Graphlet Kernels for Prediction of Functional Residues in Protein Structures

Vladimir Vacic[1†], Lilia M. Iakoucheva[2], Stefano Lonardi[1,*], Predrag Radivojac[3,*]

1) Department of Computer Science and Engineering, University of California, Riverside CA

2) Laboratory of Statistical Genetics, The Rockefeller University, New York NY

3) School of Informatics, Indiana University, Bloomington IN

* To whom correspondence should be addressed. E-mail: stelo@cs.ucr.edu; predrag@indiana.edu

† Current address: Cold Spring Harbor Laboratory, Cold Spring Harbor NY

Abstract

In this study 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 protein contact graph is then represented as a vector of counts of labeled non-isomorphic subgraphs (called 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. We compared the graphlet kernel approach against two alternative methods, a sequence-based predictor and our implementation of the FEATURE framework. On both function prediction tasks the graphlet kernel performed favorably compared to the alternatives; however, the margin of difference was considerably higher on the problem of phosphorylation site prediction. While there is both computational and experimental evidence 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 information related to surface accessibility alone nor the averaged measures calculated from the residue microenvironments utilized by FEATURE were sufficient to achieve high prediction accuracy. We believe that the key benefit of the proposed graphlet representation is its ability to capture similarity between local neighborhoods in protein structures via enumerating the patterns of local connectivity in the corresponding labeled graphs.

Introduction

With over 50,000 structures deposited in the Protein Data Bank (PDB) [1] and high-throughput efforts under way [2], functional characterization of proteins with known 3D structure is gaining importance in the global effort to understand structure-to-function determinants [3]. 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 [4,5]. 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, e.g. interface sites, hot spots, metal binding sites, catalytic sites or post-translationally modified residues [6]. At a higher level, computational methods can be used to establish connections between proteins and disease, typically via simulating protein folding pathways [7] or by using statistical inference techniques to predict gene-disease associations [8] or the effects of mutations [9].

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 [10] used spherical neighborhood to distinguish between metal binding and catalytic residues, while Gregory et al. [11] and Bagley and Altman [12] used concentric spheres to generate a score [11] or create a set of features [12] which 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 [13] and has been combined with localized molecular dynamics simulations to predict function [14]. Template-based methods, introduced in the 1990s [15,16,17,18], 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. Another template-based strategy, adopted from computer vision, is geometric hashing, in which a database of atoms or residue patterns is searched for similarities with the new structures [19,20,21]. 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, e.g. catalysis of the same reaction [22,23]. 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 [24,25]. Other authors addressed the problem via the framework of frequent subgraph mining [26,27,28], 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 [29,30,31,32].

Methods that predict function directly at the whole-molecule level [33,34] can in principle be combined with approaches that identify functional residues in the general sense [35,36,37,38,39] to achieve similar results. Finally, we note that structural-alignment algorithms (e.g. DALI [40]) 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 GO) 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 structure. 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 [41] and kernel-based statistical inference [42]. 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 [12,43] and S-BLEST [13] 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.

Materials and 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) 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* [41,44]. 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 which 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.

Graphlets and Automorphism Orbits

Graphlets are small non-isomorphic connected subgraphs [41,44] which 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.

FIGURE 1

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 (Figure 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 (Figure 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 Pržulj [44].

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 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 [34], or if a contact graph is constructed on the atom level, nodes may be labeled with the symbols of chemical elements [45].

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 graphs [46]. 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 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.

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 (see Figure 2 for the schematic representation of the counting algorithm). There is a total of deg(p) type g_1 graphlets, i.e. orbits o_1, where deg(.) denotes the degree of a vertex.

FIGURE 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 (Figure 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.

Counting 4-graphlets. There are four cases for counting 4-graphlets, namely 0111, 0112, 0122 and 0123 (Figure 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 which 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 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.

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 $\Phi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_m(x))$ and $\Phi(y) = (\varphi_1(y), \varphi_2(y), \ldots, \varphi_m(y))$ 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 semidefinite kernel matrix K, because each element of K is an inner product of vectors of counts [47]. In addition to the kernel $K(x, y)$, we also consider the *normalized kernel K'* defined as $K'(x, y) = K(x, y)/\sqrt{K(x, x) \cdot K(y, y)}$.

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 $(x_1 + x_2 + \cdots x_{|\mathcal{A}|})^2$. In general, multinomial expansion of a k-nomial raised to the n^{th} power, i.e. $(x_1 + x_2 + \cdots x_k)^n$, 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, e.g. 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 3 instead of 20. Alternatively, a separate predictor can be trained on each residue, which effectively reduces the dimensionality of the representation to $\dim\{\Phi(x)\} = 53{,}121$.

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, i.e. 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 Figure 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 subtries for which the prefix leading to the subtrie 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 [48].

Classification

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

$$score(q) = \sum_{i} \alpha_i \cdot d_i \cdot K(x_i, q)$$

where x_i is the i^{th} support vector coming from the training set, $d_i \in \{+1, -1\}$ is the class label of x_i, and α_i is the i^{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 [49].

Performance Evaluation

A prototype implementation of the graphlet kernel was coded in C++, using SVM^{light} [50] as the prediction engine. It was compared against a sequence-based predictor and our implementation of the FEATURE method [12,43].

Sequence-based predictor. The sequence-based predictor builds a model similar to DisPhos 1.3 [51], 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 ±12 positions. A support vector machine with a linear kernel was used as the learning model. We refer to the sequence-based predictor as SEQUENCE.

