Use of Extended Phylogenetic Profiles with E-Values and Support Vector Machines for Protein Family Classification

Kishore Narra and Li Liao*
University of Delaware, U.S.A.

Abstract

Protein family classification is an important means to assign functions to proteins, and use of phylogenetic profiles, which encode evolutionary history of proteins along with putative homologs, has proved to facilitate protein family classification. We proposed a new approach to compare phylogenetic profiles by incorporating the phylogenetic tree, from which the profiles are derived. Specifically, the profile is extended with new bits corresponding to the internal nodes of the tree, which encode the correlations among the bits in the original profiles. Such extension allows for direct use of E-Values, instead of imposing an ad hoc cut-off to derive binary profiles, which are commonly used in previous methods. A scoring scheme is adopted for measuring the similarity among these extended profiles, and the scores thus obtained are then provided to a classifier -- a support vector machine using a polynomial kernel function -- for classification. The method has been tested on the proteome of *Saccharomyces cerevisiae*, the budding yeast and outperformed a similar method that uses phylogenetic tree information as a tree kernel.

Keywords: classification, protein, support vector machines, phylogenetic profiles.

1. Introduction

Predicting protein functions remains a central task in computational biology. A vast number of computational tools [1,2,18] rely on sequence similarity to infer protein homology, which in turn leads to functional prediction: two homologous proteins evolved from a common ancestral protein are more likely to play the same functional role. Proteins that are remotely homologous to one another and therefore share less (below 30%) sequence similarity pose as a major challenge to many functional prediction methods, which solely rely on sequence information for making prediction. To detect remote protein homologues, various techniques have been developed, for example, iterative search with refined profiles [1], sophisticated probabilistic models, powerful statistical learning [10], and some hybrid approaches [7], to name only a few.

Some recent developments have attempted to utilize non-sequential information, either alone or in combination with sequence information, for protein functional prediction. For example, structural information was incorporated in profile hidden Markov models [6]. Some methods in comparative genomics went beyond homology for identifying proteins that are related to one another by participating in a common structural complex or metabolic pathways, or they are related because they fuse into a single gene in some genomes [4]. An important work along this line is the use of phylogenetic profiles for assigning gene functions based on evolutionary and/or co-evolutionary patterns across species [12, 14, 17]. The phylogenetic profile of a protein is represented as a vector, where each component corresponds to a specific genome and takes a value of either one or zero: with one (zero) indicating the presence (absence) of a significant homology of that protein in the corresponding genome. Similar hierarchical profiles have also been constructed from whole genome metabolic pathways, and utilized for comparing genomes based on their physiological characteristics and for clustering pathways [9, 21]. The simplistic approach to compare two profiles, which are often in binary format, is simply to count the number of matches and mismatches between the two profiles. Such approaches, although proved to be useful for prediction, apparently miss the information that is embedded in the profile, namely the hierarchical structure -- because of the correlations implied by the hierarchical structure not all matches (mismatches) are equal in telling how two genes are related. In [9], a methodology was suggested for incorporating the hierarchical structure in comparing profiles. A Bayesian based approach was developed recently in [20] to utilize the phylogenetic tree for constructing kernel function of support vector machines that are used for predicting functions of proteins based on their phylogenetic profiles.

In this paper, we proposed a novel approach to extracting information embedded in hierarchical, specifically phylogenetic, profiles, and demonstrated that the extracted information, in concatenation with the original profiles, enabled more efficient learning for support vector machines. The scheme of extending the original profiles works as well, and actually even better, when the profiles use real value numbers, such as E-values. The method shows a significant improvement for functional predictions of proteins than just by using the

* Corresponding author: Department of Computer and Information Sciences, 103 Smith Hall, Newark, DE 19716, USA. lliao@cis.udel.edu.

