospho-residues in PDB using Dictionary of Protein Secondary Structure (DSSP) (Kabsch and Sander, 1983), and found that 73.3% of the sites were located in loops and turns (Table 3), which is consistent with the fact that phosphorylation sites tend to be located in flexible regions in order to fit into the kinase recognition pocket.

Another interesting observation is the prevalence of kinases in the ordered subset of phosphosites from PDB (Table S1; see online Supplementary Material at [www.liebertonline.com](http://www.liebertonline.com)). For example, the majority (14 out of 25) of threonine phosphorylation sites (and to a lesser degree of those of serine and tyrosine sites) were found in kinases, which may suggest that ordered phosphosites preferentially occur in kinases and might be important for their regulation or for regulation of the phosphorylation process catalyzed by the kinases. Currently, there are only 112 kinases in PDB (with <90% pairwise sequence identity); thus, an alternative explanation that structures of kinases are more frequently studied and hence overrepresented in PDB seems unlikely. Additionally, we observed that the majority of ordered tyrosine phosphorylation sites in kinases (9 out of 14) are in fact autophosphorylation sites.

In summary, we conclude that (1) protein phosphorylation sites are indeed preferentially located in disordered regions because only a very small subset of them could be found in PDB; (2) ordered phosphorylation sites are preferentially located in protein loops thereby potentially facilitating the access of kinases to the phosphorylatable residue; (3) for the cases of ordered phosphorylation sites currently present in PDB there are minimal structural changes that occur upon phosphorylation with only a few examples of order-to-disorder transitions; (4) ordered phosphorylation sites, especially for threonine, are enriched among kinases; and (5) ordered tyrosine phosphorylation sites are frequently found to be autophosphorylation sites.

#### 4. DISCUSSION

In this study, we propose and evaluate a computational method for predicting functional residues in protein structures. The method is based on a graph representation of protein structure and a kernel-based strategy for probabilistic binary labeling of vertices. In the broader context of machine learning, our graphlet kernel belongs to the graph classification methods because our implementation considers vertex neighborhoods up to a fixed level and these neighborhoods can be treated as isolated graphs. In this framework, we are given a set  $\mathcal{G} = \{(G, y)\}_1^n$ , where  $G$  is an undirected labeled graph,  $y \in \{+1, -1\}$  is the class label of  $G$ , and the objective is to develop a classifier. Several kernel methods have been recently developed for this problem, for example, a random-walk kernel (Gaertner et al., 2003), a cycle pattern kernel (Horvath et al., 2004), weighted decomposition kernel (Menchetti et al., 2005), and others (Kashima et al., 2003; Ralaivola et al., 2005). However, in the vertex labeling problem considered in this study, each graph  $G$  also contains a special node called pivot, and our method exploits its presence effectively.

The graphlet kernel was applied to the problem of residue-level function prediction from protein structure. In the world of microenvironment-based and template-based approaches developed for the prediction of protein functional sites, our method appears to be the most similar to the FEATURE framework (Bagley and Altman, 1995; Wei and Altman, 1998, 2003). FEATURE works at the atomic level and the residue level simultaneously and also exploits various physicochemical properties of amino acids. As mentioned previously, its residue level component can be seen as a special case of the graphlet kernel, where only graphlets of type  $g_1$  are used and where the distance thresholds for the construction of protein structure graphs are varied. It would be relatively straightforward to extend the graphlet kernel to incorporate multiple distance thresholds as well as the atom-level component, for example, via a fusion kernel

(Lanckriet et al., 2004a,b) or the hyper-kernel approach (Borgwardt et al., 2005), but such an extension is beyond the scope of this study.

It is worth mentioning that the graphlet kernel framework is also related to the spectrum kernel (Kuang et al., 2005; Leslie et al., 2002; Leslie and Kuang, 2004). The spectrum kernel is a strategy for classifying proteins based on the counts of  $k$ -mers in their primary structure. If one was to construct a protein structure graph such that only vertices corresponding to the neighboring residues in protein sequence are connected by edges, the graphlet representation would effectively count strings, thus resembling the spectrum kernel strategy. In addition, the computation of the graphlet kernel function is similar to an efficient algorithm proposed by Leslie et al. (2002) which was later further formalized and systematically evaluated by Rieck and Laskov (2008).