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 in [12,43]: (0, 1.875], (1.875, 3.75], (3.75, 5.625], and (5.625, 7.5]Å; (2) FEATURE$_{6\text{-}12\text{-}18}$ – based on radial intervals (0, 6], (6, 12], and (12, 18]Å, and (3) FEATURE$_{18}$ – based on a single sphere of radius of 18Å. FEATURE$_{6\text{-}12\text{-}18}$ representation was chosen to mimic our construction of the protein contact graph (with C_α-C_α distance of 6Å) and the level-3 neighborhood considered by the graphlet kernel, while FEATURE$_{18}$ was selected to quantify the difference between the cases of one sphere of radius 18Å and 3 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^{light} package.

The two predictors were chosen in order to provide fair and useful comparisons between methods, e.g. 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.

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 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.

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^{light}) 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, twenty distinct models were built for the CSA data set and three models for PHOS.

Data Sets

CSA. We selected all catalytic residues from the Catalytic Site Atlas (CSA) v.2.2.9 [52] found in the ASTRAL 40 v.1.73 structures [53], as positive examples. Catalytic activity in CSA has been assigned either experimentally or via function transfer, using PSI-BLAST [54]. 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 detailed description of the data sets, see Tables 1-2.

TABLE 1

TABLE 2

PHOS. It has previously been shown that protein phosphorylation sites are preferentially located in intrinsically disordered protein regions [51,55]. There are, however, examples where phosphorylation sites can also be found in ordered regions [51,56]. 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 [56,57,58,59], leading to mischaracterization of disordered regions. In order to investigate the structural properties of phosphorylation sites in greater detail, we searched PDB for records which contain keywords ”phosphoserine,” ”phosphothreonine,” or ”phosphortyrosine,” 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 (Suppl. Data). 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 [54]. This set of sites was compiled from the proteins annotated in UniProt release 54.3 [60], Phospho.ELM [61], Phosida [62], dbPTM [63], and through a survey of the literature [64,65,66,67,68] (see Table S2, Suppl. Data). We included only alignments longer than fifty 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.

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_α or C_β atoms, with appropriately chosen thresholds (e.g., 8.5Å [69], or in the 3-6Å range [70]). 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 [71] or the sum of their van der Waals radii plus 0.5Å [72]. Brinda et al. [73] proposed to consider the strength of interaction between the amino acids, defined as the normalized number of atom-atom pairs below a cutoff distance.

Pollastri et al. [74] analyzed protein coordination numbers, which correspond to degrees of vertices in our framework. They performed experiments with distances between C_α atoms with thresholds set at 6, 8, 10, and 12Å and reported high similarity between residues in contact for thresholds ≥8 Å. 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_α, C_β, 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 [75]. 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 (Suppl. Data).

The methods based on the distances between C_α and C_β 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 which cannot avoid an expensive square root operation. Based on our experiments C_α- and C_β-based methods display a relatively strong dependency on the threshold parameter, which is indicated by the low Jaccard coefficients between sets of edges (see Table S5, Suppl. Data). 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_α and C_β methods for 6Å threshold. These results have led us to choose the C_α-based method with 6Å distance cut-off to build the protein contact graph, as a good compromise between speed and sensitivity.

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 (Figure 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 the Figures S2-S24 and Tables S2-S3 (Suppl. Data).

FIGURE 3

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 2 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 $\text{FEATURE}_{6\text{-}12\text{-}18}$. The list of all results is provided in Tables S2-S3 and Figures S2-S24 (Suppl. Data). The reduced alphabets were created by hierarchical clustering of amino acids using BLOSUM50 scoring matrix as a measure of similarity (Figure S1, Suppl. Data).

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±7.3% (SEQUENCE), 76.7±7.8% (FEATURE), 76.7±5.2% (non-normalized graphlet kernel) and 78.8±5.9% (normalized graphlet kernel). On the PHOS data set the mean AUC values were 74.8±1.8% (SEQUENCE), 65.2±2.0% (FEATURE), 77.6±3.1% (non-normalized graphlet kernel) and 76.4±4.4% (normalized graphlet kernel). Figure 3 shows a bar plot with the AUC values corresponding to the CSA data set and 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.

FIGURE 4

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 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. 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 phsophosite Tyr394 (Figure 5A) along with its level-3 neighborhood in the protein contact graph (Figure 5B).

FIGURE 5

Structure of Ordered Protein Phosphorylation Sites

It was previously proposed that phosphorylation sites preferentially, although not exclusively, appear in intrinsically disordered protein regions [51]. 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 [55]. Nonetheless, there are examples of ordered phosphorylation sites [56,76,77,78] and several structures containing phosphorylated sites have been deposited in PDB (see Data Sets section). In addition, several studies addressed conformational changes in proteins following the addition of the phosphate group (e.g. [57,79,80]). In the supplementary material (Table S1) 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 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 DSSP [81], 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.

TABLE 3

Another interesting observation is the prevalence of kinases in the ordered subset of phosphosites from PDB (Table S1). 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.

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, e.g. a random-walk kernel [82], a cycle pattern kernel [83], weighted decomposition kernel [84] and others [45,85]. 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 [12,43,86]. 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, e.g. via a fusion kernel [87,88] or the hyper-kernel approach [34], 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 [89,90,91]. 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 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. [89] which was later further formalized and systematically evaluated by Rieck and Laskov [48].

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. [92] 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 [93]. In addition, many phosphorylation sites lie in the disordered regions for which a time-invariant contact map may not even exist [51]. Thus, the model by Blom et al., as well as many others [51,67,94,95,96,97], 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 [98].

