Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Classification of Protein Structure using SVM

Caleb Lal[1], Er. Arachana Singh[2]

[1]*M.Tech Student, Department of CSE SSET, SHIATS, Allahabad*
[2]*Assistant professor Department of CSE SSET, SHIATS, Allahabad*

Abstract- Protein domains are portion block of protein sequence that evolved independent function. Therefore, the classification of protein domain is becoming very important in order to produce new sequence with new function. However the main issue in protein domain classification is to classify the domain correctly into their category since the sequence coincidently classify to both category. Therefore, to overcome this issue, this dissertation proposed a method of functionally classifying genes by using gene expression data from DNA microarray hybridization experiments. The method is based on the theory of support vector machines (SVMs). SVMs are considered a supervised computer learning method because they exploit prior knowledge of gene function to identify unknown genes of similar function from expression data. SVMs avoid several problems associated with unsupervised clustering methods, such as hierarchical clustering and self-organizing maps.
SVMs have many mathematical features that make them attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large data sets, the ability to handle large feature spaces, and the ability to identify outliers. We test SVM that use different similarity metrics, as well as some other supervised learning methods and find that the SVM best identified different sets of genes with a common function using expression data. Finally, we use SVM to predict functional roles for uncharacterized yeast (Open Reading Frames) ORF based on their expression data.

Keywords- SVM, DNA, Knowledge Base, ORF, TSP, GP, GEP

I. INTRODUCTION

Most biological processes, such as regulation of metabolic and signaling pathways, DNA replication and transcription, cell adhesion, immunologic recognition as well as protein synthesis, are dominated by protein-protein interactions and complex formation. One of the primary objectives of the post-genomic era is the elucidation of the interactions in cellular systems. Localization of such interactions to so-called “functional sites” or “binding sites” will allow us to understand how the protein recognizes other molecules, to gain clues about its likely function at the level of the cell and the organism, and to identify important binding sites that may serve as useful targets for pharmaceutical design [1]. X-ray crystallography, NMR spectroscopy and electron microscopy can obtain structural information of proteins and protein complexes, which facilitate understanding the mechanism of protein interactions on a residue and atom level. However, the number of structures of macromolecular assemblies solved by these experimental methods is still quite small compared to that of the individual proteins [2] and it cannot meet the requirement for the development of proteome. Computational approaches are therefore needed to assist the finding of potential binding sites from these physical and chemical properties. these methods include detecting the presence of “proline brackets” [3], solvent accessible surface area buried upon association [4], free energy changes upon alanine-scanning mutations [5], in silico two hybrid systems [6], sequence hydrophobicity distribution [7], patch analysis using six parameters of surface [8]. Also, several studies have attempted to predict protein-protein interaction sites from sequence or structure conservation information [9-14]. Three published methods encode some of protein sequence

characteristics, i.e., residue sequence profile and solvent accessibility, into neural networks (NN) or support vector machines (SVM) in order to predict binding regions in the structure-known or structure-unknown proteins [9-12]

II. LITERATURE SURVEY

Protein domains are considered as structural and functional units of protein where the protein domains are defined using multiple or combination of protein sequence criteria. The protein domain classification is important to understand the protein structure and function. Because of this, previous research works have proposed various structures of protein domain classification methods based on protein sequence, but the information from sequence is limited.

The merging of protein domains classification in protein sequence is an evolutionary process, contributing to the great diversity of proteins that produces a lot of favourable energy. However, the classification of protein domain based on protein sequence only is more likely to introduce incorrectly folded regions and making the classification to single and multiple domain categories are difficult task. To analyze new protein with new function, accurate classification of protein domain is much needed to make scientist work easier since protein domain is important to analyze the different functions of protein sequences. The different functions of protein sequence enable us to probe the function of the protein, perform drug design and construct novel protein. However, various problems are found in protein domain classification throughout the years is coincidently classified to both categories either single or multiple domain. Currently there are several computational protein domain classification methods available such as method based on similarity and multiple sequence alignment, known protein structure, dimensional structure, used model based and protein sequence information. Methods based on similarity and use multiple sequence alignments to represent protein domain. A sequence database search provides information on pairwise similarities.