We chose to evaluate our method against a sequence-based predictor, the original FEATURE algorithm (though counts of atoms were ignored in our implementation), and two commonsense variations thereof. All predictors were systematically evaluated on a classical problem of the catalytic residue prediction and also on a less explored problem of prediction of phosphorylation sites from protein structure. Blom et al. (1999) were the first to develop a phosphorylation site predictor from the predicted contact maps given a protein sequence; however, its accuracy was inferior to their sequence-based model. We believe that this was due to the fact that protein contact maps cannot be precisely inferred from sequence data alone compared to the fragment assembly approaches (Izarzugaza et al., 2007). In addition, many phosphorylation sites lie in the disordered regions for which a time-invariant contact map may not even exist (Iakoucheva et al., 2004). Thus, the model by Blom et al., as well as many others (Brinkworth et al., 2003; Fujii et al., 2004; Hjerrild et al., 2004; Iakoucheva et al., 2004; Kim et al., 2004; Obenauer et al., 2003), was developed from amino acid sequence or aligned sequences. Here we provide evidence that the knowledge of protein 3D structure is beneficial for the prediction of ordered phosphorylation sites. We also demonstrate that the graphlet kernel fared favorably against the alternative strategies. While our model was constructed from proteins deposited in PDB, it is straightforward to extend it to structural models which can be constructed with increasing accuracy (Kopp et al., 2007).

In previous work, it was hypothesized that phosphorylation sites preferentially occur in intrinsically disordered regions (Iakoucheva et al., 2004). This hypothesis has been validated in recent experimental studies (Collins et al., 2008; Gsponer et al., 2008), and in several cases, disorder-to-order transition upon phosphorylation has been predicted (Espinoza-Fonseca et al., 2007; Hamelberg et al., 2007; Hegedus et al., 2008). Furthermore, it has recently been shown that disordered proteins are substrates of twice as many kinases as are ordered proteins (Gsponer et al., 2008). However, for a subset of phosphorylatable residues that are structured under physiological conditions, current study strongly suggests that not only the average structural and physicochemical properties are important, but also the particular interconnectedness of the residues within the microenvironments also considered by FEATURE. In addition, the consistently inferior performance of the unlabeled graphlet kernel ( $|\mathcal{A}| = 1$ ) suggests that in the analysis of protein structure graphs, unlike protein-protein interaction networks, the connectivity information alone is not sufficient to generate useful classification models.

## ACKNOWLEDGMENTS

We would like to thank Chia-en Angelina Chang (UC Riverside) for insightful comments about the structure of protein phosphorylation sites and Matthew W. Hahn (Indiana University) for proofreading the manuscript. This work was supported by the following grants: NIH 1R21CA113711 (Principal Investigator [PI]: Iakoucheva), NSF IIS-0447773 (PI: Lonardi), NSF DBI-0321756 (co-PI: Lonardi), and NSF DBI-0644017 (PI: Radivojac).

## NOTE ADDED IN PROOF

It was brought to our attention that during the review of this manuscript a graphlet kernel approach for unlabeled graphs has been proposed by Dr. Karsten M. Borgwardt and collaborators at AISTATS 2009 conference. The corresponding reference is: Shervashidze N, Vishwanathan SVN, Petri TH, Mehlhorn K, Borgwardt KM. Efficient graphlet kernels for large graph comparison. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, pp. 488–495, April 2009. After communication with Dr. Karsten M. Borgwardt, we concluded that the graphlet kernel

terminology has been independently proposed in two Ph.D. dissertations: (1) Borgwardt KM, Graph kernels, University of Munich, Germany, 2007, and (2) Vacic V, Computational methods for discovery of cellular regulatory mechanisms, University of California, Riverside, CA, 2008.

## DISCLOSURE STATEMENT

No competing financial interests exist.