In previous work, it was hypothesized that phosphorylation sites preferentially occur in intrinsically disordered regions [51]. This hypothesis has been validated in recent experimental studies

[55,99] and in several cases disorder-to-order transition upon phosphorylation has been predicted [58,100,101]. Furthermore, it has recently been shown that disordered proteins are substrates of twice as many kinases as are ordered proteins [99]. 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.

Acknowledgements

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 (PI: Iakoucheva), NSF IIS-0447773 (PI: Lonardi), NSF DBI-0321756 (co-PI: Lonardi), and NSF DBI-0644017 (PI: Radivojac).

Contributions

VV and PR conceptualized the framework, designed, and conducted the experiments. VV implemented the graphlet kernel. LMI contributed with the analysis and significance of structural preferences of phosphorylation sites. SL and PR supervised the project. All authors participated in writing.

Note

The graphlet kernel and the counting method have been originally proposed in: Vacic, Vladimir (2008), Computational methods for discovery of cellular regulatory mechanisms. Ph.D. dissertation, University of California, Riverside, United States -- California. Retrieved from Dissertations & Theses @ University of California database. (Publication No. AAT 3332641).

References

1. Berman H, Bhat TN, Bourne P, Feng Z, Gilliand G, et al. (2000) The protein data bank and the challenge of structural genomics. Nat Struct Biol (Structural Genomics Supplement) 7: 957-959.
2. Burley SK, Almo SC, Bonanno JB, Capel M, Chance MR, et al. (1999) Structural genomics: beyond the human genome project. Nat Genet 23: 151-157.

3. Laskowski RA, Thornton JM (2008) Understanding the molecular machinery of genetics through 3D structures. Nat Rev Genet 9: 141-151.

4. Watson JD, Laskowski RA, Thornton JM (2005) Predicting protein function from sequence and structural data. Curr Opin Struct Biol 15: 275-284.

5. Lee D, Redfern O, Orengo C (2007) Predicting protein function from sequence and structure. Nat Rev Mol Cell Biol 8: 995-1005.

6. Rost B, Liu J, Nair R, Wrzeszczynski KO, Ofran Y (2003) Automatic prediction of protein function. Cell Mol Life Sci 60: 2637-2650.

7. Dobson CM (2001) The structural basis of protein folding and its links with human disease. Philos Trans R Soc Lond B Biol Sci 356: 133-145.

8. Dalkilic MM, Costello JC, Clark WT, Radivojac P (2008) From protein-disease associations to disease informatics. Front Biosci 13: 3391-3407.

9. Mooney SD (2005) Bioinformatics approaches and resources for single nucleotide polymorphism functional analysis. Brief Bioinform 6: 44-56.

10. Zvelebil MJ, Sternberg MJ (1988) Analysis and prediction of the location of catalytic residues in enzymes. Protein Eng 2: 127-138.

11. Gregory DS, Martin AC, Cheetham JC, Rees AR (1993) The prediction and characterization of metal binding sites in proteins. Protein Eng 6: 29-35.

12. Bagley SC, Altman RB (1995) Characterizing the microenvironment surrounding protein sites. Protein Sci 4: 622-635.

13. Mooney SD, Liang MH, DeConde R, Altman RB (2005) Structural characterization of proteins using residue environments. Proteins 61: 741-747.

14. Glazer DS, Radmer RJ, Altman RB (2008) Combining molecular dynamics and machine learning to improve protein function recognition. Pac Symp Biocomput: 332-343.

15. Wallace AC, Laskowski RA, Thornton JM (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.

16. Russell RB (1998) Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. J Mol Biol 279: 1211-1227.

17. Fetrow JS, 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.

18. Kleywegt GJ (1999) Recognition of spatial motifs in protein structures. J Mol Biol 285: 1887-1897.

19. Nussinov R, Wolfson HJ (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.

20. Wallace AC, Borkakoti N, Thornton JM (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.

21. Wolfson HJ, Rigoutsos I (1997) Geometric hashing: an overview. IEEE Computational Science & Engineering 4: 10-21.

22. Hermann JC, Marti-Arbona R, Fedorov AA, Fedorov E, Almo SC, et al. (2007) Structure-based activity prediction for an enzyme of unknown function. Nature 448: 775-779.

23. Song L, Kalyanaraman C, Fedorov AA, Fedorov EV, Glasner ME, et al. (2007) Prediction and assignment of function for a divergent N-succinyl amino acid racemase. Nat Chem Biol 3: 486-491.

24. Grindley HM, Artymiuk PJ, Rice DW, Willett P (1993) Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. J Mol Biol 229: 707-721.