Example works done in this category are SVMFold [1], EVEREST [2] and Biozon [3]. Methods that depend on known protein structure to identify the protein domain since structural data are available for only a relatively small number of proteins. Several methods handle the problem of domain prediction by employing structure classification methods by using other types of predicted known information such as from sequence databases. Example works are AutoSCOP [4], Class of Architecture, Topology and Homologous superfamily (CATH) [5] and Structural Classification of Proteins (SCOP) [6]. Methods that use dimensional structure to assume protein domain boundaries is based on the same general principle that assumes domains to be structurally compact and separate substructures with higher density of contacts within the substructures than with their surroundings. Works done in this category are PROMALS [7], DDBASE [8] and Mateo [9]. Methods that used comparative model to identify other member of protein domain family such as Protein Family database (Pfam) [10], Conserved Domain Database (CDD) [11] and Simple Modular Architecture Research Tool (SMART) [12]. Methods that are based only on sequence information to provide an appealing alternative, especially for large-scale domain classification such as Domain Guess by Size (DGS) [13]. Classification algorithm is a procedure for selecting a hypothesis from a set of alternatives where that is best fits a set of observations.

The goal of classification is to build a set of models that can correctly classify the class of the different objects into their categories. Recently, several classification algorithm have been produced and used in bioinformatics such as algorithms based on fuzzy clustering [14], Neural Network [15], Bayesian classification [16], decision tree [17], logistic regression [18] and Support Vector Machine (SVM) [19, 20]. However, some of these methods were tested on small datasets, often with relatively high sequence identity, which resulted in high classification accuracy such as works done by Chen et. al. [15]. The SVM is usually used to map the input vector into one feature space which is relevant with kernel function and seek an optimized linear division that construct the n-separated hyperplane where the n is classes of protein sequence in dataset. These steps are important to make the classification by SVM more accurate and will achieve higher performance.

Cloud-CoXCS, is a machine learning classification system which divide the the searching space into multiple sub searching spaces and assign an independent XCS to each one. Cloud-CoXCS benefits from running parallel XCSs on the Cloud infrastructure to speed up the learning process. Cloud-CoXCS is composed of three components: CoXCS, Aneka, and Offspring. In the remainder of the section, a brief overview of all these three components will be provided.

A. CoXCS: CoXCS is a coevolutionary learning classifier based on feature space partitioning. It extends the XCS model by introducing a coevolutionary approach. A schematic example of how different classifiers learn from the feature space and interact with each other. The CoXCS architecture is based on a collection of independent populations of classifiers that are trained using different partitions of the feature space within the training dataset. The model uses a modified covering operator and crossover operators, which improves the generation of new classifiers during the evolutionary process. After a fixed number of iterations, selected classifiers from each of the independent populations are transferred to a different population, the evolutionary cycle is then repeated.

This process continues until a specific accuracy threshold is reached.

B. Aneka: Aneka is a platform for developing applications and deploying them on Clouds. It provides a runtime environment and a set of APIs that allow developers to build .NET applications that offload their computation on both public and private clouds. One of the key features of Aneka is the ability to support multiple programming models (ways of expressing the execution logic of applications by using specific abstractions). This is accomplished by creating a customizable and extensible service oriented runtime environment represented by a collection of software containers connected together. By leveraging this architecture, advanced services including resource reservation, persistence, storage management, security,

and performance monitoring have been implemented. On top of this infrastructure, different programming models can be plugged to provide support for different scenarios such as engineering, life science, and business applications. The internal architecture of the Aneka Container. A container is the building block of Aneka Clouds. It provides a collection of services that perform all the operations required by the system: security, scheduling, job execution, and storage. The container can be deployed on either physical machine or virtual resources that are dynamically provisioned on demand by interacting virtual machine managers such as Amazon, VMWare, and Xen. On top of this architecture, three programming models are supported: independent bag of tasks (Task Model), distributed threads (Thread Model), and mapreduce (MapReduce Model). Developers can define their own abstraction for programming distributed applications with Aneka and simply configure the services required for the scheduling and the execution of the units of work. The setup prepared for Cloud-CoXCS has been configured with the Task Model for the execution of the classification jobs. The Task Model provides a very simple set of abstractions that allows developers to define a sequence of unrelated tasks that do not have precedence or sequencing constraints. By using the Task Model it is possible to wrap existing legacy applications or also implement new tasks with any language supported by the .NET runtime. In the case of Cloud-CoXCS the existing CoXCS application has been packaged into a legacy task and remotely executed.

