Published in final edited form as:

Biochem Biophys Res Commun. 2009 June 26; 384(2): 155–159. doi:10.1016/j.bbrc.2009.04.096.

Prediction of functionally important sites from protein sequences using sparse kernel least squares classifiers

Ke Tang^a, Ganesan Pugalenti^b, P.N. Suganthan^b, Christopher J. Lanczycki^c, and Saikat Chakrabarti^{c,*}

^aNICAL, Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China.

^bSchool of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

^cNCBI, NLM, National Institutes of Health, USA.

Abstract

Identification of functionally important sites (FIS) in proteins is a critical problem and can have profound importance where protein structural information is limited. Machine learning techniques have been very useful in successful classification of many important biological problems. In this paper, we adopt the sparse kernel least squares classifiers (SKLSC) approach for classification and/or prediction of FIS using protein sequence derived features. The SKLSC algorithm was applied to 5435 FIS that have been extracted from 312 reliable alignments for a wide range of protein families. We obtained 68.28% sensitivity and 68.66% specificity for training dataset and 65.34% sensitivity and 66.88% specificity for testing dataset. Further, large scale benchmarking study using alignments of 101 protein families containing 1899 FIS showed that our method achieved an average ~70% sensitivity in predicting different types of FIS, such as active sites, metal, ligand or protein binding sites. Our findings also indicate that active sites and metal binding sites are comparably easier to predict compared to the ligand and protein binding sites. Despite moderate success, our results suggest the usefulness and potential of SKLSC approach in prediction of FIS using only protein sequence derived information.

Keywords

functionally important sites; protein functional templates; machine learning algorithms; sparse kernel least squares classifiers

Introduction

Successful prediction of functionally important sites (FIS) in protein is very crucial and has major implications in molecular biology research, which aims to understand the functional properties and evolutionary diversities of proteins. Numerous approaches have addressed this issue using three-dimensional structural information [1,2], sequentially conserved residues [3,4], multiple sequence alignments (MSAs) [5–7] and information extracted from experimental studies [8–10]. Despite these efforts, the accuracy and/or sensitivity of FIS prediction still remains low. Prediction performance may suffer due to the lack of comprehensive knowledge of evolutionary conservation patterns of amino acid residues at known FIS.

*Corresponding author, Email: chakraba@ncbi.nlm.nih.gov.

Recently, we performed a detailed analysis of compositional and evolutionary conservation patterns at molecular and biological function level for a large set of known FIS [11]. We calculated the compositional and evolutionary conservation measures for known FIS and developed a library of protein functional templates (PFTs). Further we utilized the compositional and evolutionary conservation features towards prediction of FIS by developing a software tool named LIMACS (linear matching of conservation scores), which transfers the molecular function annotation from the most-similar functional site in the template library to the predicted sites [11,12]. The performance of this simple approach was encouraging but moderate and was also limited to utilization of fewer number of compositional and evolutionary conservation features. Therefore, more extensive range of feature selection and an exhaustive classification of the PFTs were warranted for better prediction performance.

Encouraged by the overwhelming success of machine learning methods in engineering, medical and financial applications, many research groups have been using neural networks (NN) [13], support vector machines (SVM) [14] and other machine learning techniques in biological field especially in classification and prediction of protein structure and functional profile. For example, neural networks have been used in prediction of trypsin protease cleavage sites [15], signal peptide cleavage sites [16] and enzyme active sites [17]. SVMs have also been successfully applied in various fields of computational biology, *e.g.*, pattern recognition problems [18,19], micro array and gene expression analysis [20] and prediction of FIS [21, 22].

In this study, we utilized a sparse kernel least squares classifiers (SKLSC) for classification and prediction of FIS that are involved with catalysis, ligand binding, protein binding, metal binding and miscellaneous molecular functions. Our study shows that intelligent feature design and subsequent utilization of a robust and efficient machine learning algorithm can lead to successful prediction of a wide variety of FIS.