25. Artymiuk PJ, Poirrette AR, Grindley HM, Rice DW, Willett P (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.
26. Wangikar PP, Tendulkar AV, Ramya S, Mali DN, Sarawagi S (2003) Functional sites in protein families uncovered via an objective and automated graph theoretic approach. J Mol Biol 326: 955-978.
27. Huan J, Bandyopadhyay D, Wang W, Snoeyink J, Prins J, et al. (2005) Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. J Comput Biol 12: 657-671.
28. Bandyopadhyay D, Huan J, Liu J, Prins J, Snoeyink J, et al. (2006) Structure-based function inference using protein family-specific fingerprints. Protein Sci 15: 1537-1543.
29. Liang MP, Banatao DR, Klein TE, Brutlag DL, Altman RB (2003) WebFEATURE: An interactive web tool for identifying and visualizing functional sites on macromolecular structures. Nucleic Acids Res 31: 3324-3327.
30. Laskowski RA, Watson JD, Thornton JM (2005) ProFunc: a server for predicting protein function from 3D structure. Nucleic Acids Res 33: W89-93.
31. Laskowski RA, Watson JD, Thornton JM (2005) Protein function prediction using local 3D templates. J Mol Biol 351: 614-626.
32. Pal D, Eisenberg D (2005) Inference of protein function from protein structure. Structure 13: 121-130.
33. Pazos F, Sternberg MJ (2004) Automated prediction of protein function and detection of functional sites from structure. Proc Natl Acad Sci U S A 101: 14754-14759.
34. Borgwardt KM, Ong CS, Schonauer S, Vishwanathan SV, Smola AJ, et al. (2005) Protein function prediction via graph kernels. Bioinformatics 21 Suppl 1: i47-56.
35. Elcock AH (2001) Prediction of functionally important residues based solely on the computed energetics of protein structure. J Mol Biol 312: 885-896.
36. Ondrechen MJ, Clifton JG, Ringe D (2001) THEMATICS: a simple computational predictor of enzyme function from structure. Proc Natl Acad Sci U S A 98: 12473-12478.
37. Kalinina OV, Mironov AA, Gelfand MS, Rakhmaninova AB (2004) Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. Protein Sci 13: 443-456.
38. Glaser F, Morris RJ, Najmanovich RJ, Laskowski RA, Thornton JM (2006) A method for localizing ligand binding pockets in protein structures. Proteins 62: 479-488.
39. Reva B, Antipin Y, Sander C (2007) Determinants of protein function revealed by combinatorial entropy optimization. Genome Biol 8: R232.
40. Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. J Mol Biol 233: 123-138.
41. Przulj N, Corneil DG, Jurisica I (2004) Modeling interactome: scale-free or geometric? Bioinformatics 20: 3508-3515.
42. Scholkopf B, Tsuda K, Vert J-P, editors (2004) Kernel methods in computational biology. Cambridge, MA: MIT Press.
43. Wei L, Altman RB (1998) Recognizing protein binding sites using statistical descriptions of their 3D environments. Pac Symp Biocomput: 497-508.
44. Przulj N (2007) Biological network comparison using graphlet degree distribution. Bioinformatics 23: e177-183.
45. Ralaivola L, Swamidass SJ, Saigo H, Baldi P (2005) Graph kernels for chemical informatics. Neural Netw 18: 1093-1110.
46. Atilgan AR, Akan P, Baysal C (2004) Small-world communication of residues and significance for protein dynamics. Biophys J 86: 85-91.

47. Haussler D (1999) Convolution kernels on discrete structures. Technical Report UCSCCRL-99-10, University of California at Santa Cruz.
48. Rieck K, Laskov P (2008) Linear-time computation of similarity measures for sequential data. Journal of Machine Learning Research 9: 23-48.
49. Platt JC (1999) Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Smola AJ, Bartlett P, Scholkopf B, Schuurmans D, editors. Advances in large margin classifiers: MIT Press. pp. 61-74.
50. Joachims T (2002) Learning to classify text using support vector machines: methods, theory, and algorithms: Kluwer Academic Publishers.
51. Iakoucheva LM, Radivojac P, Brown CJ, O'Connor TR, Sikes JG, et al. (2004) The importance of intrinsic disorder for protein phosphorylation. Nucleic Acids Research 32: 1037-1049.
52. Porter CT, Bartlett GJ, Thornton JM (2004) The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. Nucleic Acids Res 32 Database issue: D129-133.
53. Chandonia JM, Hon G, Walker NS, Lo Conte L, Koehl P, et al. (2004) The ASTRAL Compendium in 2004. Nucleic Acids Res 32: D189-192.
54. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25: 3389-3402.
55. Collins MO, Yu L, Campuzano I, Grant SG, Choudhary JS (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.
56. Johnson LN, Lewis RJ (2001) Structural basis for control by phosphorylation. Chem Rev 101: 2209-2242.
57. Shen T, Zong C, Hamelberg D, McCammon JA, Wolynes PG (2005) The folding energy landscape and phosphorylation: modeling the conformational switch of the NFAT regulatory domain. FASEB J 19: 1389-1395.
58. Espinoza-Fonseca LM, Kast D, Thomas DD (2007) Molecular dynamics simulations reveal a disorder-to-order transition on phosphorylation of smooth muscle myosin. Biophys J 93: 2083-2090.
59. Espinoza-Fonseca LM, Kast D, Thomas DD (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.
60. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, et al. (2005) The Universal Protein Resource (UniProt). Nucleic Acids Res 33 Database Issue: D154-159.
61. Diella F, Cameron S, Gemund C, Linding R, Via A, et al. (2004) Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins. BMC Bioinformatics 5: 79.
62. Gnad F, Ren S, Cox J, Olsen JV, Macek B, et al. (2007) PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites. Genome Biol 8: R250.
63. Lee TY, Huang HD, Hung JH, Huang HY, Yang YS, et al. (2006) dbPTM: an information repository of protein post-translational modification. Nucleic Acids Res 34: D622-627.
64. Ficarro SB, McCleland ML, Stukenberg PT, Burke DJ, Ross MM, et al. (2002) Phosphoproteome analysis by mass spectrometry and its application to *Saccharomyces cerevisiae*. Nat Biotechnol 20: 301-305.
65. Ballif BA, Villen J, Beausoleil SA, Schwartz D, Gygi SP (2004) Phosphoproteomic analysis of the developing mouse brain. Mol Cell Proteomics 3: 1093-1101.
66. Beausoleil SA, Jedrychowski M, Schwartz D, Elias JE, Villen J, et al. (2004) Large-scale characterization of HeLa cell nuclear phosphoproteins. Proc Natl Acad Sci U S A 101: 12130-12135.