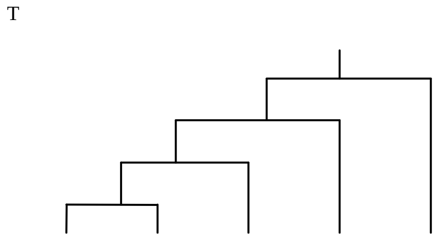
plain phylogenetic profiles, and it also outperforms the Bayesian based tree kernel method in [20].

2. Methods

2.1. Tree encoded profiles

The phylogenetic profile of a protein is represented as a vector, where each component corresponds to a specified genome and takes a value of either one or zero: with one (zero) indicating the presence (absence) of a significant homology of that protein in the corresponding genome. The similarity of these profiles can be used to detect protein homology; since proteins that tend to evolve in a coordinated way and thus have similar phylogenetic profiles. In this study, a group of 24 complete genomes is used to construct phylogenetic profiles for all proteins in Yeast [15].

The Hamming distance between a pair of phylogenetic profiles is perhaps the most straightforward way to measure the similarity. Yet, when correlation exists among the components in a vector, the Hamming distance becomes inadequate. For example, shown in Figure 1 are a phylogenetic tree of five species and three derived profiles x = (0, 1, 1, 1, 1), y = (1, 1, 1, 1, 1), z = (1, 1, 1, 1, 0). The Hamming distance d(x, y) = -1+1+1+1+1 = 3, where the minus one is contributed from the mismatch between x and y at the first position. Similarly, the Hamming distance d(y, z) = 1+1+1+1-1 = 3. However, using biological intuition, one would suspect that y and z should be farther apart since they mismatch at the fifth position, which corresponds to an attribute directly descendent from the root and consequently should be weighted more.



X =	0	1	1	1	1
Y =	1	1	1	1	1
Z =	1	1	1	1	0

Figure 1. A phylogenetic tree of five species and three phylogenetic profiles derived from this tree.

In this work, we propose a novel method to compare hierarchical profiles, which addresses both knowledge representation and efficient learning. To capture the information encoded in the hierarchical structure (a phylogenetic tree in this case) of a profile, a two-step procedure is adopted: 1) a score is assigned at each internal tree node; 2) the score labeled tree is then flatten into an extended vector. For an internal tree node in a phylogenetic tree, as it is interpreted as ancestor of the nodes underneath it, one way to assign a score for it is to take the average of the scores from its children nodes. This scoring scheme works top-down recursively until the leaves are reached: the score at a leaf is just the value of the corresponding component in the hierarchical profile. The same scoring scheme was first suggested in [9] to compare two phylogenetic trees by the thus obtained scores at the root of each tree. Unlike [9], where only the score at the root node was used, naturally suffering from certain information loss, here we instead retain the scores at all internal nodes: mapping them into a vector via a post-order tree traversal and concatenating this vector with the original profile vector to form an extended vector, which we call tree-encoded profile. For example, given a two-component vector <a, b>, where a and b correspond to two genomes and have a parent node c, our two-step procedure will first assign a score (a+b)/2 for node c, and then generate as extended vector as <a, b, (a+b)/2>. The newly added component will help enhance the similarity among the two-component vectors where (a+b)/2 is equal. For example, when profiles <0.3, 0.7>, <0.4, 0.6> and <0.2, 0.8> are extended, they become <0.3, 0.7, 0.5>, <0.4, 0.6, 0.5> and <0.2, 0.8, 0.5> respectively. Note that the values for the expanded components are real number in the range [0, 1].

As the extended profiles include real-value numbers, we can also use real-value numbers in the original profiles, whose binary values are derived by imposing a cutoff on E-values from BLAST search (see Section 2.3 for details). Obviously, by using E-values directly, we could avoid the loss of information this is incurred when converting to binary values. However, since E-values can not generally be considered as metric distance, in next section, we devise an ad hoc scoring scheme based on the E-value distribution.