Materials and methods

Dataset

The dataset used in this study was obtained from a library of PFTs [11,12] that has been assembled from FIS specified in a set of 312 curated alignments taken from the Conserved Domain Database [CDD; 10]. The CDD alignments for these families contain FIS (*e.g.* active sites, ligand binding sites, protein binding sites) identified in an extensive manual curation effort, and are based on evidence available in the literature and other relevant scientific sources [10]. 5435 FIS and 43899 non-FIS extracted from 312 alignments were used for this study. 3805 (~70% of the PFT library) FIS and 26876 non-FISs were selected for training. Testing was performed on the dataset containing 1630 (~30%) FIS and 11518 non-FIS. In addition, we also used a separate benchmarking dataset containing 1899 FIS from 101 CDD alignments that do not overlap with training and testing datasets. The distribution of various FIS in training, testing and benchmarking datasets is listed in Table 1.

Features

Each site in our training dataset is represented by 533 features (supplementary material 1) that include sequence and evolutionary conservation information extracted from the source MSAs. The details of the features used in this study are briefly mentioned below.

Amino acid type and functional group composition—The compositional diversity of a site was measured by calculating the frequencies of 20 amino acids and 10 functional groups within the alignment positions [11,12]. 20 amino acids were categorized into 10 functional groups based on the presence of side chain chemical group such as phenyl (F/W/Y), carboxyl

(D/E), imidazole (H), primary amine (K), guanidino (R), thiol (C), sulfur (M), amido (Q/N), hydroxyl (S/T) and nonpolar (A/G/I/L/V/P).

Dipeptides and tripeptides—Frequencies of dipeptides and tripeptides were used to represent protein sequences for classification purpose [23]. All possible pair wise and triplet peptide combinations were extracted from the 10 functional groups, giving rise to 66 dipeptide and 220 triplet combinations. The dipeptide and tripeptide frequencies were computed at each position (using 3 residues window) in the alignment and were represented by one or more pair wise and triplet combinations, respectively.

Content of secondary structural element (SSE)—SSE (helix: H, strand: E and coil: C) information was assigned to all the sequences in the alignment using a secondary structure prediction program, PSIPRED [24]. Composition of H, E and C is calculated using 3 residues window at each position.

Position specific scoring matrix (PSSM) scores—At each alignment column the median of the PSSM score for all the amino acid residues at that position, the frequency of occurrence of negative PSSM scores and the relative weight of negative PSSM scores [11] were calculated.

Physico-chemical properties—Matrices containing quantitative values for amino acids' physico-chemical properties scaled between 0 and 1 were obtained from the AAIndex database [25]. The computed physico-chemical properties includes hydration potential, molecular weight, isoelectric point, molecular volume, mutability, average accessible surface area, melting point, side chain volume, hydropathy, refractivity, polarity, optical activity, flexibility and normalized frequency for helix, coil, beta sheets (parallel/anti parallel) and beta turn.

Classification protocol

The classification/prediction models presented here is called sparse kernel least squares classifiers (SKLSC). The kernel least squares (KLS) approaches [26] emerged as the combination of traditional linear least square algorithms and the so-called “kernel trick” technique, which was first proposed for the well-known support vector machine (SVM) to handle nonlinearly separable data [14,27]. Despite the different problem formulations adopted by KLSC and SVM, the two approaches usually exhibit comparable classification performance. The original KLSC is readily applicable to the problems addressed in this study. However, it involves manipulation (including multiplication, inversion, etc.) of a matrix of size n -by- n , where n is the number of training samples. The time complexity of training a KLSC is typically $O(n^3)$, and applying the trained model to predict a new datum involves time complexity of $O(n)$. Hence, the original KLSC becomes highly time-consuming for both training and testing when n is large. To alleviate such disadvantage, we resort to the SKLSC. Below we describe the detailed procedures by which we obtained the SKLSC models.

