J Mol Model (2006) 12: 453–461
DOI 10.1007/s00894-005-0070-2

ORIGINAL PAPER

**Dariusz Plewczynski · Adrian Tkacz ·
Lucjan Stanisław Wyrwicz · Adam Godzik ·
Andrzej Kloczkowski · Leszek Rychlewski**

Support-vector-machine classification of linear functional motifs in proteins

Received: 3 February 2005 / Accepted: 18 October 2005 / Published online: 10 December 2005

Abstract Our algorithm predicts short linear functional motifs in proteins using only sequence information. Statistical models for short linear functional motifs in proteins are built using the database of short sequence fragments taken from proteins in the current release of the Swiss-Prot database. Those segments are confirmed by experiments to have single-residue post-translational modification. The sensitivities of the classification for various types of short linear motifs are in the range of 70%. The query protein sequence is dissected into short overlapping fragments. All segments are represented as vectors. Each vector is then classified by a machine learning algorithm (Support Vector Machine) as potentially modifiable or not. The resulting list of plausible post-translational sites in the query protein is returned to the user. We also present a study of the human protein kinase C family as a biological application of our method.

Keywords Kinase substrate prediction · Profile–profile sequence similarity · Local structural segments · Linear functional motifs · Swiss-Prot database · Support vector machine (SVM)

Introduction

The rapid increase in genomic information requires new automatic techniques to learn protein functions, which are crucial for controlling intracellular processes. A function of a protein is partially determined by sequence motifs. In the case of the phosphorylation process, the location of post-translationally modified residues is largely determined by the primary sequence of the target site. Although many types of kinases are known, the identification of their potential biological targets is still incomplete. High substrate specificity of protein kinases ensures correct transmission of signals in cells, but we lack general, efficient and error-free tools for identification of functional motifs in proteins. Here we present a machine-learning approach to predicting various types of linear functional motifs in proteins. Our approach is based on the classification of biological functional information regarding post-translational modification sites in proteins acquired from the Swiss-Prot database. This classification is then used to predict new modification sites in proteins.

Most methods that predict *functional motifs* in proteins rely on multiple sequence alignments. Proteins can be grouped into limited numbers of families using similarities between their sequences. Protein domains or whole proteins belonging to one family share functional attributes, and are probably derived from a common ancestor. By studying conserved regions of protein sequences within a single family, one can derive a signature for such a family or domain. These signatures, which infer a function of a protein or its three-dimensional structure, distinguish mem-

D. Plewczynski · A. Tkacz · L. Rychlewski
BioInfoBank Institute,
Limanowskiego 24A/16,
60-744, Poznan, Poland

D. Plewczynski (✉)
Interdisciplinary Centre for Mathematical and
Computational Modeling, University of Warsaw,
Pawinskiego 5a Street,
02-106, Warsaw, Poland
e-mail: darman@icm.edu.pl
Tel.: +48-61-8653520
Fax: +48-61-8643350

L. S. Wyrwicz
Bioinformatics Unit, Department of Physics,
Adam Mickiewicz University,
ul.Umultowska 85,
61-614, Poznan, Poland

A. Godzik
Bioinformatics Core JCSG, University of California San Diego,
La Jolla, CA, USA

A. Godzik
The Burnham Institute,
La Jolla, CA, USA

A. Kloczkowski
Baker Center for Bioinformatics and Biological Statistics,
Iowa State University,
Ames, IO, USA

bers of a group from unrelated proteins. A protein signature can also be assigned to new proteins by formulating hypotheses about their function. Conserved motifs that represent conserved biochemical properties or biological functions can be used to identify divergent sequences with low overall sequence similarity. Homology, in such a case, can be detected even if the sequence similarity is low. One can also search for protein fingerprints that are defined as groups of conserved signatures. Such fingerprints encode protein folds and functional sites in more detail than single motifs. The PRINTS database [1] provides a compendium of such fingerprints based on the Swiss-Prot and the TrEMBL databases. eMOTIF [2, 3] discovers conserved sequence motifs in families of proteins with a wide range of specificities and sensitivities. The eMOTIFS database is derived from multiple sequence alignments from the BLOCKS+ database [4] and the PRINTS database. A hybrid database approach combines signature-recognition methods from different sources to scan a given query protein sequence against protein signatures. Such methods search specific databases with pre-configured cutoff thresholds. They return lists of hits in individual databases, and then these hit lists are combined. InterProScan [5] makes annotations based upon InterPro member databases. The PROSITE database [6] allows one to infer function and classification of proteins using a set of tools: ScanProsite [7], PRATT [8], PPSearch, PROSCAN and PPscan.