67. Fujii K, Zhu G, Liu Y, Hallam J, Chen L, et al. (2004) Kinase peptide specificity: improved determination and relevance to protein phosphorylation. Proc Natl Acad Sci U S A 101: 13744-13749.
68. Rush J, Moritz A, Lee KA, Guo A, Goss VL, et al. (2005) Immunoaffinity profiling of tyrosine phosphorylation in cancer cells. Nat Biotechnol 23: 94-101.
69. Dokholyan NV, Li L, Ding F, Shakhnovich EI (2002) Topological determinants of protein folding. Proc Natl Acad Sci U S A 99: 8637-8641.
70. Hu Z, Bowen D, Southerland WM, del Sol A, Pan Y, et al. (2007) Ligand binding and circular permutation modify residue interaction network in DHFR. PLoS Comput Biol 3: e117.
71. Greene LH, Higman VA (2003) Uncovering network systems within protein structures. J Mol Biol 334: 781-791.
72. Keskin O, Nussinov R (2007) Similar binding sites and different partners: implications to shared proteins in cellular pathways. Structure 15: 341-354.
73. Brinda KV, Kannan N, Vishveshwara S (2002) Analysis of homodimeric protein interfaces by graph-spectral methods. Protein Eng 15: 265-277.
74. Pollastri G, Baldi P, Fariselli P, Casadio R (2002) Prediction of coordination number and relative solvent accessibility in proteins. Proteins 47: 142-153.
75. Hobohm U, Sander C (1994) Enlarged representative set of protein structures. Protein Sci 3: 522-524.
76. Keane NE, Chavanieu A, Quirk PG, Evans JS, Levine BA, et al. (1994) Structural determinants of substrate selection by the human insulin-receptor protein-tyrosine kinase. Eur J Biochem 226: 525-536.
77. Quirk PG, Patchell VB, Colyer J, Drago GA, Gao Y (1996) Conformational effects of serine phosphorylation in phospholamban peptides. Eur J Biochem 236: 85-91.
78. Tholey A, Pipkorn R, Bossemeyer D, Kinzel V, Reed J (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.
79. Groban ES, Narayanan A, Jacobson MP (2006) Conformational changes in protein loops and helices induced by post-translational phosphorylation. PLoS Comput Biol 2: e32.
80. Latzer J, Shen T, Wolynes PG (2008) Conformational switching upon phosphorylation: a predictive framework based on energy landscape principles. Biochemistry 47: 2110-2122.
81. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22: 2577-2637.
82. Gaertner T, Flatch P, Wrobel S. On graph kernels: hardness results and efficient alternatives; 2003. pp. 129–143.
83. Horvath T, Gaertner T, Wrobel S (2004) Cyclic pattern kernels for predictive graph mining. Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining. Seattle, WA. pp. 158-167.
84. Menchetti S, Costa F, Frasconi P (2005) Weighted decomposition kernels. Proceedings of the 22nd International Conference on Machine Learning: ACM Press. pp. 585-592.
85. Kashima H, Tsuda K, Inokuchi A (2003) Marginalized kernels between labeled graphs. Proceedings of the 20th International Conference on Machine Learning: AAAI Press. pp. 321-328.
86. Wei L, Altman RB (2003) Recognizing complex, asymmetric functional sites in protein structures using a Bayesian scoring function. J Bioinform Comput Biol 1: 119-138.
87. Lanckriet GR, De Bie T, Cristianini N, Jordan MI, Noble WS (2004) A statistical framework for genomic data fusion. Bioinformatics 20: 2626-2635.
88. Lanckriet GR, Deng M, Cristianini N, Jordan MI, Noble WS (2004) Kernel-based data fusion and its application to protein function prediction in yeast. Pac Symp Biocomput: 300-311.
89. Leslie C, Eskin E, Noble WS (2002) The spectrum kernel: a string kernel for SVM protein classification. Pac Symp Biocomput: 564-575.

90. Leslie C, Kuang R (2004) Fast string kernels using inexact matching for protein sequences J Mach Learn Res 5: 1435-1455.
91. Kuang R, Ie E, Wang K, Wang K, Siddiqi M, et al. (2005) Profile-based string kernels for remote homology detection and motif extraction. J Bioinform Comput Biol 3: 527-550.
92. Blom N, Gammeltoft S, Brunak S (1999) Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. J Mol Biol 294: 1351-1362.
93. Izarzugaza JM, Grana O, Tress ML, Valencia A, Clarke ND (2007) Assessment of intramolecular contact predictions for CASP7. Proteins 69 Suppl 8: 152-158.
94. Obenauer JC, Cantley LC, Yaffe MB (2003) Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic Acids Research 31: 3635-3641.
95. Brinkworth RI, Breinl RA, 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.
96. Hjerrild M, Stensballe A, Rasmussen TE, Kofoed CB, Blom N, 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.
97. Kim JH, Lee J, Oh B, Kimm K, Koh I (2004) Prediction of phosphorylation sites using SVMs. Bioinformatics 20: 3179-3184.
98. Kopp J, Bordoli L, Battey JN, Kiefer F, Schwede T (2007) Assessment of CASP7 predictions for template-based modeling targets. Proteins 69 Suppl 8: 38-56.
99. Gsponer J, Futschik ME, Teichmann SA, Babu MM (2008) Tight regulation of unstructured proteins: from transcript synthesis to protein degradation. Science 322: 1365-1368.
100. Hamelberg D, Shen T, McCammon JA (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.
101. Hegedus T, Serohijos AW, Dokholyan NV, He L, Riordan JR (2008) Computational studies reveal phosphorylation-dependent changes in the unstructured R domain of CFTR. J Mol Biol 378: 1052-1063.