## REFERENCES

- Altschul, S.F., Madden, T.L., Schaffer, A.A., et al. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.
- Artymiuk, P.J., Poirrette, A.R., Grindley, H.M., et al. 1994. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J. Mol. Biol.* 243, 327–344.
- Atilgan, A.R., Akan, P., and Baysal, C. 2004. Small-world communication of residues and significance for protein dynamics. *Biophys. J.* 86, 85–91.
- Bagley, S.C., and Altman, R.B. 1995. Characterizing the microenvironment surrounding protein sites. *Protein Sci.* 4, 622–635.
- Bairoch, A., Apweiler, R., Wu, C.H., et al. 2005. The Universal Protein Resource (UniProt). *Nucleic Acids Res.* 33, Database Issue, D154–D159.
- Ballif, B.A., Villen, J., Beausoleil, S.A., et al. 2004. Phosphoproteomic analysis of the developing mouse brain. *Mol. Cell Proteomics* 3, 1093–1101.
- Bandyopadhyay, D., Huan, J., Liu, J., et al. 2006. Structure-based function inference using protein family-specific fingerprints. *Protein Sci.* 15, 1537–1543.
- Beausoleil, S.A., Jedrychowski, M., Schwartz, D., et al. 2004. Large-scale characterization of HeLa cell nuclear phosphoproteins. *Proc. Natl. Acad. Sci. U.S.A.* 101, 12130–12135.
- Berman, H., Bhat, T.N., Bourne, P., et al. 2000. The protein data bank and the challenge of structural genomics. *Nat. Struct. Biol.* 7, 957–959.
- Blom, N., Gammeltoft, S., and Brunak, S. 1999. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *J. Mol. Biol.* 294, 1351–1362.
- Borgwardt, K.M., Ong, C.S., Schonauer, S., et al. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21, Suppl 1, i47–i56.
- Brinda, K.V., Kannan, N., and Vishveshwara, S. 2002. Analysis of homodimeric protein interfaces by graph-spectral methods. *Protein Eng.* 15, 265–277.
- Brinkworth, R.I., Breinl, R.A., and Kobe, B. 2003. Structural basis and prediction of substrate specificity in protein serine/threonine kinases. *Proc. Natl. Acad. Sci. U.S.A.* 100, 74–79.
- Burley, S.K., Almo, S.C., Bonanno, J.B., et al. 1999. Structural genomics: beyond the human genome project. *Nat. Genet.* 23, 151–157.
- Chandonia, J.M., Hon, G., Walker, N.S., et al. 2004. The ASTRAL Compendium in 2004. *Nucleic Acids Res.* 32, D189–D192.
- Collins, M.O., Yu, L., Campuzano, I., et al. 2008. Phosphoproteomic analysis of the mouse brain cytosol reveals a predominance of protein phosphorylation in regions of intrinsic sequence disorder. *Mol. Cell Proteomics* 7, 1331–1348.
- Dalkilic, M.M., Costello, J.C., Clark, W.T., et al. 2008. From protein-disease associations to disease informatics. *Front. Biosci.* 13, 3391–3407.
- Diella, F., Cameron, S., Gemund, C., et al. 2004. Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins. *BMC Bioinform.* 5, 79.
- Dobson, C.M. 2001. The structural basis of protein folding and its links with human disease. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356, 133–145.
- Dokholyan, N.V., Li, L., Ding, F., et al. 2002. Topological determinants of protein folding. *Proc. Natl. Acad. Sci. U.S.A.* 99, 8637–8641.
- Elcock, A.H. 2001. Prediction of functionally important residues based solely on the computed energetics of protein structure. *J. Mol. Biol.* 312, 885–896.
- Espinoza-Fonseca, L.M., Kast, D. and Thomas, D.D. 2007. Molecular dynamics simulations reveal a disorder-to-order transition on phosphorylation of smooth muscle myosin. *Biophys. J.* 93, 2083–2090.
- Espinoza-Fonseca, L.M., Kast, D. and Thomas, D.D. 2008. Thermodynamic and structural basis of phosphorylation-induced disorder-to-order transition in the regulatory light chain of smooth muscle myosin. *J. Am. Chem. Soc.* 130, 12208–12209.