Linear functional motifs contain local sequence information around post-translational modification sites. A simple approach to retrieving this information is based on the application of regular expression searches. Regular expressions are built from experimentally verified functional sites known in proteins and reported in the scientific literature. The ELM server produces a large number of false positives. Regular expression searches have difficulty in describing linear functional motifs by a simple letter pattern. To improve the predictive efficiency and lower the number of false positives, context-based rules and logical filters (taking into account taxonomic range, cell compartment and globular organization) are applied in the ELM resources at http://elm.eu.org/ [9]. Our tool is more conservative, and can be used as an additional filter to remove some of the false positives. Another approach focuses on the statistical description of known instances, i.e. short linear protein-sequence motifs. This method is based on position-specific scoring matrices constructed from oriented peptide libraries, phage display or other experiments. These matrices of selectivity values provide relative scores of candidate functional motifs in evaluated protein sequences. ScanSite [10] identifies short protein-sequence motifs recognized by modular signaling domains, phosphorylated by kinases or mediating specific interactions with proteins or ligands. The current release of ScanSite (ver. 2.0) includes 62 motifs. Sulfinator [11] focuses only on the prediction of tyrosine sulfation sites in protein sequences. It uses Hidden Markov Models (built on the basis of multiple sequence alignments within sequence windows of 25 amino acids) to recognize sulfated residues. Other methods utilize various machine-learning approaches to characterize the neighborhood of a post-translational modification site. PhosphoBase provides a large collection of phosphorylated residues in proteins and information about peptide phosphorylation by protein kinases [12, 13]. The data have been collected from the literature and cover (version 2.0, 1998) 414 proteins with 1,052 phosphorylated residues. The functional motifs are built by statistical analysis of the sequence specificity of protein kinases. The analysis is conducted on 9-amino-acid segments around phosphorylation sites. Our tool uses the same 9-amino-acid sequence window around phosphorylation sites to predict annotation. This database is used as a training set for the NetPhos 2.0 server, which predicts serine, threonine and tyrosine phosphorylation sites for eukaryotic proteins by using neural networks [14]. Our service is naturally complementary to the NetPhos tool. It is based on a different machine-learning methodology (SVM). It builds an independent list of plausible phosphorylation sites for a given query protein. Both lists can be compared and used to develop a consensus prediction method, which usually improves the accuracy of prediction. A detailed comparison with this tool, SVM results from the PhosphoBase database, and details of the consensus algorithm with sensitivity/specificity scores will be presented in our next paper.

In our previous paper [15] we developed a library of local structural segments and a profile–profile matching algorithm that predicts local structure of proteins from their sequence information. The Fragments Library prediction method server (FRAGlib, publicly available at http://ffas.ljcrf.edu/Servers/frag.html) allows prediction of local structural conformations of sequence segments around phosphorylated sites. The algorithm has been applied successfully to the characterization of local structure around phosphorylation sites in proteins [16, 17]. Our results strongly suggest that sequence information is the most crucial for successful prediction of phosphorylation sites in proteins. It can be supplemented by additional structural-context information (predicted by our segment similarity method). Only proteins phosphorylated by PKA and PKC kinases, which represent the largest number of instances in the Swiss-Prot database, can be used as the benchmark and the test dataset for our automatic annotation method. The structural counterpart of prediction is evaluated using the database of real (experimentally confirmed) structures, focusing on parts of the main C^{α} chains around phosphorylation sites. These structures are collected using the PSI-Blast server running on the PDB database (PDB-Blast) (http://www.bioinfo.pl/).

In the Materials and methods section we provide detailed information about the preparation of the database of short sequence fragments annotated to undergo post-translational modification. The Methods section describes the automatic annotation algorithm for prediction of post-translational modification sites in proteins. In the Results section we present benchmarks used for statistical analysis of quality of the automatic annotation service. We present also a study of the human protein kinase C family as an example. Finally, we present conclusions and discuss possible future improvements of our service.

Materials and methods

The database of short linear functional motifs in proteins

Our method of predicting plausible post-translational modification sites is based on classification of known experimental occurrences. We use sequence information as an input, because in most cases only the sequence of a potential target protein is known. Biological information is acquired from the *Swiss-Prot database* [18]. All training cases for our algorithm represent proteins with experimentally annotated biological function contained in this database. For initial tests we have selected proteins with acetylation, phosphorylation (by PKA, PKC, CK, CK2 and CDC2 kinases), sulfation, amidation, hydroxylation, methylation, and pyrrolidone and γ-carboxyglutamic modification sites. These types of biological processes have the largest number of known experimental instances, which is crucial for building classification models.

We focus our attention only on those proteins in which single residues are annotated to perform a specific function. For each type of post-translational modification, the list of proteins with at least one site annotated to undergo the particular modification is fetched from the Swiss-Prot database. In order to maximize the classification accuracy of models, we neglect all sites annotated “by similarity”, “partial”, “potential”, “probable” or “predicted”. The remaining sites are used to create the dataset of positive cases, which includes all sequence segments dissected from parent proteins with a length of nine amino acids (*positive instances*). All sequence segments are centered on the annotated residue. If a segment is near a protein end, missing positions are substituted by '*X*', so the central location of annotated residues in all segments is preserved. All redundant segments, having the same sequence, are removed from the dataset. The length of a linear motif (nine residues) is optimized to ensure the maximum performance of the method for all types of linear functional motifs. It is possible that certain types of motifs would benefit from a longer length that would be more specific. To sample the background preferences for each position in these short sequence segments, we build for each type of post-translational modification a dataset of negative cases (*negative instances*). The list of proteins is again fetched from the Swiss-Prot database. From this list we randomly choose a large number of short sequence fragments (appropriate to the central amino acid to be modified) that are not annotated to undergo this particular modification. Those two datasets (positive and negative instances) for each type of functional motif are then used in training of the SVM.