Figure Captions

Figure 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.

Figure 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.

Figure 3: Area under the ROC curve (AUC) for the CSA data sets. Method compared: S - SEQUENCE, F - FEATURE$_{6\text{-}12\text{-}18}$, 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.

Figure 4: ROC plots for the PHOS data sets. Red curve - FEATURE$_{6\text{-}12\text{-}18}$, black curve - SEQUENCE, green curve - graphlet kernel with full alphabet, blue curve - normalized graphlet kernel with full alphabet.

Figure 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).

Tables

Table 1: Summary of the CSA data set.

Residue	**Sites**	**Chains**	**Non-sites**	**NS/S Ratio**	**Total**
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
Total	6,830	2,025	94,633	13.86	101,463

Table 2: Summary of the PHOS data set.

Residue	**Sites**	**Chains**	**Non-sites**	**NS/S Ratio**	**Total**
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
Total	1,157	686	8,718	7.54	9,875

Table 3: Secondary structure assignments of the phosphorylation sites found in PDB: number of sites and percentage.

	Helix	**Sheet**	**Loop**	**Turn**
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%)
Total	15 (16.7%)	9 (10%)	46 (51.1%)	20 (22.2%)

Figure 1

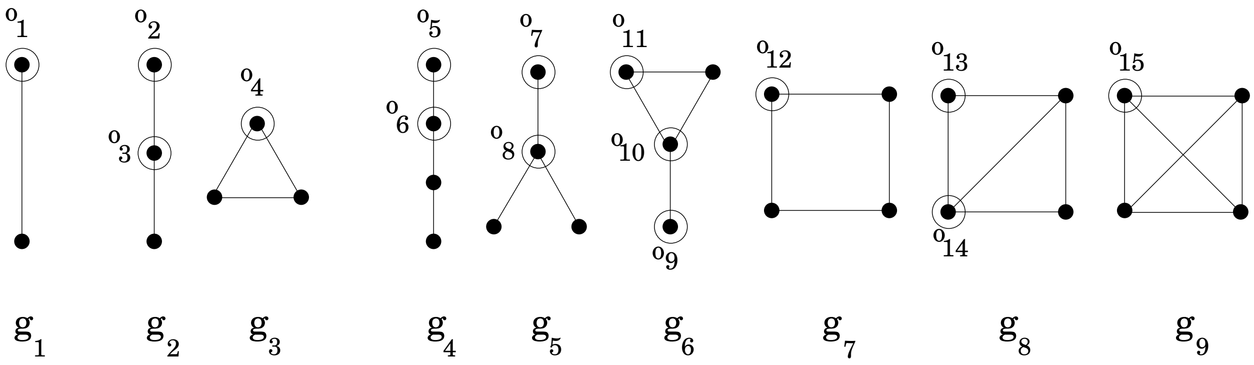

o_1
o_2
o_3
o_4
o_5
o_6
o_7
o_8
o_{11}
o_{10}
o_9
o_{12}
o_{13}
o_{14}
o_{15}
g_1
g_2
g_3
g_4
g_5
g_6
g_7
g_8
g_9