A further refinement is attained by introducing weights in calculating the average score for an internal tree node. The weights are collected as the frequency of presence and absence occurring in different tree leaves. For example in Figure 2, the frequency for a presence is 67% at the left most position, and is 100% at the next position. Apparently, this only works when the original profiles are binary. In order to combine the benefits of using E-values in the original profiles and using weights for calculating the extended part, a threshold (of value 1) is used only for the purpose of collecting the weights. Once the weights at the tree leaves are counted, the E-values will be used to populate the internal nodes. In the result section, it is shown that such extended profiles, referred to as "TEEWP" in Figure 3, achieve the best performance.

2.2. Kernel function

With the tree-encoded profiles as input, a support vector machine using a polynomial kernel is utilized to classify proteins for different functional categories or families.

As a powerful statistical learning method, support vector machines (SVMs) [19] have recently been applied broadly to many problems in computational biology [13], including remote protein homology detection, microarrray gene expression analysis, protein secondary structure prediction, and problems in other domains [3] such as face detection and text categorization. The power of SVMs comes partly from the data representation, where an entity (e.g., a protein) is represented by a set of attributes instead of a single score. As those attributes may not be equally representative in distinguishing a true positive from a true negative, the boundary line between the two classes, if depicted in a vector space, can be highly nonlinear. The SVMs method will find a nonlinear mapping embodied as a kernel function such that the data can then be linearly separable in a higher dimensional space called feature space. The actual learning power of SVMs lies in the kernel function, which defines how to measure the "distance" in the feature space between two points by using their values in the original space called input space.

The polynomial kernel used in this work is defined for vectors x and y as

$$K(x, y) = [1+ s\, D(x, y)]^d$$

where s and d are two adjustable parameters. Unlike ordinary polynomial kernels, D(x, y) is not the dot product of vectors x and y, but rather, a generalized Hamming distance for real value vectors:

$$D(x, y) = \Sigma_{i=1 \text{ to } n} (S(|x_i-y_i|)$$

where the ad hoc function S has value 7 for a match , 5 for a mismatch by a difference less then 0.1, 3 for a mismatch by a difference less than 0.3, and 1 for a mismatch by a difference less than 0.5. As mentioned before, the values for function S are assigned based on the E-value distribution of the protein dataset. So, if another dataset is used, these score values may be slightly different, depending on the E-value distribution. Thus customized kernel function is allowed for and actually can be conveniently implemented in the software package SVM Light [8] used in this work.

To test our method, we compiled three variations: 1) TEAHP, that encodes the tree, uses ad hoc function S, and polynomial kernel; 2) TEWP, that encodes the tree by using weights, and uses polynomial kernel; and 3) TEEWP, that encodes the tree with weights, takes the E-values directly, and uses polynomial kernel. We compared these variations of our method with a linear kernel and a tree kernel reported in [20]. As a baseline for evaluating the utility of the tree encoding, we also extend the original profile simply by adding randomly assigned values, in the range of [0, 1] into each extended bit. Classification based on such randomly extended profiles is referred as "rand14_teahp" in Figure 4.

2.3. Data

The data set used in this work is the same data set as in [14, 17]. Proteins (or the genes encoding these proteins) with accurate functional classifications were selected from the budding yeast *Saccharomyces cerevisiae* genome. To ensure adequate training and testing examples, only the functional classes that contain at least 10 genes were extracted from the several hundred classes in the Munich Information Center for Protein Sequences Comprehensive Yeast Genome Databases [15]. The resulting dataset contains 2465 genes in 133 classes. The binary profiles of these genes were built by BLAST search against each of the 24 genomes. Each bit in the profile for a gene was set to 0 or 1 if the E-value of the BLAST search for the gene against the corresponding organism was larger or smaller than 1 respectively. The phylogenetic tree of these 24 genomes is the same as in [20], and is used to obtain tree-encoded profiles, which are 38 bit vectors, with the last 14 bits corresponding to the internal nodes.

A 3-fold cross validation was adopted for the experiments. For each functional class, two third of its members are randomly selected as positive training examples, and the rest one third as positive testing examples. Genes not belonging in that class were randomly split into two thirds as negative training and one third as negative testing examples.