In the case of phosphorylation by PKA and PKC kinases, we use 67 proteins with PKA phosphorylation (86 different sequence segments) and 49 proteins with PKC phosphorylation (56 different segments). Sequence segments with the proper central residue (according to the type of phosphorylation process) that are not annotated as functional, are used as negative cases. Here, in order to obtain background preferences for phosphorylation sites, we extract 14,353 PKA-negative and 14,369 PKC-negative sequence fragments with the correct central residue (S or T amino acids). These negative instances are chosen randomly from proteins in the Swiss-Prot database and annotated to have at least one phosphorylation site.

Methods

Local structure preferences of linear functional motifs

The next step in our analysis is to quantify local structural preferences around post-translational motification sites using protein databases. First we obtain experimental structures of proteins with the PDB-Blast server (http://www.bioinfo.pl/) developed by our group, in order to get possible PDB-deposited structures of proteins with post-translational motification sites. The number of collected structures for proteins with post-translational modifications is quite low, and is inadequate for training purposes any machine learning algorithm. In many cases even though coordinates of annotated proteins are available, coordinates for the actual modification sites are missing. Functional motifs are frequently located in unstructured parts of proteins.

In order to improve the structural statistics we have developed (FRAGlib server, publicly available at http://ffas.ljcrf.edu/Servers/frag.html) [15] a profile–profile matching algorithm that predicts local structures of short linear sequence segments. This algorithm has been applied successfully to the characterization of local structure around phosphorylation sites in proteins [16, 17]. Predicted local structures are in qualitative agreement with the real structures. A comparison with other available structure prediction tools like ROSETTA [19, 20] or HMMstr/I-sites [21] has been performed [17]. The difference between the results of those methods and our results (in modeling local structural preferences around phosphorylation sites) is within the accuracy of our method. Our results strongly suggest that sequence information is the most crucial for successful classification of functional linear motifs in proteins.

In the case of phosphorylation by PKA and PKC kinases, we collect structural models for 56 proteins with PKA and 38 with PKC phosphorylation sites. However, we found only 11 crystal structure segments around sites with both PKA and PKC phosphorylations. Most phosphorylation sites are located in unstructured parts of proteins, which are difficult to crystallize and frequently those coordinates are missing in the PDB. In order to obtain background preferences for sites with known structures, we also extract 340 PKA-negative and 141 PKC-negative sites from protein segments with assigned coordinates and correct central residues (S or T). We analyze the local structure composition of these positive and negative cases. While sequence compositions of both types of instances display clear differences, much less significant differences are observed between local structures [16]. The structural part of the prediction score helps in predictions, but the main difference between phosphorylation by PKA and PKC kinases was due to the sequence-related part of the score. This is the reason for using only sequence information and ignoring local structure information in our automatic predictor based on SVM statistical learning theory.

Sequence representation of linear functional motifs

In order to build a detailed sequence model, both datasets of segments (positives and negatives) for each type of post-translational modification are projected onto multi-dimensional space. Statistical learning theory is used to classify all cases and construct the boundary between positives and negatives. We used five different generic representations of a short protein sequence segment and five additional combinations of them.

The first representation (the binary one called here *BIN*) encodes each position of a segment into a long 20 dimensional vector of binary values 0 and 1. The value 1 is taken if the specific type of amino acid is present at this position in a segment and 0 otherwise. Such a representation in the case of a segment nine residues long has dimensionality 9*20=180. For example a single residue Tyr (Y) is represented here as a vector with coordinates [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] (assuming that the first position in the vector corresponds to tyrosine). Any vector representing a single segment has nine coordinates equal to 1, all remaining coordinates have 0 values.

The second representation (*BLOSUM*) is based on the BLOSUM62 matrix. Each position of a sequence segment is represented by a 20 dimensional vector of the substitution scores between the amino acid found in the projected segment at this position and all existing 20 amino acids. For example if Tyr is found at the first position in a segment we represent it with the appropriate Tyr column from the BLOSUM62 matrix. In the case of fragments nine residues long, we have a 180 dimensional representation built from nine columns of the substitution matrix for all amino acids in the projected segment.

The *LOOKUP* representation does not use a 20 dimensional vector for each residue in a segment but a scalar value equal to its normalized sequence preference. Normalized preferences are calculated separately for nine positions within a segment. For example, the normalization for Tyr in the first position of a segment is made by dividing the probability of finding Tyr at the first position in annotated segments (positives) by the probability of finding it in unannotated segments (negatives). This projection uses 9, dimensional vectors to represent segments 9 amino acids long.

The profile projection (PROF) uses the same normalized preferences for each amino acid in a segment but takes them as 20-dimensional vectors. It constructs a Position-Specific Scoring Matrix (PSSM) of length 9 from the positive examples of sites, and represents each residue in a 9-amino-acid segment by its scores against a motif. Each position is projected as a 20-dimensional vector of normalized preferences for all types of amino acids multiplied by the appropriate amino-acid column from the BLOSUM62 matrix. If we find Tyr in the segment, we multiply all amino-acid preferences by the Tyr column of the substitution matrix. All nine positions of a segment are represented in the 180 dimensional space.

The last generic representation (*SPARSE*) is similar to the binary one, but instead of each binary value 1 it takes the normalized preference for the type of amino acid found at a certain position of a segment. For all other amino acids we put into the vector.