Figure 2

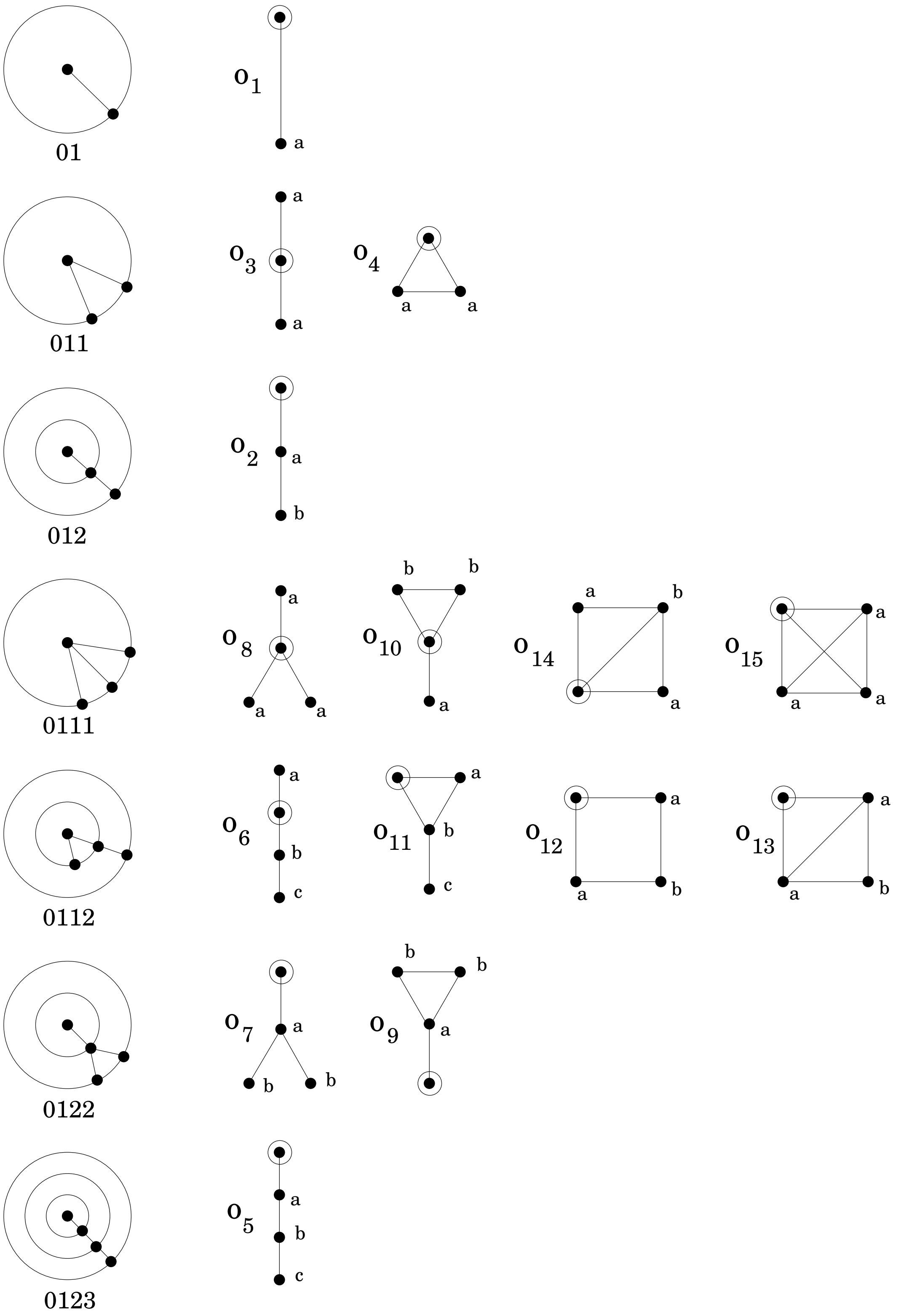

01
o1
a
011
o3
a
a
o4
a
a
012
o2
a
b
0111
o8
a
a
a
o10
b
b
a
o14
a
b
a
o15
a
a
a
0112
o6
a
b
c
o11
a
b
c
o12
a
a
b
o13
a
a
b
0122
o7
a
b
b
o9
b
b
a
0123
o5
a
b
c