Let $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, be the training data set, where $\mathbf{x}_i$ and y_i denote the d -dimensional sample vector and the corresponding class label, respectively. Further, we also define $V = \{\mathbf{v}_1, \dots, \mathbf{v}_l\}$ where l is a predefined number and $\mathbf{v}_j$ is a d -dimensional vector, called basis vector. A SKLSC (denoted as f) typically classifies a datum $\mathbf{x}$ by:

$$f(\mathbf{x}) = \text{sgn} \left[\sum_{j=1, \mathbf{v}_j \in V}^l \alpha_j k(\mathbf{v}_j, \mathbf{x}) \right] \quad (1)$$

where k is the so-called kernel function that models the relationship between $\mathbf{v}_i$ and $\mathbf{x}$, and the coefficients α_i 's are to be computed by training. In practice, the kernel function is usually defined before training, and the α_i 's can be computed by solving a system of linear equations:

$$(\mathbf{K}^T \mathbf{K} + \lambda \mathbf{I}) \boldsymbol{\alpha} = \mathbf{K}^T \mathbf{Y} \quad (2)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$, $\mathbf{Y} = [y_1, y_2, \dots, y_n]^T$ and λ is a predefined positive constant called the regularization parameter. $\mathbf{I}$ is an identity matrix of size n . $\mathbf{K}$ is a n -by- l matrix, whose element can be computed as $K_{ij} = k(\mathbf{x}_i, \mathbf{v}_j)$.

Unlike KLSC, the SKLSC only involves time complexity of $O(l^3)$ and $O(l)$ for training and testing, respectively. Hence, it requires significantly less computational cost as long as $l \ll n$. Obviously, the solution $\boldsymbol{\alpha}$ to Eq. (2) largely depends on the kernel matrix $\mathbf{K}$ and the regularization parameter λ , and the former is determined by the basis vector set V and the kernel function used. In this work, a Gaussian kernel $k(\mathbf{x}_i, \mathbf{v}_j) = \exp(-\sigma \|\mathbf{x}_i - \mathbf{v}_j\|^2)$ is used for the SKLSC model. One off-the-shelf approach for obtaining the basis vectors is to randomly sample the training data, which has been frequently adopted for achieving sparse kernel models [28]. In other words, we may acquire V by randomly sampling a subset (of size l) from S . With the Gaussian kernel function and the random sampling scheme, the performance of SKLSC now can be said to depend on three parameters, i.e., the kernel-parameter σ , the size (l) of V , and the regularization parameter λ . Choosing appropriate values for these parameters is usually referred to as model selection in machine learning literature. In most cases, such problem can be addressed in a generate-and-test manner. Following this framework, we first generated a set of parameter vectors. Each parameter vector is composed of three elements, representing the values for the three parameters. Then, we evaluated every parameter vector by a 5-fold cross-validation procedure on the training data. The parameter vector with the largest average Area Under Curve (AUC) was chosen as the parameters for building our final SKLSC. In general, 1000 appeared to be an upper-bound for l , and setting l to a larger value did not lead to significant improvement of the performance of SKLSC. Therefore, in comparison to the original KLSC, which has n larger than 30000 in our study, SKLSC is much more advantageous in terms of computational cost.

Feature Selection

Since the number of features in this study is high, it is natural to ask whether all the features are really indispensable for building an effective classification model. For this reason, we further conducted a feature selection procedure. We adopted the minimum redundancy–maximum relevance (MRMR) feature selection approach initially proposed by Ding and Peng [29]. MRMR is a framework rather than an exact algorithm, and the algorithm we adopted is an instantiation of MRMR proposed in [29].