We also have tested five additional combinations the above projections like BIN+LOOKUP, SPARSE+LOOKUP etc. These are built by representing a sequence fragment by both projections, using all dimensions from the first and the second representation and taking the Cartesian product of the two vectors. The resulting combined representation has additional information that may help to increase prediction accuracy.

Support-Vector-Machine classification models of linear functional motifs in proteins

We use a statistical learning approach to classify positive and negative datasets and construct the boundary between them for each type of post-translational modification. The classification of all known instances after embedding them into one abstract feature space is done within the support-vector-machine (SVM) framework [22–24]. A detailed description of the version of the method used, together with a list of references is available on our server's website. In order to extract relevant information from heterogeneous data, a SVM tries to separate the two sets of training vectors with an optimal hyperplane. The optimum is reached for a hyperplane that maximizes the separating margin between the two classes of the training vectors. A typical SVM method uses several hundred-thousands of training examples and many thousands of support vectors for large datasets of positive and negative instances. Our sequence-fragments database is highly sparse (only a small number of positives). The SVM approach even in our case of low numbers of available observations enables us to construct predictive models with great generalization power. SVMs seek globally optimized solutions and avoid over-fitting even for large dimensionality of the data, so large number of features (as in our binary *BIN* representation of sequence segments) are allowed.

The output of the training phase for each type of post-translational modification is a classification model. It consists of a set of D support vectors T_j and α_i, which are non-zero, positive real numbers that are obtained from the optimization procedure. For any projection T of the input space of segments *[x]* onto the representation space, all models are given in a form of the cost function:

$$f(T[x]) = \sum_{i=1}^{i=D} l_i \alpha_i K(\varphi\{T[x]\}, \varphi\{T_i\}), \qquad (1)$$

where $K(T,T_i)$ is a polynomial kernel function that defines the feature space, φ is a nonlinear mapping function, and l_i are the known *a priori* class labels for support vectors. We use $l_i=+1$ for positive cases and $l_i=-1$ for negative ones. The polynomial kernel function is a positive definite function $K(T,T_i) = (a\langle\varphi\{T\},\varphi\{T_i\}\rangle + c)^d$ reflecting the similarity between an input sample and the set of support vectors T_i. This type of kernel has been used extensively in bioinformatics [25–27].

The number of free parameters for this quadratic programming problem is equal to the number of all instances in the training dataset. The non-zero parameters α_i describe the strength of the particular *i*-th support vector in the decision function. The SVM chooses as support vectors those points that are the closest to the separating hyperplane. The mapping function φ need not be defined explicitly because in the kernel function only its inner product is used.

Single residue post-translational modification predictor

The methodology used by our post-translational modification-site predictor is as follows. It receives the sequence of a query protein as an input and it predicts the post-translational modification sites of a certain type. The server uses the SVM classification models for all types of short linear functional motifs described in the previous section. First it dissects a query protein into overlapping short segments of nine amino acids. For each segment $x_{j,}$ it assigns a label using the SVM model constructed according to its cost function (see Eq. (1)). Residues that have a score (the value of the cost function) larger than a given cutoff value *b* are annotated as plausible modification sites. These points representing sequence segments lay in a region classified as positive by the SVM model's hyperplane with *b* given as the margin value. For purposes of our Web

Table 1 Recall values of the SVM training with polynomial kernel ((*sa***b*+*c*)^*d*) for all types of considered short linear functional motifs

Recall precision	Number of positives/ negatives	BIN (%)	BIN+ LOOKUP (%)	SPARSE (%)	SPRASE+ LOOKUP (%)	BLOSUM+ LOOKUP (%)	LOOKUP (%)	BLOSUM+ SUM_PROF (%)	SUM_PROF (%)	PROF (%)	PROF+ LOOKUP (%)
PKA	86/14353	12	43	36.1	37.2	41.9	41.9	39.5	37.2	41.9	41.9
PKC	56/14368	2	16	14.3	14.3	17.9	0	0	0	17.9	17.9
CDC2	41/14375	0	29	22.0	24.4	24.4	22.0	0	0	9.8	17.1
SULF	83/6426	40	40	38.6	39.8	47.0	38.6	13.3	7.2	48.2	57.8
CK2	62/11746	0	18	19.4	21.0	12.9	14.5	0	0	11.3	12.9
CK	85/11739	0	11	11.8	12.9	8.2	5.9	0	0	9.4	9.4
ACETY	552/10014	90	84	87.7	84.4	87.7	90.0	11.6	0	10.0	87.7
METHY	215/10000	31	32.1	35.4	32.6	33.0	17.7	0	0	34.0	36.3
HYDRO	363/10000	69	60.9	57.9	66.4	68.0	56.5	64.7	64.7	70.8	68.3
AMID	723/10000	96	58.8	58.9	58.8	58.0	58.5	39.0	37.8	51.3	58.8
PYRRO	390/10000	59	47.4	47.4	47.4	47.4	47.4	1.3	0	13.3	47.4
GAMMA	232/10000	59	45.7	53.9	47.0	58.6	31.5	21.6	15.1	47.8	47.8