Figure 3

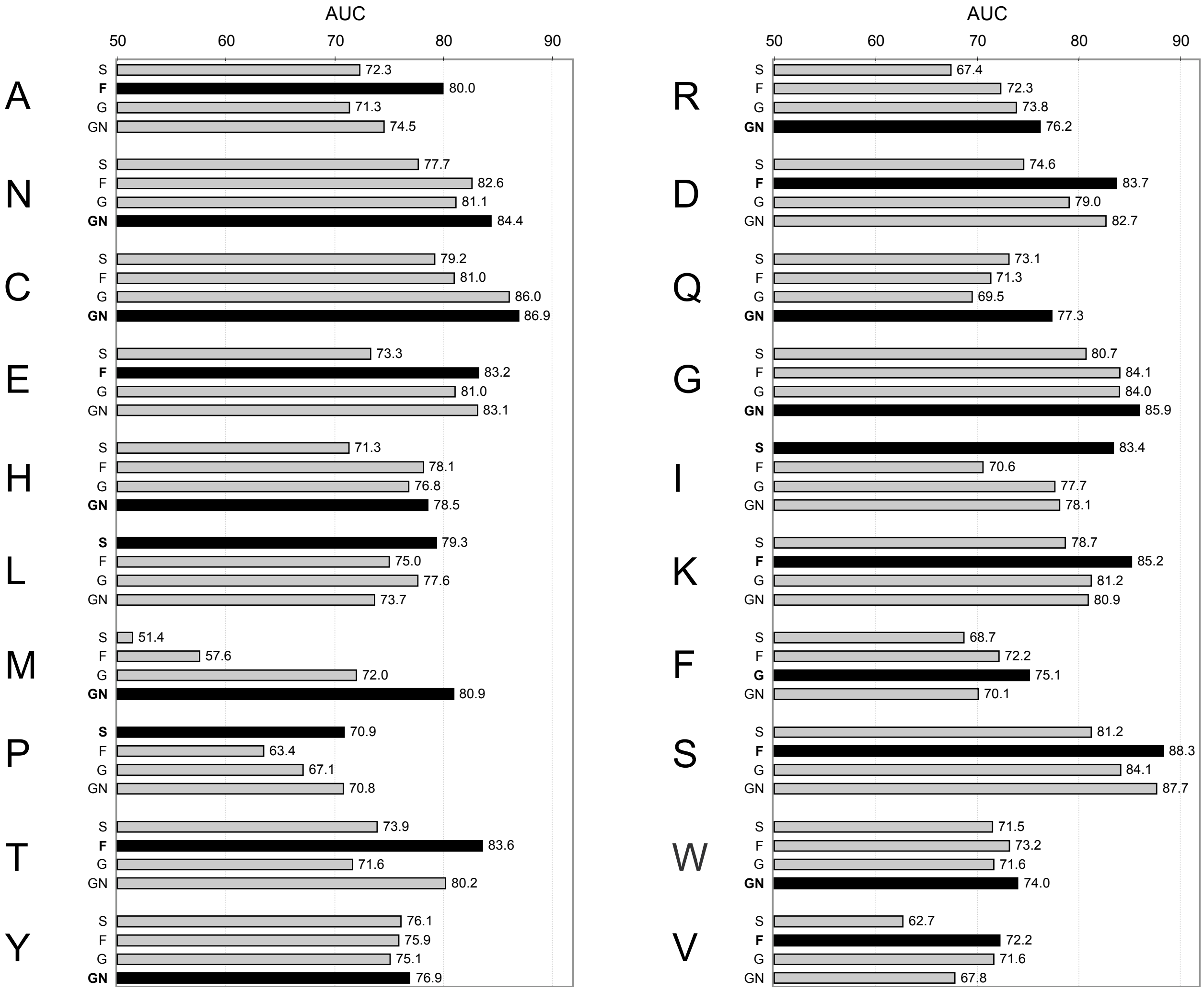

AUC
50
60
70
80
90
A
S 72.3
F 80.0
G 71.3
GN 74.5
N
S 77.7
F 82.6
G 81.1
GN 84.4
C
S 79.2
F 81.0
G 86.0
GN 86.9
E
S 73.3
F 83.2
G 81.0
GN 83.1
H
S 71.3
F 78.1
G 76.8
GN 78.5
L
S 79.3
F 75.0
G 77.6
GN 73.7
M
S 51.4
F 57.6
G 72.0
GN 80.9
P
S 70.9
F 63.4
G 67.1
GN 70.8
T
S 73.9
F 83.6
G 71.6
GN 80.2
Y
S 76.1
F 75.9
G 75.1
GN 76.9
AUC
50
60
70
80
90
R
S 67.4
F 72.3
G 73.8
GN 76.2
D
S 74.6
F 83.7
G 79.0
GN 82.7
Q
S 73.1
F 71.3
G 69.5
GN 77.3
G
S 80.7
F 84.1
G 84.0
GN 85.9
I
S 83.4
F 70.6
G 77.7
GN 78.1
K
S 78.7
F 85.2
G 81.2
GN 80.9
F
S 68.7
F 72.2
G 75.1
GN 70.1
S
S 81.2
F 88.3
G 84.1
GN 87.7
W
S 71.5
F 73.2
G 71.6
GN 74.0
V
S 62.7
F 72.2
G 71.6
GN 67.8

Figure 4

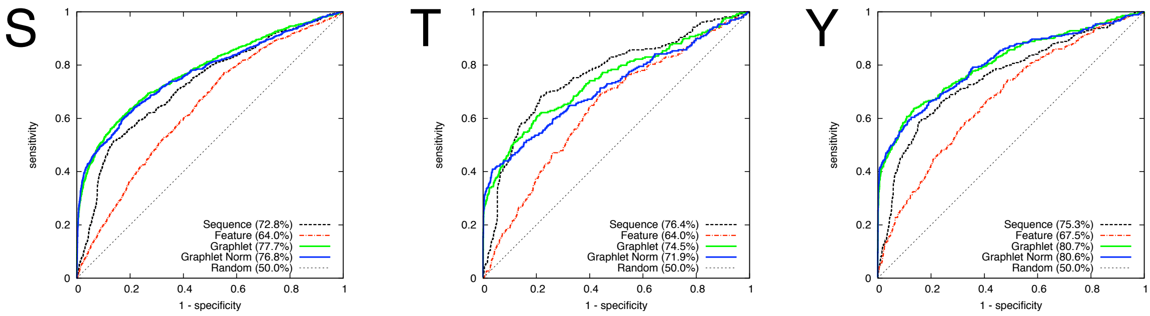
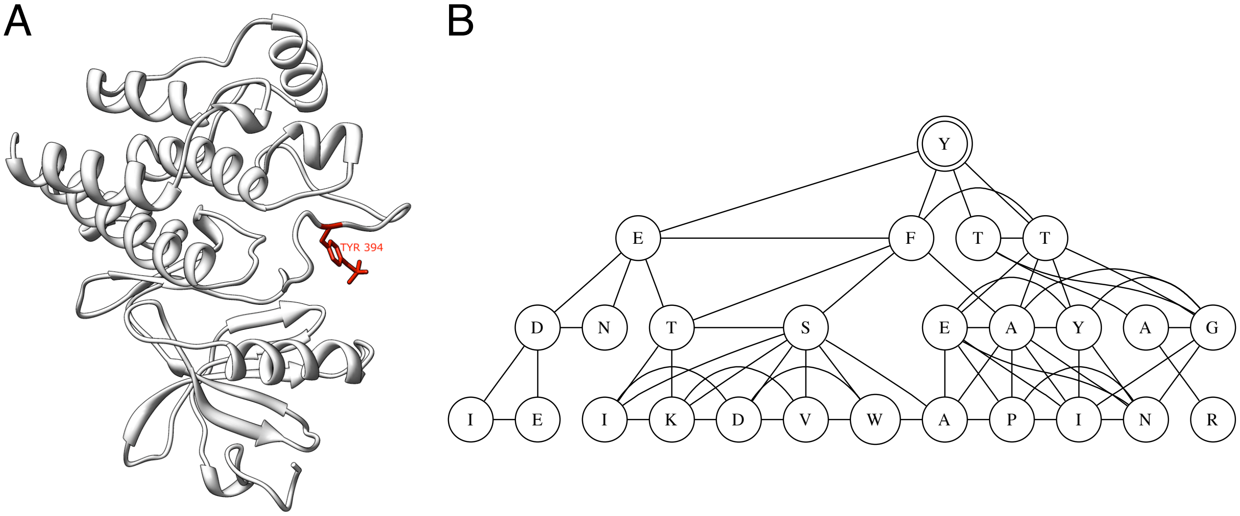

S
T
Y
sensitivity
1 - specificity
0
0.2
0.4
0.6
0.8
1
Sequence (72.8%)
Feature (64.0%)
Graphlet (77.7%)
Graphlet Norm (76.8%)
Random (50.0%)
Sequence (76.4%)
Feature (64.0%)
Graphlet (74.5%)
Graphlet Norm (71.9%)
Random (50.0%)
Sequence (75.3%)
Feature (67.5%)
Graphlet (80.7%)
Graphlet Norm (80.6%)
Random (50.0%)

Figure 5


A
TYR 394
B
Y
E F T T
D N T S E A Y A G
I E I K D V W A P I N R