Suppose that the training data are represented as an n -by- d matrix, where n and d is the number of samples and number of features, respectively. Let, $\mathbf{g}$ denotes a single n -dimensional feature vector (i.e., a column of the n -by- d matrix). The MRMR calculated the relevance score for $\mathbf{g}$ by:

$$F = \left[\sum_{k=1}^K n_k (\bar{g}_k - \bar{g})^2 / (K - 1) \right] / \sigma^2 \quad (3)$$

where K is the number of classes, n_k is the number of training samples of the k th class. $\bar{g}$ is the mean value of $\mathbf{g}$ on all samples and $\bar{g}_k$ is the mean value of $\mathbf{g}$ within the k th class. σ^2 is the

pooled variance that can be calculated by $\sigma^2 = [\sum_k (n_k - 1) \sigma_k^2] / (n - K)$ (where σ_k^2 is the variance of $\mathbf{g}$ within the k th class. Based upon Eq. (3), the relevance score of a feature subset G can be calculated by:

$$R_F = \frac{1}{|G|} \sum_{i \in G} F_i \quad (4)$$

Accordingly, MRMR measures the redundancy of G by

$$R_C = \frac{1}{|G|^2} \sum_{i,j \in G} C(\mathbf{g}_i, \mathbf{g}_j) \quad (5)$$

where $C(\mathbf{g}_i, \mathbf{g}_j)$ is the Pearson correlation coefficient between feature vectors $\mathbf{g}_i$ and $\mathbf{g}_j$.

In order to maximize the relevance while minimize the redundancy among the selected feature set, MRMR aims at finding the feature subset with the maximum value of R_F / R_C . We employed sequential forward selection scheme [29] to search for the optimal feature subset. First, the feature corresponding to the largest F_i (Eq. 3) is selected. After that, the rest of the features are selected one by one. At each iteration, the previously unselected features are evaluated according to Eq. (4) and Eq. (5), the one with the largest R_F / R_C is selected.

Results and discussion

Classification and prediction of FIS

The sparse kernel least squares classifiers [SKLSC] were applied to the PFT dataset(s). Each PFT in our dataset(s) is represented by 533 feature vectors. The SKLSC algorithm was trained using the training dataset and subsequently the performance of the classifier was tested on the testing dataset. With all 533 features, we achieved 68.28% sensitivity and 68.66% specificity in the training data and 65.34% sensitivity and 66.88% specificity in the testing data (Table 2 and Table 3).

Generally, not all features contribute equally to the classification; some features play a relatively more prominent role than others in specific aspects. It is therefore of interest to examine which feature properties play more prominent roles in the classification of PFT. We selected 8 feature subsets by decreasing the number of features to analyze the impact of the feature selection procedure on the classification performance. The performance of the method for discriminating between FIS and non-functional sites is summarized in Table 2 and Table 3. As can be observed, feature selection (reduction) generally does not deteriorate the classification performance much. The usage of smaller number of features only leads to a slight decrease in the sensitivity and specificity rate. When reducing the number of features from 533 to 20, the sensitivity of our model merely decreased from 65.34% to 60.80% (on testing samples), while specificity decrease from 66.88% to 62.04%. Markedly lower performance was only observed when we further reduced the feature numbers to 5.

A closer look at the top 10 or 20 selected features (supplementary material 2) reveals that they mostly include features representing the fractions of functional groups (*e.g.*, phenyl, carboxyl, amine, guanidine etc.), evolutionary conservation scores (information content, PSSM score etc.) and distribution of dipeptides and tripeptides patterns within the close vicinity of functional sites. This finding indicates that distribution pattern of neighboring residues together with

compositional and evolutionary conservation patterns might play an important role in delineation of FIS.

The performance of the SKLSC in prediction of individual type of FIS (*e.g.*, active sites, ligand binding sites, protein binding sites) was also investigated. Table 4 and Table 5 list the sensitivity values for each individual type or class of FIS within training and test datasets. The classifier performed well in prediction of catalytic and metal binding sites whereas moderate successes were achieved in case of ligand binding, protein binding and miscellaneous sites (Table 4 and Table 5).