We present results for all types of projections (columns) and post-translational modifications (rows) using the recall *R* value, which measures the percentage of correct predictions (the probability of correct predictions). *R* is computed using the leave-one-out procedure, which removes from the training data one sample, constructs the model on the basis of the remaining training dataset and then tests the prediction of the model on the removed sample. The resulting error estimators are averaged for all such models (for all positive and all negative instances)

Data are collected from the Swiss-Prot DB annotation tables (excluding "BY SIMILARITY", "PREDICTED", "PROBABLE", "POTENTIAL" or "PARTIAL" annotations). *Blue* color denotes the best results, *purple* the second best. The worst results (no trained model) are marked in *brown*, the non-zero lowest results in red. Recall equals to 0% and precision is not well defined if the SVM training cannot be finished. (for some types of linear functional motifs and some types of projections the training procedure fails). The most stable are: profile PROF+LOOKUP, SPARSE+LOOKUP or BLOSUM+LOOKUP methods. Other types of methods have lower efficiencies (recall/precision). The second column in the table gives the number of positives and negatives for each type of activation process. In the following columns we present results for 10 different methods for preparing SVM input vectors representing each segment (of length 9 amino acids). The first one (BIN) is the simplest. The BIN method uses binary representation of amino acids in the SVM input vector. The BIN+LOOKUP includes additional vectors of nine values (the size of segments) of frequency ratios between positives and negatives for particular amino acids found in the input segment and at each position in every predicted segment. The SPARSE method puts instead of 1 the value of frequency ratio between positives and negatives for the particular amino acids found in the input segment at each position of the predicted segment. The SPARSE+LOOKUP includes also the frequency ratios for segments. The LOOKUP vector uses only frequency ratios for amino acids found in a query segment. The BLOSUM+LOOKUP method rescales them with the BLOSUM62 coefficients. The SUM_PROF uses only the sum over the all frequency ratios (dot product), and the BLOSUM +SUM_PROF additionally employs BLOSUM62 similarity matrix. The last two methods use the entire frequency information calculated for both (positives and negatives) datasets with, or without separate LOOKUP information server, we use only the single most effective type of a kernel (a polynomial one). Our method is a simple one-vote-wins approach, where we annotate all segments with positive verification by at least one model.

The output page of our service contains two main parts. The first part is a detailed description of each scan type and post-translational modification pattern. For each SVM model, the server lists a number of positive and negative instances used in training and precision and recall errors calculated during the training phase. The second part of the output provides results of predictions for each model. It contains the parent protein information, a local segment sequence predicted at the modificated site, its position and its output score with values in the range [0.000–5.000]. The higher the output score, the greater the confidence of prediction.

Potential functional motifs are sometimes repeated when predicted by various methods (with different scores). Each method predicts somewhat different set of peptides as a phosphorylation functional motif. Our automatic predictor uses identity search or SVM scan. Users should analyze sequences using both methods in order to investigate a broader set of possibilities. A higher output score indicates a greater confidence of prediction. This means that potential segments are more similar to some of functional motifs used in the SVM training.

Results

The performance of SVM classification models for each type of linear functional motif is described by three measures of accuracy: classification error E, recall R and precision P:

$$\begin{aligned} \mathrm{E} &= \frac{\mathrm{fp} + \mathrm{fn}}{\mathrm{tp} + \mathrm{fp} + \mathrm{tn} + \mathrm{fn}} * 100\%, \\ R &= \frac{tp}{tp + fn} * 100\%, \\ P &= \frac{tp}{tp + fp} * 100\%. \end{aligned} \qquad (2)$$

The tp is the number of true positives, fp is the number of false positives, tn is the number of true negatives and fn is the number of false negatives. The classification error E provides an overall error measure; whereas recall R measures the percentage of correct predictions (the probability of correct prediction). Precision P gives the percentage of observed positives that are correctly predicted (the measure of reliability of the positive instances prediction). These measures of accuracy are calculated a using a precise but computationally intensive leave-one-out procedure. The leave-one-out test removes from the training data one sample, constructs the model on a basis of the remaining training dataset and then tests the prediction of the model on the removed sample. The resulting error estimators are averaged over all such models (for all positive and all negative instances). For the purpose of the leave-one-out test in the PROF/LOOKUP representation, we calculate each time the normalized sequence preferences for all types of amino acids for any position in the linear sequence fragment without using the removed samples in testing. (By doing this we avoid potential bias in the results).

We collect results for all projections of sequence fragments used separately for all types of post-translational

Table 2 Precision P of the SVM learning with polynomial kernel ((s a*b+c)^d) for all types of considered short linear functional motifs