3. Results

The results of the experiments are summarized in Figures 3, 4, and 5. The function prediction for each class is measured by its receiver operating characteristic (ROC) score. ROC score is the normalized area under a curve that plots the true positives as a function of false positives for varying classification thresholds [5]. ROC50 scores are ROC scores that are calculated by integrating the area up to the first 50 false positives. A curve in Figures 3 and 4 is a histogram of ROC50 scores for 133 classes, averaged over 50 random runs for a function prediction method. Each curve shows the number of classes (Y-axis) that the respective method performs better than a given ROC50 score (X-axis). Therefore, a higher curve indicates more accurate prediction performance. As demonstrated in Figures 3 and 4, our method using the tree-encoded E-value based profile and polynomial kernel (TEEWP) has the best performance among the various methods tested here. In particular, it is worth noting that our method outperformed the tree kernel method reported in [20], not only with a better prediction accuracy, but also significantly faster. In Figure 5, a class-by-class comparison of ROC50 scores from TEEWP and Tree Kernel methods is displayed. We hypothesize that the superior performance of our method derives mainly from our better way of capturing and representing the correlations existed among various bits of the original profile. To validate this hypothesis, we had just randomly extended the original profile by 14 bits, and then trained on the same dataset using the generalized polynomial kernel SVM. The results were reported in Figure 4 and it is easy to notice that the histogram curve of ROC scores is much worse than our method's.

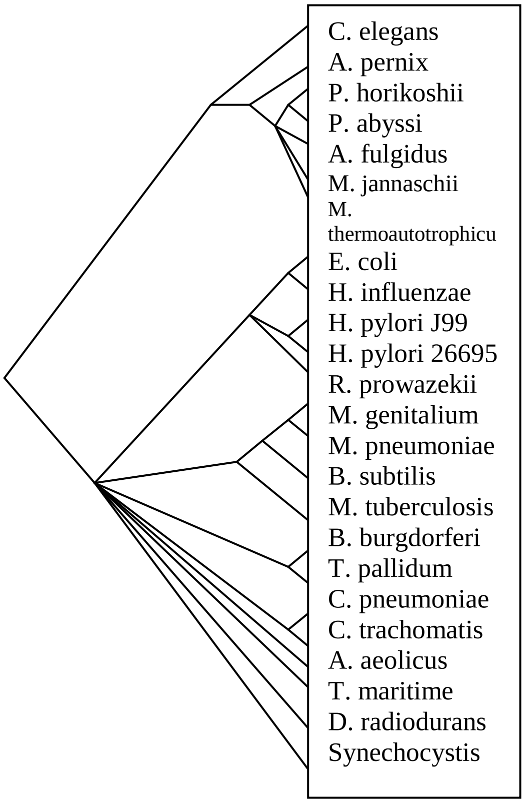


Figure 2. The 24 genomes and the phylogenetic tree of these 24 genomes.

4. Discussion

A novel approach was proposed in this work for extracting information that is embedded in hierarchical, specifically phylogenetic, profiles. It was demonstrated that the extracted information, in concatenation with the original profiles, enabled more efficient learning for support vector machines, leading to a significant improvement for functional predictions of proteins than by just using the plain phylogenetic profiles.

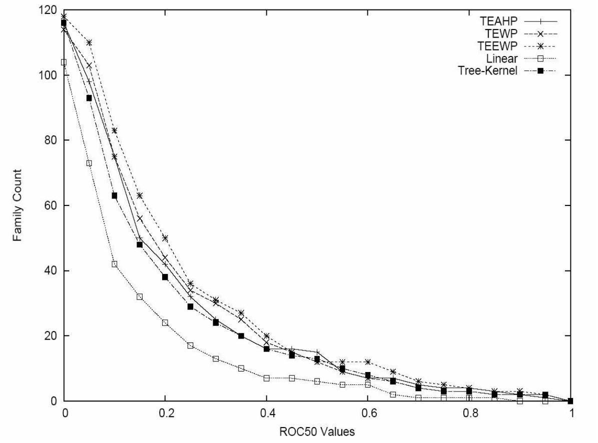


Figure 3. Histograms of ROC50 scores for various methods on 133 functional classes. TEAHP kernel refers to the method presented in this paper, and Linear kernel and Tree kernel refer to two methods reported in [17].