C. Offspring: Offspring is a software tool that allows scientists and developers to quickly prototype distributed applications Strategies are programmable client-side workflows that developers can define and plug into the environment. By defining a strategy, developers can coordinate the execution of existing legacy applications, as in the case of Cloud-CoXCS, or implement more sophisticated models by implementing their own tasks. A strategy is composed of a sequence of phases in which a collection of tasks is generated. Each of these tasks are submitted through Offspring and executed remotely. Their successful completion (or failure) can trigger the generation of additional tasks within the same phase or move strategy to the next phase. It is possible to model either simple parameter sweeping applications or complex dynamic workflows. In the case of Cloud-CoXCS, a multi-phase strategy has been implemented. In each phase, a number of parallel learning tasks are generated. The output of a learning task is a population of classifiers that have been trained

against a given dataset. Once all the learning tasks complete, an additional task that applies migration among the population of classifiers will be submitted. It sets the completion of the phase once its execution finishes. This process is repeated for a specified number of iterations decided by the user.

III. PROPOSED METHODOLOGY

3.1 Dataset:

Protein domain classification with modified SVM using sigmoid kernel is tested from SCOP version 1.75 [22]. The SCOP 1.75 with 40% less identity in PDB contains 1070. The protein sequences are reconstructed from which short protein sequences that are less than 40 amino acids are removed. Then, BLAST is performed to search seed protein sequences as a dataset. The protein sequences that have more than 20 hits are kept. The dataset is split into training and testing datasets. We divided the dataset into training and testing datasets. Training dataset is used for optimizing the DNN parameters and for training the DNN classifier to predict unseen protein domain boundaries. Testing dataset is used for evaluating the performance of the DNN. The dataset are split into training and testing datasets with 80:20 ratios.

If the protein sequences are longer than 600 amino acids, the protein sequences are separated into a segment based on ordered and disordered regions [22]. Multiple sequence alignment (MSA) is performed using Clustal Omega algorithm [23] in order to give the information of protein domain. Alignments are represented as a protein sequence of alignment column that associated with one position in protein sequence.

Then the pairwise alignments generated by Clustal Omega are extracted to make the protein domain boundaries clearer. In the extraction, a domain boundary signal is defined as a gap which begins at the N or C terminal. The gap with 45 residues or more will remove and the continuous sequence over 45 residues will remain for generating the protein domain signal.

The extractions of pairwise alignment are expected to increase PSI-BLAST e-value [24].

3.2 Preliminary concepts:

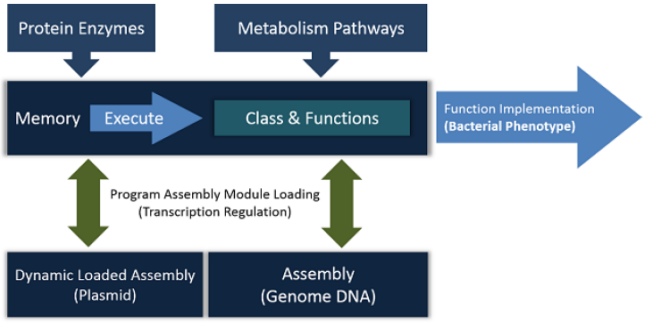

Fig. 1: Cell System

If we comparing the cell process with the .NET program assembly and its running way, then we will found out that:

Gene expression processing just like we create an object instance from specific type information in the programming, and the protein enzymes is the class instance of a gene. Then the expressed proteins will implements some phenotype function from catalyzed some metabolism pathways, and this is just like a method invoke. So if a cell system architecture can be treat as a program assembly, and the cell components is equals to the object class instance in a .NET program, then which means we can modify the cell function process from we modify the genome information. Actually the traditional genetic engineering method is a way of genome reprogramming method, with modify the genome then create a mutant, then its cell function changed. This is just like modify the source code and compile a new assembly. So the DNA sequence just like the binary sequence of the compiled program assembly, and the molecular experiments in the laboratory is the work of disassembly. One

conclusion about the cell system: Comparing with a program running way, the cell system processes are more likely the threads in a program.