- Fetrow, J.S., and Skolnick, J. 1998. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. *J. Mol. Biol.* 281, 949–968.
- Ficarro, S.B., McClelland, M.L., Stukenberg, P.T., et al. 2002. Phosphoproteome analysis by mass spectrometry and its application to *Saccharomyces cerevisiae*. *Nat. Biotechnol.* 20, 301–305.
- Fujii, K., Zhu, G., Liu, Y., et al. 2004. Kinase peptide specificity: improved determination and relevance to protein phosphorylation. *Proc. Natl. Acad. Sci. U.S.A.* 101, 13744–13749.
- Gaertner, T., Flatch, P. and Wrobel, S. 2003. On graph kernels: hardness results and efficient alternatives. *Proc. 16th Annu. Conf. Comput. Learn. Theory 7th Kernel Workshop* 129–143.
- Glaser, F., Morris, R.J., Najmanovich, R.J., et al. 2006. A method for localizing ligand binding pockets in protein structures. *Proteins* 62, 479–488.
- Glazer, D.S., Radmer, R.J., and Altman, R.B. 2008. Combining molecular dynamics and machine learning to improve protein function recognition. *Pac. Symp. Biocomput.* 332–343.
- Gnad, F., Ren, S., Cox, J., et al. 2007. PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites. *Genome Biol.* 8, R250.
- Greene, L.H., and Higman, V.A. 2003. Uncovering network systems within protein structures. *J. Mol. Biol.* 334, 781–791.
- Gregory, D.S., Martin, A.C., Cheetham, J.C., et al. 1993. The prediction and characterization of metal binding sites in proteins. *Protein Eng.* 6, 29–35.
- Grindley, H.M., Artymiuk, P.J., Rice, D.W., et al. 1993. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* 229, 707–721.
- Groban, E.S., Narayanan, A., and Jacobson, M.P. 2006. Conformational changes in protein loops and helices induced by post-translational phosphorylation. *PLoS Comput. Biol.* 2, e32.
- Gsponer, J., Futschik, M.E., Teichmann, S.A., et al. 2008. Tight regulation of unstructured proteins: from transcript synthesis to protein degradation. *Science* 322, 1365–1368.
- Hamelberg, D., Shen, T., and McCammon, J.A. 2007. A proposed signaling motif for nuclear import in mRNA processing via the formation of arginine claw. *Proc. Natl. Acad. Sci. U.S.A.* 104, 14947–14951.
- Haussler, D. 1999. Convolution kernels on discrete structures. Technical Report UCSCCRL-99-10. University of California at Santa Cruz.
- Hegedus, T., Serohijos, A.W., Dokholyan, N.V., et al. 2008. Computational studies reveal phosphorylation-dependent changes in the unstructured R domain of CFTR. *J. Mol. Biol.* 378, 1052–1063.
- Hermann, J.C., Marti-Arbona, R., Fedorov, A.A., et al. 2007. Structure-based activity prediction for an enzyme of unknown function. *Nature* 448, 775–779.
- Hjerrild, M., Stensballe, A., Rasmussen, T.E., et al. 2004. Identification of phosphorylation sites in protein kinase A substrates using artificial neural networks and mass spectrometry. *J. Proteome Res.* 3, 426–433.
- Hobohm, U., and Sander, C. 1994. Enlarged representative set of protein structures. *Protein Sci.* 3, 522–524.
- Holm, L., and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233, 123–138.
- Horvath, T., Gaertner, T., and Wrobel, S. 2004. Cyclic pattern kernels for predictive graph mining. *Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discov. Data Mining* 158–167.
- Hu, Z., Bowen, D., Southerland, W.M., et al. 2007. Ligand binding and circular permutation modify residue interaction network in DHFR. *PLoS Comput. Biol.* 3, e117.
- Huan, J., Bandyopadhyay, D., Wang, W., et al. 2005. Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. *J. Comput. Biol.* 12, 657–671.
- Iakoucheva, L.M., Radivojac, P., Brown, C.J., et al. 2004. The importance of intrinsic disorder for protein phosphorylation. *Nucleic Acids Res.* 32, 1037–1049.
- Izarzugaza, J.M., Grana, O., Tress, M.L., et al. 2007. Assessment of intramolecular contact predictions for CASP7. *Proteins* 69, Suppl 8, 152–158.
- Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer Academic Publishers, Amsterdam.
- Johnson, L.N., and Lewis, R.J. 2001. Structural basis for control by phosphorylation. *Chem. Rev.* 101, 2209–2242.
- Kabsch, W., and Sander, C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.
- Kalinina, O.V., Mironov, A.A., Gelfand, M.S., et al. 2004. Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. *Protein Sci.* 13, 443–456.
- Kashima, H., Tsuda, K. and Inokuchi, A. 2003. Marginalized kernels between labeled graphs. *Proc. 20th Int. Conf. Mach. Learn.* 321–328.
- Keane, N.E., Chavanieu, A., Quirk, P.G., et al. 1994. Structural determinants of substrate selection by the human insulin-receptor protein-tyrosine kinase. *Eur. J. Biochem.* 226, 525–536.
- Keskin, O., and Nussinov, R. 2007. Similar binding sites and different partners: implications to shared proteins in cellular pathways. *Structure* 15, 341–354.