Benchmarking studies

To further test the efficiency of the method, we applied SKLSC algorithm on a benchmarking dataset, containing 1899 FIS extracted from 101 CDD alignments that do not overlap with the training and testing datasets. MSAs from these 101 families were used as the input for the SKLSC function prediction algorithm and each column of the MSA is matched against the training datasets. SKLSC method was very successful in identifying active sites (~84% sensitivity) and metal binding sites 100% sensitivity) while provided moderate (ligand binding and miscellaneous sites) to low (protein binding sites) performance in predicting other functional categories. Overall, SKLSC method performed slightly better than LIMACS [12] in predicting individual classes while achieved average 69.52% sensitivity in prediction of FIS involved in all classes of molecular functions. Conservation based scoring schemes are widely used to predict FIS [30,31]. We compared the performance of SKLSC method, which is also a sequence based approach with other available conservation score based methods, such as AL2CO [6] and SCORECONS [31] using 101 CDD alignments within our benchmarking dataset. We compared the prediction results by calculating fraction of true positive rate (sensitivity) and false positive rate (error rate) in prediction of FIS. As shown in Table 7, SKLSC classifier performed comparably well in prediction of FIS.

Conclusion

We employed a sparse kernel least squares classifiers (SKLSC) approach for prediction of FIS in proteins using sequence derived features. We used a library of PFTs that has been assembled from FIS specified in a set of 312 curated alignments taken from the Conserved Domain Database. Each PFT in this library contains a quantitative part represented by 16 compositional and evolutionary conservation features. In this study, we aimed to strengthen the feature repertoire for each PFT by including additional features representing amino acids' physico-chemical properties, dipeptide/tripeptide combination pattern and content of secondary structural element. We applied the SKLSC algorithm on this enriched PFT dataset and obtained 68.28% sensitivity and 68.66% specificity on training dataset and 65.34% sensitivity and 66.88% specificity for testing dataset. Large scale benchmarking using a separate dataset of 101 protein families containing 1899 FIS shows ~70% sensitivity for the SKLSC algorithm. Our results also indicated that the classifier performed quite well in prediction of active site and metal binding sites whereas reasonable successes were achieved in case of ligand binding, protein binding sites. Nonetheless, this approach could be extremely useful in prediction of FIS especially considering the fact that it utilizes protein sequence derived information.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

K.T is financially supported by a National Natural Science Foundation of China Grant (No. 60802036). GP and P.N.S acknowledge the financial support offered by the A*Star (Agency for Science, Technology and Research). S.C and C.J.L acknowledge the support provided by the Intramural Research Program of the National Library of Medicine at National Institutes of Health/DHHS.