3.3 Work Flow:

Proposed algorithm do prediction of structure very effectively with following simple working steps:

a) Define the population size
b) Select the selection method as follow:
a. Elite
b. Rank
c. Roulette
c) Define function set as:
a. Simple
b. Extended
d) Select Genetic method:
a. GP
b. GEP
e) Define window size
f) Define prediction size
g) Choose number of iterations (0 – infinity)

Current iteration will generate two important parameters a) Learning error value and b) prediction error value.

3.4 Experimental setup used:

Experimental setup is as follow:

1. Hardware configuration:
a) Processor: Intel core i3
b) HDD: 500 GB
c) RAM: 8 GB
2. Software configuration:
a. OS: Windows 7
b. Framework: Visual Studio 2010
c. Language: c#

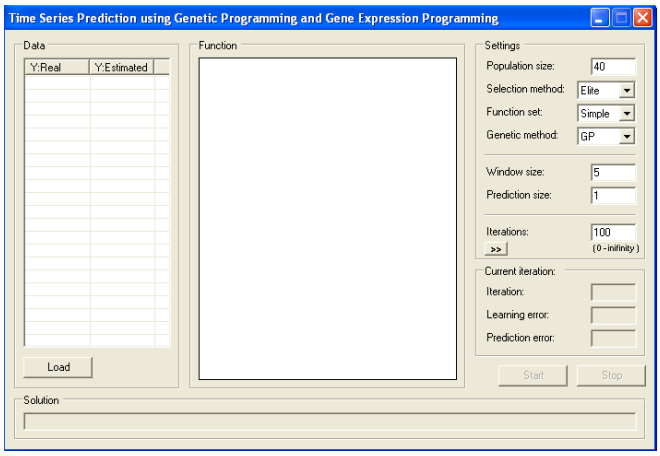

Fig. 2: GUI of TSP using GP

3.5 Proposed co evolutionary algorithm:

Complete work flow consists of following algorithms:

1. Population Algorithm
2. Selection Algorithms
a. Elite Selection

b. ISelection Method
c. Rank Selection
d. Roulette Wheel Selection
3. Fitness Functions
a. IFitness Function
b. Optimization Function 1D
c. Optimization Function 2D
d. Symbolic Regression Fitness
e. Time Series Prediction Fitness

Time series prediction fitness function algorithm is one of the most important algorithms. Their details are as follow:

```
Start;
Evaluate( IChromosome chromosome )
{string function = chromosome.ToString( );
double error = 0.0;
for ( int i = 0, n = data.Length - windowSize - predictionSize; i < n; i++ )
{for ( int j = 0, b = i + windowSize - 1; j < windowSize; j++ )
{variables[j] = data[b - j];}
{
double y = PolishExpression.Evaluate( function, variables );
if ( double.IsNaN( y ) )
return 0;
error += Math.Abs( y - data[i + windowSize] );}}
return 100.0 / ( error + 1 );
End;
```

IV. RESULT AND DISCUSSION

The resultant function graph of time series prediction using GP is as follow:

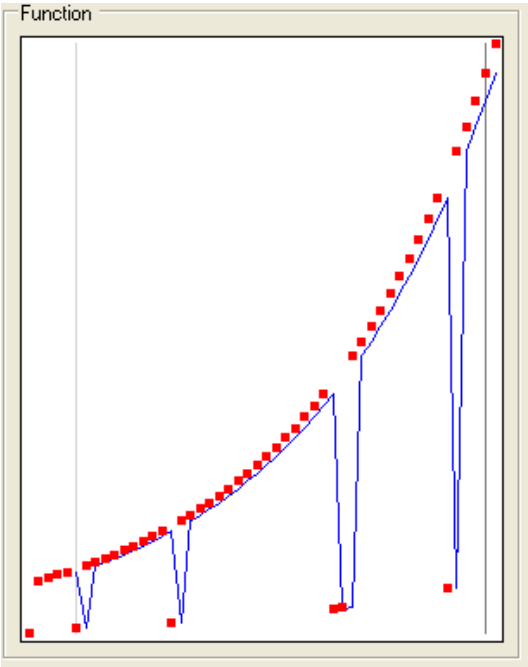

Fig. 3: Function graph of TSP using GP

Prediction error value is considered as result evaluation parameter. Here following algorithm are compared:

1. Rank
2. Elitte
3. Roulette

In this experiment following standard values are assumed:

a) Population size: 40
b) Function Set: Simple
c) Genetic Method: GP
d) Window size: 5
e) Prediction size: 1
f) Number of iteration = 100

On the execution following results are found:

Algorithm	Prediction Error Value
Roulette	5651
Elite	19
Rank	5651

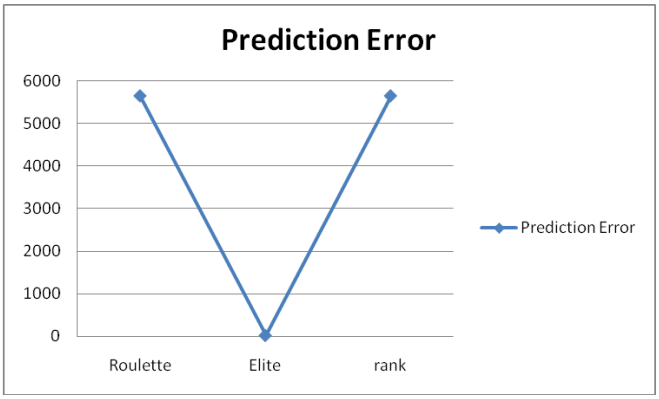

Fig. 4: Graphical representation of prediction error value

On the basis of result this is observed that for the prediction of time series prediction co evolution algorithm with elite selection method work far better than Rank and Roulette methods.

V. CONCLUSION

Proposed solution, which use co evolutionary novel technique with elitte selection algorithm based on Genetic Programming provides better result for identification and prediction of Time series prediction for protein structure.

REFERENCES

[1] T. Hamp, F. Birzele, and F. Buchwald, "Improving structure alignmentbased prediction of SCOP families using Vorolign Kernels," Bioinformatics, vol. 27, pp. 204-210, November 2010.
[2] E. Portugaly, A. Harel, N. Linial, and M. Linial, "M.: EVEREST: Automatic Identification and Classification of Protein Domains in All Protein Sequences," BMC Bioinformatics, vol. 7, pp. 27-286, Jun 2006.
[3] N. Nagaranjan, and G.Yona, "Automatic Prediction of Protein Domain from Sequence Information using a Hybrid Learning System," Bioinformatics, vol. 20, pp. 1335-1360, February 2004.
[4] J.E. Gewehr, and R. Zimmer, "SSEP-Domain: Protein Domain Prediction by Alignment of Secondary Structure Elements and Profiles," Bioinformatics, vol. 22, pp. 181-187, January 2006.
[5] A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH-a Hierarchic Classification of Protein Domain Structures," Structure, vol. 5, pp. 1093-1108, August 1997.
[6] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A Structural Classification of Protein Database for the Investigation of Sequences and Structures," Journal of Molecular Biology, vol. 247, pp. 536-540, April 1995.
[7] J. Pei, and N.V. Grishin, "PROMALS: Towards Accurate Multiple Sequence Alignments of Distantly Related Protein," Bioinformatics, vol. 23, pp. 802-808, April 2007.