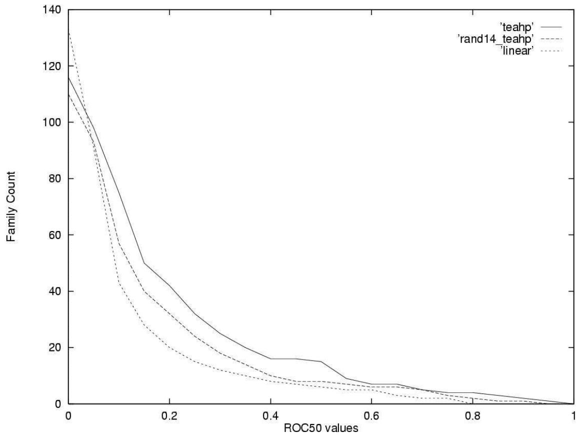


Figure 4. Histogram of the ROC50 scores. "rand14_teahp" refers to extending the phylogenetic profiles by 14 random bits.

Our method also performed better than a tree kernel method that involved more sophisticated Bayesian analysis and probabilistic assumptions, which are ad hoc and sometimes causing some type of data unusable. For example, while it is intuitive to assign prior probabilities for ones and zeros in a binary profile when they are interpreted respectively as presence and absence of some events, it would be very difficult to do so for real value profiles, e.g., profiles that contain e-values directly from BLAST search. Our method, without resorting to assigning prior probabilities, can be readily applied to real value profiles.

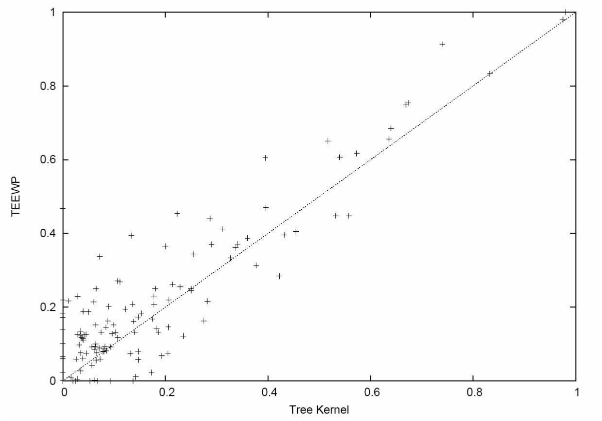


Figure 5. Class-by-class comparison of ROC50 scores from TEEWP and Tree Kernel methods. Each point in the plot corresponds to a single MIPS functional class, out of 133 classes used in this study.

Acknowledgements The authors thank Dr. Chandra Kambhamettu for useful discussions. The work is partly funded by a DuPont Science and Engineering Award. L.L. is grateful for an International Travel Award from the Center for International Studies at the University of Delaware supporting his trip to the SNPD conference where part of the work was presented.

References

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool", *Journal of Molecular Biology*, vol. 215, pp. 403-410, 1990.

[2] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped blast and psi-blast: A new generation of protein database search programs", *Nucleic Acids Research* vol. 25, pp. 3389-3420, 1997.

[3] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based learning methods*. Cambridge University Press, Cambridge, UK, 2000.

[4] A.J. Enright, I. IIiopoulos, N.C. Kyrpides, and C.A. Ouzounis, "Protein interaction maps for complete genome based on gene fusion events", *Nature*, vol. 403, pp. 86-90, 1999.