- Kim, J.H., Lee, J., Oh, B., et al. 2004. Prediction of phosphorylation sites using SVMs. *Bioinformatics* 20, 3179–3184.
- Kleywegt, G.J. 1999. Recognition of spatial motifs in protein structures. *J. Mol. Biol.* 285, 1887–1897.
- Kopp, J., Bordoli, L., Battey, J.N., et al. 2007. Assessment of CASP7 predictions for template-based modeling targets. *Proteins* 69, Suppl 8, 38–56.
- Kuang, R., Ie, E., Wang, K., et al. 2005. Profile-based string kernels for remote homology detection and motif extraction. *J. Bioinform. Comput. Biol.* 3, 527–550.
- Lanckriet, G.R., De Bie, T., Cristianini, N., et al. 2004a. A statistical framework for genomic data fusion. *Bioinformatics* 20, 2626–2635.
- Lanckriet, G.R., Deng, M., Cristianini, N., et al. 2004b. Kernel-based data fusion and its application to protein function prediction in yeast. *Pac. Symp. Biocomput.* 300–311.
- Laskowski, R.A., and Thornton, J.M. 2008. Understanding the molecular machinery of genetics through 3D structures. *Nat. Rev. Genet.* 9, 141–151.
- Laskowski, R.A., Watson, J.D., and Thornton, J.M. 2005a. ProFunc: a server for predicting protein function from 3D structure. *Nucleic Acids Res.* 33, W89–W93.
- Laskowski, R.A., Watson, J.D., and Thornton, J.M. 2005b. Protein function prediction using local 3D templates. *J. Mol. Biol.* 351, 614–626.
- Latzer, J., Shen, T., and Wolynes, P.G. 2008. Conformational switching upon phosphorylation: a predictive framework based on energy landscape principles. *Biochemistry*, 47, 2110–2122.
- Lee, D., Redfern, O., and Orengo, C. 2007. Predicting protein function from sequence and structure. *Nat. Rev. Mol. Cell Biol.* 8, 995–1005.
- Lee, T.Y., Huang, H.D., Hung, J.H., et al. 2006. dbPTM: an information repository of protein post-translational modification. *Nucleic Acids Res.* 34, D622–D627.
- Leslie, C., Eskin, E., and Noble, W.S. 2002. The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput.* 564–575.
- Leslie, C., and Kuang, R. 2004. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.* 5, 1435–1455.
- Liang, M.P., Banatao, D.R., Klein, T.E., et al. 2003. WebFEATURE: an interactive web tool for identifying and visualizing functional sites on macromolecular structures. *Nucleic Acids Res.* 31, 3324–3327.
- Menchetti, S., Costa, F., and Frasconi, P. 2005. Weighted decomposition kernels. *Proc. 22nd Int. Conf. Mach. Learn.* 585–592.
- Mooney, S.D. 2005. Bioinformatics approaches and resources for single nucleotide polymorphism functional analysis. *Brief. Bioinform.* 6, 44–56.
- Mooney, S.D., Liang, M.H., DeConde, R., et al. 2005. Structural characterization of proteins using residue environments. *Proteins* 61, 741–747.
- Nussinov, R., and Wolfson, H.J. 1991. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc. Natl. Acad. Sci. U.S.A.* 88, 10495–10499.
- Obenaus, J.C., Cantley, L.C., and Yaffe, M.B. 2003. Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. *Nucleic Acids Res.* 31, 3635–3641.
- Ondrechen, M.J., Clifton, J.G., and Ringe, D. 2001. THEMATIC: a simple computational predictor of enzyme function from structure. *Proc. Natl. Acad. Sci. U.S.A.* 98, 12473–12478.
- Pal, D., and Eisenberg, D. 2005. Inference of protein function from protein structure. *Structure* 13, 121–130.
- Pazos, F., and Sternberg, M.J. 2004. Automated prediction of protein function and detection of functional sites from structure. *Proc. Natl. Acad. Sci. U.S.A.* 101, 14754–14759.
- Platt, J.C. 1999. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, 61–74. In Smola, A.J., Bartlett, P., Scholkopf, B. and Schuurmans, D., eds. *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.
- Pollastri, G., Baldi, P., Fariselli, P., et al. 2002. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins* 47, 142–153.
- Porter, C.T., Bartlett, G.J. and Thornton, J.M. 2004. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res.* 32, D129–D133.
- Przulj, N. 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics* 23, e177–e183.
- Przulj, N., Corneil, D.G., and Jurisica, I. 2004. Modeling interactome: scale-free or geometric? *Bioinformatics* 20, 3508–3515.
- Quirk, P.G., Patchell, V.B., Colyer, J., et al. 1996. Conformational effects of serine phosphorylation in phospholamban peptides. *Eur. J. Biochem.* 236, 85–91.
- Ralaivola, L., Swamidass, S.J., Saigo, H., et al. 2005. Graph kernels for chemical informatics. *Neural Netw.* 18, 1093–1110.
- Reva, B., Antipin, Y., and Sander, C. 2007. Determinants of protein function revealed by combinatorial entropy optimization. *Genome Biol.* 8, R232.