[8] A. Vinayagam, J. Shi, G. Pugalenthi, B. Meenakshi, T.L. Blundell, and R. Sowdhamini, "DDBASE2.0: Updated Domain Database with Improved Identification of Structural Domains," Bioinformatics, vol.
19, pp. 1760-1764, September 2003.
[9] M. Lexa, and G. Valle, "Pimex: Rapid Identification of Oligonucleotide Matches in whole Genomes," Bioinformatics, vol. 19, pp. 2486-2488, May 2003.
[10] R.D. Finn, J. Mistry, B. Schuster-Bockler, S. Griffiths-Jones, V. Hollich, T. Lassmann, S. Moxon, M. Marshall, A. Khanna, R. Durbin, S.R. Eddy, E.L Sonnhammer, and A. Bateman, "Pfam: Clans, Web Tools and Services," Nucleic Acids Research, vol. 34, pp. D247-D251 January 2006.
[11] A. Marchler, J.B. Anderson, M.K. Derbyshire, C. DeWeese-Scott, N.R. Gonzales, M. Gwadz, L.S. HaoHe, D.I. Hurwitz, J.D. Jackson, K. Zhaoxi, D. Krylov, C.J. Lanczycki, C.A. Liebert, C. Liu, F. Lu, S. Lu, G.H. Marchler, M. Mullokandov, J.S. Song, N. Thanki, R.A. Yamashita, J.J. Yin, D. Zhang, and S.H. Bryant, "CDD: A Conserved Domain Database for Interactive Domain Family Analysis," Nucleic Acids Research, vol. 35, pp. D237-D240, January 2007.
[12] I. Letunic, R.R. Copley, B. Pils, S. Pinkert, J. Schultz, and P. Bork, "SMART 5: Domains in the Context of Genomes and Networks," Nucleic Acids Research, vol. 34, pp. D257-D260, January 2006.
[13] S.J. Wheelan, A. Marchler-Bauer, and S.H. Bryant, "Domain Size Distributions can Predict Domain Boundaries," Bioinformatics, vol. 16, pp. 613-618, July 2000.
[14] T. Lu, Y. Dou, and C. Zhang, "Fuzzy clustering of CPP family in plants with evolution and interaction analyses," BMC Bioinformatics, vol. 14, pp. S10, October 2013.
[15] Y. Chen, J. Xu, B. Yang, Y. Zhao, and W. He, "A novel method for prediction of protein interaction sites based on integrated RBF neural networks," Comput. Biol. Med., Vol. 42, pp. 402-407, January 2012.
[16] L. Liang, and P.L. Felgner, "Predicting antigenicity of proteins in a bacterial proteome; a protein microarray and naive Bayes classification approach," Chem. Biodivers, vol. 9, pp. 977-990, May 2012.
[17] F. Medina, S. Aguila, M.C. Baratto, A. Martorana, R. Basosi, J.B. Alderete, and R. Vazquez-Duhalt, "Prediction model based on decision tree analysis for laccase mediators," Enzyme Microb. Technol., vol. 52, pp. 68-76, November 2013.
[18] H. Sun, and S. Wang, "Penalized logistic regression for highdimensional DNA methylation data with case-control studies,"Bioinformatics, vol. 28, pp. 1368-1375, March 2012.
[19] M. Xin, W. Jiansheng, and X. Xiaoyun, "Identification of DNA Binding Proteins Using Support Vector Machine with Sequence Information," Computational Mathematical Methods in Medicine, vol. 1, pp. 524502, August 2013.
[20] N. Vinay, D. Monalisa, S.M. Sowmya, K.S. Ramya, and K. J. Valadi, "Identification of Penicillin-binding proteins employing support vector machines and random forest," Bioinformation, vol. 9, pp. 481-484, May 2013.
[21] Christian Vecchiola, Mani Abedini, Michael Kirley, Xingchen Chu, and Rajkumar Buyya **"**Gene Expression Classification with a novel coevolutionary based learning classifier system on Public Clouds**"** in Sixth IEEE International Conference on e–Science Workshops, IEEE, Vol. 6, pp 92- 97, 2010
[22] T. Hamp, F. Birzele, and F. Buchwald, "Improving structure alignment based prediction of SCOP families using Vorolign Kernels," Bioinformatics, vol. 27, pp. 204-210, November 2010.
[23] H.U. Kalsum, R.M. Othman, R. Hassan, S.M. Rahim, H. Asmuni, J. Taliba, and Z. Zakaria, "SplitSSI-SVM: an algorithm to reduce the misleading and increase the strength of domain signal," Computers in Biology and Medicine, vol. 39, pp. 1013-1019, November 2009.
[24] F. Sievers, A. Wilm, D. Dineen, T.J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J.D. Thompson, and H.G. Higgins, "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega," Molecular Systems Biology, vol. 7, pp. 539, October 2011.
[25] J. Eickholt, X. Deng, and J. Cheng, "DoBo: Protein domain boundary prediction by integrating evolutionary signals and machine learning," BMC Bioinformatics, vol. 12, pp. 1471, February 2011.