[5] M. Gribskov, and N. Robinson, "Use of receiver operating characteristic analysis to evaluate sequence matching", *Computers and Chemistry*, vol. 10, pp. 25-33, 1996.

[6] S. Griffiths-Jones, and A. Bateman, "The use of structure information to increase alignment accuracy does not aid homologue detection with profile HMMs", *Bioinformatics*, vol. 18, pp. 1243-1249, 2002.

[7] T. Jaakola, M. Diekhans, and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies", Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology. 1999, pp. 95-114.

[8] T. Joachims, "Making large-scale svm learning practical", Advances in kernel Methods – Support Vector Learning, Scholkopf, B., Burges, C., and Smola A. (eds), MIT Press, 1999. pp. 169-184.

[9] L. Liao, S. Kim, and J.F. Tomb, "Genome Comparisons Based on Profiles of Metabolic Pathways", The *Proceedings of The Sixth International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES2002)*, , September 2002, Crema, Italy, pp. 469-476.

[10] L. Liao, and W.S. Noble, "Combining pairwise sequence similarity and support vector machines for remote protein homology detection", The *Proceedings of The Sixth International Conference on Research in Computational Molecular Biology (RECOMB 2002)*, April 2002, pp225-232.

[11] L. Liao, and W.S. Noble, "Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships", *Journal of Computational Biology*, vol. 10, pp. 857- 868, 2003.

[12] D. A. Liberles, A. Thoren, G. vonHeijne, and A. Elofsson, "The use of phylogenetic profiles for gene predictions", *Current Genomics*, vol. 3, pp. 131-137, 2002

[13] Noble, William Stafford. "Support vector machine applications in computational biology." *Kernel Methods in Computational Biology*. B. Schoelkopf, K. Tsuda and J.-P. Vert, ed. MIT Press, 2004. pp. 71-92.

[14] Marcotte, E.M., Pellegrini, M., Thompson, M.J., Yeates, T.O. and Eisenberg, D. "A combined algorithm for genome-wide prediction of protein function", *Nature*, vol. 402, pp. 83-86, 1999.

[15] H.W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkoetter, S. Rudd, and B. Weil, "MIPS: a database for genomes and protein sequences", *Nucleic Acids Research,* vol. 30, pp. 31-34, 2002.

[16] P. Pavlidis, J. Weston, J. Cai, and W.N. Grundy, "Gene functional classification from heterogeneous data", Proceedings of the Fifth International Conference on Computational Biology. pp. 249-255.

[17] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, and T.O. Yeates, "Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles", *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 4285-4288, 1999.

[18] T.F. Smith, and W.S. Waterman, "Identification of common molecular subsequences", *Journal of*

Molecular Biology, vol. 147, pp. 195-197, 1981.

[19] V.N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

[20] J.P. Vert, "A tree kernel to analyze phylogenetic profiles", *Bioinformatics*, vol. 18 pp. S276-S284, 2002.

[21] S. Zhang, L. Liao, J.F Tomb, and J.T.L. Wang, "Clustering and classifying enzymes in metabolic pathways: some preliminary results", *ACM SIGKDD Workshop on Data Mining in Bioinformatics (BioKDD2002)*, 2002, pp19-24.

Kishore Narra is a graduate student in The Department of Computer and Information Sciences, University of Delaware, and has been doing research in bioinformatics for two years. He received a Bachelor of Engineering degree from Osmania University in India.

Li Liao received his Ph.D. degree from Peking University, and is currently an assistant professor in the Department of Computer and Information Sciences, University of Delaware. His research interests and experience span a wide range, including computer simulation of molecular systems, genome sequencing, protein homology detection, and genome comparisons. He has published 25 peer-reviewed scientific papers and co-authored one book. He is an ACM member and a member of the International Society for Computational Biology. His research is funded by grants from the NIH, the US Army, and DuPont Company. He received Accomplishment Award in 2001 and 2002 from DuPont Central Research and Development, where he worked as a senior research physicist from 1998 to 2000.