- Rieck, K., and Laskov, P. 2008. Linear-time computation of similarity measures for sequential data. *J. Mach. Learn. Res.* 9, 23–48.
- Rost, B., Liu, J., Nair, R., et al. 2003. Automatic prediction of protein function. *Cell Mol. Life Sci.* 60, 2637–2650.
- Rush, J., Moritz, A., Lee, K.A., et al. 2005. Immunoaffinity profiling of tyrosine phosphorylation in cancer cells. *Nat. Biotechnol.* 23, 94–101.
- Russell, R.B. 1998. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *J. Mol. Biol.* 279, 1211–1227.
- Scholkopf, B., Tsuda, K., and Vert, J.-P., eds. 2004. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA.
- Shen, T., Zong, C., Hamelberg, D., et al. 2005. The folding energy landscape and phosphorylation: modeling the conformational switch of the NFAT regulatory domain. *FASEB J.* 19, 1389–1395.
- Song, L., Kalyanaraman, C., Fedorov, A.A., et al. 2007. Prediction and assignment of function for a divergent *N*-succinyl amino acid racemase. *Nat. Chem. Biol.* 3, 486–491.
- Tholey, A., Pipkorn, R., Bossemeyer, D., et al. 2001. Influence of myristoylation, phosphorylation, and deamidation on the structural behavior of the *N*-terminus of the catalytic subunit of cAMP-dependent protein kinase. *Biochemistry* 40, 225–231.
- Wallace, A.C., Borkakoti, N., and Thornton, J.M. 1997. TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites. *Protein Sci.* 6, 2308–2323.
- Wallace, A.C., Laskowski, R.A., and Thornton, J.M. 1996. Derivation of 3D coordinate templates for searching structural databases: application to Ser-His-Asp catalytic triads in the serine proteinases and lipases. *Protein Sci.* 5, 1001–1013.
- Wangikar, P.P., Tendulkar, A.V., Ramya, S., et al. 2003. Functional sites in protein families uncovered via an objective and automated graph theoretic approach. *J. Mol. Biol.* 326, 955–978.
- Watson, J.D., Laskowski, R.A., and Thornton, J.M. 2005. Predicting protein function from sequence and structural data. *Curr. Opin. Struct. Biol.* 15, 275–284.
- Wei, L., and Altman, R.B. 1998. Recognizing protein binding sites using statistical descriptions of their 3D environments. *Pac. Symp. Biocomput.* 497–508.
- Wei, L., and Altman, R.B. 2003. Recognizing complex, asymmetric functional sites in protein structures using a Bayesian scoring function. *J. Bioinform. Comput. Biol.* 1, 119–138.
- Wolfson, H.J., and Rigoutsos, I. 1997. Geometric hashing: an overview. *IEEE Comput. Sci. Eng.* 4, 10–21.
- Zvelebil, M.J., and Sternberg, M.J. 1988. Analysis and prediction of the location of catalytic residues in enzymes. *Protein Eng.* 2, 127–138.

Address correspondence to:

*Dr. Predrag Radivojac*  
*School of Informatics and Computing*  
*Indiana University*  
*Bloomington, IN 47408*

*E-mail: predrag@indiana.edu*

**and**

*Dr. Stefano Lonardi*  
*Department of Computer Science and Engineering*  
*University of California*  
*Riverside, CA 92521*

*E-mail: stelo@cs.ucr.edu*