Recall precision	Number of positives/ negatives	BIN (%)	BIN +LOOKUP (%)	SPARSE (%)	SPRASE +LOOKUP (%)	BLOSUM +LOOKUP (%)	LOOKUP (%)	BLOSUM +SUM_PROF (%)	SUM_PROF (%)	PROF (%)	PROF +LOOKUP (%)
PKA	86/14353	77	59	55.4	74.4	69.2	85.7	81.0	68.1	75.0	76.6
PKC	56/14368	100	43	44.4	40.0	90.9	-	-	-	83.3	62.5
CDC2	41/14375	-	32	23.7	33.3	28.6	69.2	-	-	20.0	28.0
SULF	83/6426	97	75	74.4	73.3	76.5	72.7	100	100	87.0	78.7
CK2	62/11746	-	48	44.4	39.4	50.0	100	-	-	53.9	53.3
CK	85/11739	-	36	35.7	40.7	63.6	71.4	-	-	57.1	36.4
ACETY	552/10014	96	88.9	95.8	91.9	95.7	4.8	97.0	-	69.6	94.9
METHY	215/10000	99	70.4	80.9	76.1	76.3	74.5	-	-	93.6	78.0
HYDRO	363/10000	70	67.8	68.4	68.7	70.8	65.3	62.5	62.5	69.3	69.3
AMID	723/10000	97	91.4	91.4	91.8	92.3	91.0	92.2	89.8	91.4	96.2
PYRRO	390/10000	93	95.9	90.2	85.7	98.9	86.5	100	-	89.7	89.4
GAMMA	232/10000	100	88.3	91.2	82.6	88.9	78.5	90.9	71.4	92.5	92.5

We present results for all types of projections (columns) and post-translational modifications (rows) using the precision P value that gives the percentage of observed positives that are correctly predicted (a measure of reliability of prediction of positive cases). The P is computed using the leave-one-out procedure, which removes from the training data one sample, constructs the model on the basis of the remaining training dataset and then tests the prediction of the model on the removed sample. The resulting error estimators are averaged for all such models (for all positive and all negative instances)

“-” means precision is not well-defined

Table 3 The overall classification errors for three generic embeddings: binary (BIN), lookup and profile (PROF) with numbers of support vectors used for each type of post-translational modification

Active Site Type	#positives	BIN	LOOKUP	PROF
PKA phosphorylation	86	0.55%/587	0.39%/143	0.43%/258
PKC phosphorylation	56	0.38%/787	0.40%/154	0.33%/390
CDC2 phosphorylation	41	0.28%/486	0.25%/83	0.37%/137
CK2 phosphorylation	62	0.53%/688	0.45%/157	0.52%/328
CK phosphorylation	85	0.72%/931	0.69%/229	0.70%/509
Acetylation	552	0.74%/1037	94.92%/236	4.93%/1650
Sulfation	83	0.78%/573	0.97%/141	0.75%/262
Amidation	363	0.47%/974	3.19%/800	3.61%/1234
Hydroxylation	363	2.11%/1221	2.58%/575	2.12%/677
Methylation	215	1.47%/1605	1.86%/326	1.44%/890
Pyrrolidone	390	1.71%/1669	2.25%/563	3.31%/1263
Gamma-carboxyglutamic	232	0.94%/1215	1.75%/413	1.27%/518

The first column presents numbers of positive instances found in the Swiss-Prot DB using annotation information (excluding annotations: BY SIMILARITY, PREDICTED, PROBABLE, POTENTIAL or PARTIAL annotations)

modification sites. The results considered are shown in Tables 1, 2 and 3. For all types of post-translational modification sites, the best kernel is the polynomial one. For this type of a kernel, the most stable representations are those that are mixed with the LOOKUP projection (e.g. PROF+LOOKUP and BLOSUM+LOOKUP). Other projections (e.g. generic BIN or PROF) have some advantages for particular types of modification sites, but have lower overall efficiency (small recall and precision values). When the number of positive instances is large, the simple binary method BIN becomes the most accurate; while in cases of lower numbers of occurrences profile methods yield better results. This can be explained by a higher sequence similarity between the instances tested in a larger collection of positives. The SVM finds a proper classification scheme of a test set more easily with a simple representation than with a more complex one. The linear kernel function is not efficient in the case of more complicated sequence signatures of post-translational modification sites. However, in some cases (PKA phosphorylation with SPARSE+LOOKUP representation) SVM models of this type are more efficient for the polynomial kernel. In the case of a radial basis kernel, an SVM frequently fails to build the model. In the case of large numbers of instances, the simple LOOKUP method for this type of kernel is the most accurate. Remarkable cases are acetylation, amidation and pyrrolidone, where the system with LOOKUP embedding reaches the greatest efficiency with the polynomial kernel.

The study of human protein kinase C family with the AutoMotif server

As an example, we show in Fig. 1a simple study of phosphorylation for the human protein kinase C family, whose members are involved in many biological pro-