References

1. Fetrow JS, Skolnick J. 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* 1998;281:949–968. [PubMed: 9719646]
2. Zhang B, Rychlewski L, Pawlowski K, Fetrow JS, Skolnick J, Godzik A. From fold predictions to function predictions: automation of functional site conservation analysis for functional genome predictions. *Protein Sci* 1999;8:1104–1115. [PubMed: 10338021]
3. Hofmann K, Bucher P, Falquet L, Bairoch A. The PROSITE database, its status in 1999. *Nucleic Acids Res* 1999;27:215–219. [PubMed: 9847184]
4. Casari G, Sander C, Valencia A. A method to predict functional residues in proteins. *Nat. Struct. Biol* 1995;2:171–178. [PubMed: 7749921]
5. Hannenhalli SS, Russell RB. Analysis and prediction of functional sub-types from protein sequence alignments. *J. Mol. Biol* 2000;303:61–76. [PubMed: 11021970]
6. Pei J, Grishin NV. AL2CO: calculation of positional conservation in a protein sequence alignment. *Bioinformatics* 2001;17:700–712. [PubMed: 11524371]
7. Berezin C, Glaser F, Rosenberg J, Paz I, Pupko T, Fariselli P, Casadio R, Ben-Tal N N. ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics* 2004;20:1322–1324. [PubMed: 14871869]
8. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JP, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Gerald M, Rubin GM, Sherlock G. Gene Ontology: tool for the unification of biology. *Nature Genetics* 2000;25:25–29. [PubMed: 10802651]
9. Bateman A, Birney E, Cerruti L, Durbin R, Eddy SR, Griffiths-Jones S, Howe KL, Marshall M, Sonnhammer EL. The Pfam protein families database. *Nucleic Acids Res* 2002;30:276–280. [PubMed: 11752314]
10. Marchler-Bauer A, Panchenko AR, Shoemaker BA, Thiessen PA, Geer LY, Bryant SH. CDD: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Res* 2002;30:281–283. [PubMed: 11752315]
11. Chakrabarti S, Lanczycki CJ. Analysis and Prediction of FIS in Proteins. *Protein Science* 2006;16:4–13. [PubMed: 17192586]
12. Lanczycki CJ, Chakrabarti S. A tool for the prediction of FIS in proteins using a library of functional templates. *Bioinformation* 2008;2:279. [PubMed: 18478080]
13. Rumelhart, DE.; McClelland, JL. Parallel distributed processing: exploration in the cognition. Cambridge, MA: MIT Press; 1986.
14. Cortes C, Vapnik V. Support vector networks. *Machine Learning* 1995;20:273–297.
15. Thomson R, Hodgman TC, Yang ZR, Doyle AK. Characterising proteolytic cleavage site activity using bio-basis function neural networks. *Bioinformatics* 2003;19:1741–1747. [PubMed: 14512344]
16. Nielsen M, Lundegaard C, Worning P, Lauemoller SL, Lamberth K, Buss S S, et al. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Science* 2003;12:1007–1017. [PubMed: 12717023]
17. Gutteridge A, Bartlett GJ, Thornton JM. Using a neural network and spatial clustering to predict the location of active sites in enzymes. *J Mol Biol* 2003;330:719–734. [PubMed: 12850142]
18. Noble, WS. Support vector machine applications in computational biology, Kernel Methods in Computational Biology. Schoelkopf, B.; Tsuda, K.; Vert, J-P., editors. Cambridge, MA: MIT Press; 2004. p. 71-92.

19. Ben-Hur A, Brutlag D. Remote homology detection: a motif based approach. *Bioinformatics* 2003;19:26–33.
20. Tang EK, Suganthan PN, Yao X. Gene Selection Algorithms for Microarray Data Based on Least Square Support Vector Machine. *BMC-Bioinformatics* 2006;7:95. [PubMed: 16504159]
21. Bhardwaj N, Lu H. Residue-level prediction of DNA-binding sites and its application on DNA-binding protein predictions. *FEBS Lett* 2007;581:1058–1066. [PubMed: 17316627]
22. Pugalenth G, Kumar KK, Suganthan PN, Gangal R. Identification of catalytic residues from protein structure using support vector machine with sequence and structural features. *Biochem Biophys Res Commun* 2008;367:630–634. [PubMed: 18206645]
23. Pugalenth G, Tang K, Suganthan PN, Archunan G, Sowdhamini R. A machine learning approach for the identification of odorant binding proteins from sequence-derived properties. *BMC Bioinformatics* 2007;8:351. [PubMed: 17880712]
24. McGuffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. *Bioinformatics* 2000;16:404–405. [PubMed: 10869041]
25. Kawashima S, Ogata H, Kanehisa M. AAindex: amino acid index database. *Nucleic Acids Res* 1999;27:368–369. [PubMed: 9847231]
26. Engel Y, Mannor S, Meir R. The Kernel Recursive Least Squares Algorithm *IEEE Transactions on Signal Processing*. 2004;52:2275–2285.
27. Rifkin, R.; Yeo, G.; Poggio, T. Regularized least squares classification. In: Suykens, JAK.; Horvath, G.; Basu, S.; Micchelli, CA.; Vandewalle, J., editors. *Advances in learning theory: Methods, models and applications*. Amsterdam: IOS Press; 2003. p. 131-154.
28. Keerthi SS, Chapelle O, DeCoste D. Building Support Vector Machines with Reduced Classifier Complexity. *Journal of Machine Learning Research* 2006;8:1–22.
29. Ding, C.; Peng, H. Minimum Redundancy Feature Selection for Gene Expression Data, in *Proc. IEEE Computer Society Bioinformatics Conference*; 2003. p. 523-529.
30. Cheng G, Qian B, Samudrala R, Baker D. Improvement in protein functional site prediction by distinguishing structural and functional constraints on protein family evolution using computational design. *Nucleic Acids Res* 2005;33:5861–5867. [PubMed: 16224101]
31. Valdar WSJ. Scoring residue conservation, *Proteins: Structure, Function, and Genetics*. 2002;43:227–241.