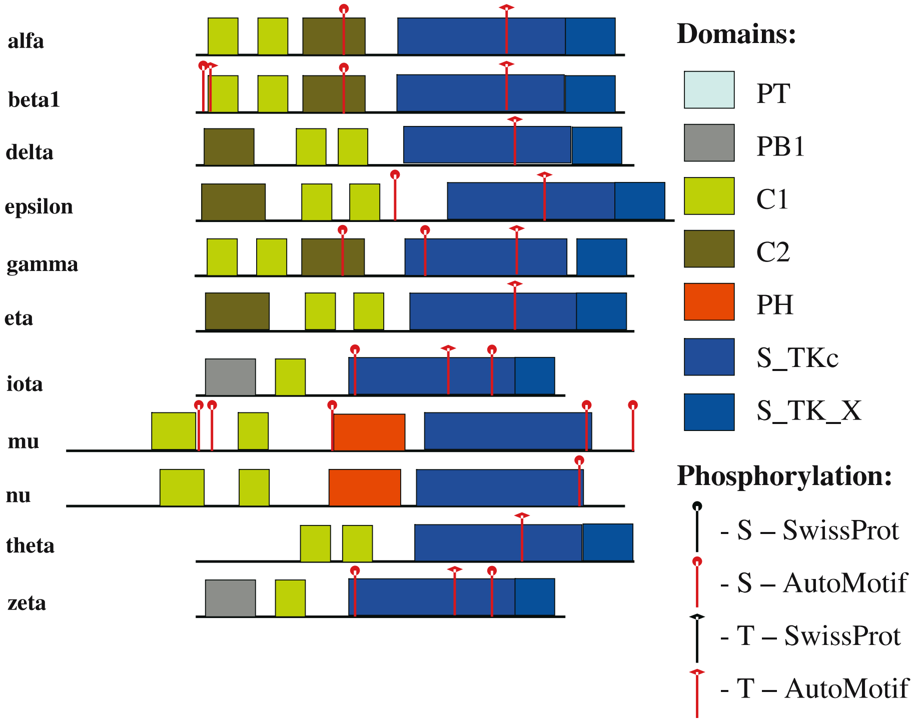

Fig. 1 Search for PKC autophosphorylation sites in PKC kinase family. Results of biological application of the AutoMotif Server for finding sites phosphorylated by PKC kinases are shown. The main type is autophosphorylation of threonine on the kinase domain, that is observed for all PKCs except mu and nu. Phosphorylation of serine on the C2 domain is observed for alfa, beta1 and gamma PKC. As the C2 domain is responsible for activation of PKC in response to secondary messangers (Ca^{2+}) this suggests a presence of feedback disabling aberrant activation of PKC in case of excess of secondary messengers

Table 4 Results of a simple study of phosphorylation by human protein kinase C family

Protein	Predicted motif	Start	Centre	End	Size	Score	Swiss-Prot annotation
PKC alfa (NP_002728.1)	DRRLSVEIW	237	241	245	9	0.22	
	VTTRTFCGT	493	497	501	9	0.67	
PKC beta1 (NP_002729.2)	EGEESTVRF	12	16	20	9	0.23	+
	GEESTVRFA	13	17	21	9	0.02	+
	DRRLSVEIW	237	241	245	9	0.22	
	VTTKTFCGT	496	500	504	9	1.00	+
PKC delta (NP_006245.2)	SRASTFCGT	503	507	511	9	1.10	
PKC epsilon (NP_005391.1)	ASGSSPSEE	333	337	341	9	0.41	
	VTTTTFCGT	562	566	570	9	0.90	
PKC gamma (NP_002730.1)	ERRLSVEVW	237	241	245	9	0.59	
	ERRGSDELY	369	373	377	9	0.68	
	TTTRTFCGT	510	514	518	9	0.42	
PKC eta (NP_006246.2)	VTTATFCGT	509	513	517	9	0.72	
PKC iota (NP_002731.2)	IGRGSYAKV	251	255	259	9	0.01	
	DTTSTFCGT	399	403	407	9	1.00	+
	PRSMSVKAA	473	477	481	9	0.24	
PKC mu (NP_002733.1)	RRRLSNVSL	201	205	209	9	0.36	
	LLQKSPSES	231	235	239	9	0.03	
	KRKSSTVMK	417	421	425	9	0.02	
	RKRYSVDKT	825	829	833	9	0.33	
	GERVSILXX	906	910	912	7	0.71	
PKC nu (NP_005804.1)	RKRYSVDKS	818	822	826	9	0.13	
PKC theta (NP_006248.1)	AKTNTFCGT	534	538	542	9	0.29	
PKC zeta (NP_002735.2)	IGRGSYAKV	258	262	266	9	0.01	
	DTTSTFCGT	406	410	414	9	1.00	+
	PRFLSVKAS	478	482	486	9	0.08	

The SVM with a binary representation (BIN) identifies similar patterns of phosphorylation in analyzed sequences. The main type is phosphorylation of threonine on the kinase domain, that is observed for all PKCs except mu and nu. Phosphorylation of serine on the C2 domain is observed for alfa, beta1 and gamma PKC. Predictions of phosphorylation are summarized in Tables 1 and 2. Notice the fact that all instances of phosphorylation found with a simple search were also identified by the SVM algorithm, but with scores varying from 0.0192 (T-17, PKB beta1) to 1.0006 (T-403, PKC iota)

cesses, including development, memory, differentiation, proliferation and carcinogenesis [28]. The 11 human isoforms of PKC can be divided into three groups. So-called conventional PKCs (PKC α, β, γ) have the same domain organization: C1-C1-C2-S/T kinase C (Fig. 1), where C1 is a phorbol ester/diacylglycerol binding domain and C2 is a calcium-dependent membrane-targeting module. The "novel PKC" group represented by δ, ε, η and θ is comprised of the same modules, but ordered differently than the group described previously (C2-C1-C1-S/T kinase C). Both of these PKC subfamilies require DAG (diacylogricelol) and calcium for their activation [29]. The remainder of the PKC protein family included in the category "atypical PKC" (μ, ν, ι, ζ) contains additional domains and needs only DAG for activation. We used the AutoMotif site to search for the phosphorylation sites in the PKC family (see Table 4).