Table 1

The distribution of various functional sites in training and testing datasets

Functional Sites	Class	Training	Testing	Benchmarking
Catalytic site	1	475	203	284
Ligand binding site	2	1165	499	544
Protein binding site	3	1785	765	917
Metal binding site	4	219	94	55
Miscellaneous function site	5	161	69	99
All functions		3805	1630	1899

Table 2

Classification results achieved on different feature subsets of training dataset

No. of Features	Sensitivity (%)	Specificity (%)	AUC
All	68.28	68.66	0.7466
400	68.04	67.33	0.7436
300	69.43	67.74	0.7496
200	70.67	65.94	0.7539
100	63.99	60.52	0.6758
50	62.97	61.30	0.6698
20	61.60	62.35	0.6702
10	65.23	59.17	0.6726
5	57.74	56.28	0.5854

Table 3

Classification results achieved on different feature subsets of testing dataset

No. of Features	Sensitivity	Specificity	AUC
All	65.34	66.88	0.7043
400	65.46	65.78	0.7030
300	63.80	66.60	0.7015
200	65.28	63.97	0.7019
100	62.94	59.63	0.6606
50	61.66	60.14	0.6577
20	60.80	62.04	0.6610
10	62.09	58.86	0.6484
5	55.46	56.46	0.5750

Table 4

Sensitivity achieved on training data from 5 classes

No. of Features	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)
All	81.47	71.67	60.11	84.02	73.91
400	78.95	71.50	60.11	86.30	73.91
300	80.63	74.33	60.67	88.13	72.67
200	82.11	73.73	63.25	89.95	70.81
100	83.37	63.78	56.47	84.47	63.98
50	82.74	63.09	54.90	84.47	63.98
20	82.11	61.37	53.33	82.65	65.84
10	82.32	65.84	57.93	81.74	68.94
5	72.00	62.32	51.37	63.01	45.96

Table 5

Sensitivity on testing data from 5 classes

No. of Features	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)
All	75.86	70.74	56.99	76.60	72.46
400	75.37	71.34	51.73	79.79	71.01
300	78.82	68.54	55.03	75.53	66.67
200	79.31	70.54	57.12	75.53	62.32
100	81.77	66.73	53.46	78.72	63.77
50	80.30	64.93	53.33	76.60	55.07
20	81.28	63.53	52.29	73.40	57.97
10	79.31	62.32	56.08	74.47	59.42
5	69.46	57.11	51.90	58.51	37.68

Table 6
Performance of SKLSC and LIMACS on the benchmarking dataset

Class	Functional Sites	Total Sites	SKLSC		LIMACS	
			Correctly Predicted	Sensitivity (%)	Correctly Predicted	Sensitivity (%)
1	Catalytic site	284	237	83.45	225	79.22
2	Ligand binding site	544	325	59.74	310	56.98
3	Protein binding site	917	420	45.80	405	44.16
4	Metal binding site	55	55	100.00	50	90.90
5	Miscellaneous functions	99	58	58.59	48	48.49
	All Functions		Average sensitivity		Average Sensitivity	
			69.52		63.95	

Table 7

Comparison of prediction performance

Error rate (%)	Sensitivity (%)		
	SKLSC	AL2CO	SCORECONS
1	8	3	4
5	21	14	17
15	45	34	37