Discussion

The analysis of post-translational modification sites by support vector machines allows for a fast and accurate (highly conservative) prediction of protein function. High overall precisions of the best methods allow users to gain deep insight into plausible functional characteristics of new proteins of unknown function. Recall efficiency ensures that information from previously verified sites will not be lost during automatic scans of known instances. The algorithm can be executed in a pipeline through our Web interface. Because of this, large-scale genomic analysis becomes feasible.

The main problem for some functional motifs is insufficient number of experimentally verified cases. Our annotation predictor can be improved significantly when statistical algorithms are utilized for more rigorous quantification of results. The numbers of support vectors for some of the models studied are large due to high dimensionalities of the embedding spaces and complicated shapes of the separation hyperplanes between positive and negative instances. The number of support vectors could be reduced significantly by choosing an initial low-dimensional encoding of amino acids in terms of their general physicochemical properties such as: hydrophobicity, hydrophilicity, polarity, volume, surface area, bulkiness or refractivity (see [30]).

Acknowledgements This work was supported by the USA grant ("SPAM" GM63208), ELM (QLRT-CT2000-00127), BioSapiens (LHSG-CT-2003-503265), GeneFun (LSHG-CT-2004-503567) projects within five and six FP EC programs. A. K. acknowledges the financial support provided by the NIH grant 1R01GM072014-01. LSW is supported by Foundation for Polish Science within Program for Young Researchers.

References

1. Attwood TK, Bradley P, Flower DR, Gaulton A, Maudling N, Mitchell AL, Moulton G, Nordle A, Paine K, Taylor P, Uddin A, Zygouri C (2003) Nucl Acids Res 31:400–402
2. Nevill-Manning CG, Wu TD, Brutlag DL (1998) Proc Natl Acad Sci USA 95:5865–5871
3. Huang JY, Brutlag DL (2001) Nucl Acids Res 29:202–204
4. Henikoff S, Henikoff JG, Pietrokovski S (1999) Bioinformatics 15:471–479
5. Zdobnov EM, Apweiler R (2001) Bioinformatics 17:847–848
6. Falquet L, Pagni M, Bucher P, Hulo N, Sigrist CJ, Hofmann K, Bairoch A (2002) Nucl Acids Res 30:235–238
7. Gattiker A, Gasteiger E, Bairoch A (2002) Applied Bioinformatics 1:107–108
8. Jonassen I, Collins JF, Higgins D (1995) Protein Science 4:1587–1595
9. Puntervoll P, Linding R, Gemünd C, Chabanis-Davidson S, Mattingsdal M, Cameron S, Martin DMA, Ausiello G, Brannetti B, Costantini A, Ferrè F, Maselli V, Via A, Cesareni G, Diella F, Superti-Furga G, Wyrwicz L, Ramu C, McGuigan C, Gudavalli R, Letunic I, Bork P, Rychlewski L, Küster B, Helmer-Citterich M, Hunter WN, Aasland R, Gibson TJ (2003) Nucl Acids Res 31:3625–3630
10. Obenauer JC, Cantley LC, Yaffe MB (2003) Nucl Acids Res 31:3635–3641
11. Monigatti F, Gasteiger E, Bairoch A, Jung E (2002) Bioinformatics 18:769–770
12. Kreegipuu A, Blom N, Brunak S, Jarv J (1998) FEBS Lett 430:45–50
13. Kreegipuu A, Blom N, Brunak S (1999) Nucl Acids Res 27:237–239
14. Blom N, Gammeltoft S, Brunak S (1999) J Mol Biol 294:1351–1362
15. Plewczynski D, Rychlewski L, Ye Y, Jaroszewski L, Godzik A (2004) BMC Bioinformatics 5:98
16. Plewczynski D, Rychlewski L (2003) Comput Methods Sci Technol 9:93–100
17. Plewczynski D, Jaroszewski L, Godzik A, Kloczkowski A, Rychlewski L (2005) J Mol Model (in press)
18. Bairoch A, Apweiler R (1999) Nucl Acids Res 27:49–54
19. Simons KT, Bonneau R, Ruczinski II, Baker D (1999) Proteins 37:171–176
20. Rohl CA, Strauss CE, Chivian D, Baker D (2004) Proteins 55:656–677
21. Bystroff C, Shao Y (2002) Bioinformatics 18:S54–S61
22. Vapnik VN (1995) The Nature of Statistical Learning Theory. Springer
23. Vapnik VN (1998) Statistical Learning Theory. Wiley, New York
24. Cristianini N, Shawe−Taylor J (2000) Support Vector Machines. Cambridge, UK
25. Zavaljevski N, Stevens FJ, Reifman J (2002) Bioinformatics 18:689–696
26. Kim H, Park H (2003) Protein Engin 16:553–560
27. Minakuchi Y, Satou K, Konagaya A (2003) Prediction of protein–protein interaction sites using support vector machines. Proceedings of the international conference on mathematics and engineering techniques in medicine and biological sciences, pp 22–28
28. Parekh DB, Ziegler W, Parker PJ (2000) EMBO J 19:496–503
29. Newton AC (1997) Curr Opin Cell Biol 9:161–167
30. Lohman R, Schneider G, Nehrens D, Wrede P (1994) Protein Sci 3:1597–1601