Tree-*k*NN: A Tree-Based Algorithm for Protein Sequence Classification

Khaddouja Boujenfa
Laboratory of Operational Research Decision and Process Control
41 rue de la Liberté, Le Bardo
Tunis, Tunisie
Khaddouja.Boujenfa@isg.rnu.tn

Nadia Essoussi
Laboratory of Operational Research Decision and Process Control
Department of Computer Science, ISG
41 rue de la Liberté, Le Bardo
Tunis, Tunisie
Nadia.Essoussi@isg.rnu.tn

Mohamed Limam
Laboratory of Operational Research Decision and Process Control
Department of Computer Science, ISG
41 rue de la Liberté, Le Bardo
Tunis, Tunisie
Mohamed.Limam@isg.rnu.tn

***Abstract*— The phylogenomic classification of protein sequences attempts to categorize a given protein within the evolutionary context of the entire family. It involves mainly four steps: selection of homologous sequences, multiple sequence alignment, phylogenetic tree construction and tree-based classification. This supposes that the tree used as a basis of protein classification is correct. Sequence alignment is the first step for tree construction. Thus, the accuracy of the alignment produced should affect the topology of the phylogenetic tree. This work proposes a kNN tree-based algorithm for protein classification, namely Tree-kNN, which uses a phylogenetic tree estimated from pair-wise and multiple alignment approaches. We compare the classification performance of Tree-kNN with an existing method, called TreeNN. Results show that Tree-kNN gives better results than TreeNN. Based on four datasets we show that classification performances of the two algorithms using pair-wise alignment are better than using multiple alignment.**

***Index Terms*—Pair-wise alignment, multiple alignment, protein classification, kNN classifier, similarity measures.**

I. INTRODUCTION

Protein classification is one of the fundamental and traditional problems in bioinformatics. As outlined by Busa-Fekete et al. [1], three main categories of classification methods can be identified. Sequence comparison is the most commonly used approach for protein classification. In this framework, a protein is compared against other proteins in a database, and if a sequence can be detected whose similarity is statistically significant, the class of the unknown protein is inferred based on the known class of the similar sequence. When distant sequence similarities are observed in a protein database, methods based on consensus descriptions are most efficient. For all the classes of a protein sequence database, a consensus description is prepared. As the previous method, the query protein is compared to each of the consensus description and is assigned the class label with the highest similarity.

A more recent type of protein classification is called phylogenomics and is, originally, outlined by Eisen [2]. Phylogenomics does not just rely on the similarities in sequences, but it also considers the phylogenetic information stored in a tree. This external source of knowledge is accumulated in the fields of taxonomy and molecular phylogeny and is the basis of protein classification. The phylogenomics approach attempts to overcome the systematic errors associated with sequence comparison tools and increase the classification performance.

Effectively, sequence comparison tools are routinely used by researchers in protein classification and systematic errors associated with these tools have been pointed out by numerous studies [3], [4], and [5]. Gene duplication is the greatest factor of function diversity observed in protein super-families as well as to errors in protein classification by sequence comparison. When gene duplication occurs, one copy supplies the original function, while the other is allowed to evolve novel functions. Paralogs and orthologs are homologous proteins, which share a common ancestry. Paralogous proteins, related by duplication events, share high sequence similarity and are more likely to have divergent function while orthologous proteins, related by speciation, have divergent sequence and are more likely to share a common function. Thus, the protein hit given by sequence comparison tools does not, usually, share the same function of the query protein, despite the high sequence similarity observed. Domain shuffling [6] and [7] is another contributing factor to errors in protein classification. In fact, sequence comparison tools usually disregard whether proteins align globally or locally. Thus, the absence or presence of domains has a great impact in protein classification. Finally, existing errors in protein sequence databases are propagated through function prediction by sequence comparison.

The rationale behind applying tree-based algorithms is to disambiguate the relationship between paralogs and orthologs and to provide a structure description that is simple and computationally inexpensive, but still may allow one to exceed the performance of simple sequence comparison algorithms such as BLAST [8]. The phylogenomic approach is a multistep process involving mainly, 1) the detection of homologous sequences, 2) multiple sequence alignment of these homologos, 3) phylogenetic tree construction and 4) classification of the unknown protein sequence based on this tree. Multiple sequence alignment is a fundamental step, which affects the topology of the tree produced. Busa-Fekete et al. [1] have proposed a tree-based method for protein classification, namely TreeNN, which shows higher performance than sequence comparison tools. TreeNN compare homologous sequences using an all versus all pair-wise comparisons instead of multiple sequence alignment of the whole homologs. Thus, a question asked here is to know if pair-wise alignment approach provides a better classification performance for tree-based classification than multiple sequence alignment.

This paper proposes a method, namely Tree-*k*NN, for protein sequence classification based on phylogenetic tree. Tree-*k*NN uses the principles of *k*NN classifiers on a phylogenetic tree to find the closest neighbour of a query protein and infer its class label. Section II describes the tree-based algorithms, Tree-*k*NN and TreeNN and their implementations. Section III gives the different datasets and the alignment programs used. Section IV compares, in one hand, the classification performance of our algorithm with TreeNN, and on the other hand, evaluates pair-wise and multiple alignment approaches on classification performance. In the last section, we discuss the different results obtained by our method and give some perspectives to our work.

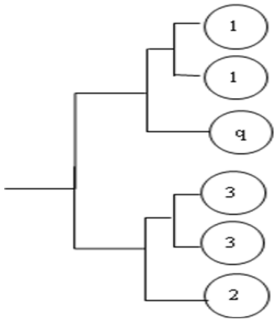

Figure 1. A phylogenetic tree of proteins overlaid with class labels. There are three different classes. Two proteins with class label 1, two with class label 3 and one protein with class label 2. The query protein is q.

II. TREE-KNN: PROTEIN CLASSIFICATION USING A WEIGHTED BINARY TREE

The algorithm described in this work belongs to the broad area of protein classification supported by phylogenetic information, and termed phylogenomics. The phylogenomic classification of an unknown protein starts with the identification of homologous proteins for the protein of interest. A multiple sequence alignment is constructed for the cluster of homologous proteins and a phylogenetic tree is inferred. The tree topology is analyzed to discriminate between paralogs and orthologs. Finally, the tree is overlaid with experimental data and used as basis for protein classification. Every step in this process is more prone to error when applied large and divergent protein super-families than when applied to smaller and more closely related protein families. For closely related taxa, alignment accuracy can be expected to be high, with corresponding increased classification performance in the resulting tree topology. However, when large numbers of divergent sequences are included, alignment and tree topology can be expected to decrease.

A. Algorithm Description

We compared a given database of a priori classified proteins and a query protein. The algorithm first constructs a common tree that includes the member of the database and the query protein. In the following step the algorithm attempts to assign class label to an unknown protein using the known class labels found in its neighbourhood within the tree.

A weighting schema of similarity/dissimilarity measure is applied. The class label of the neighbour protein with the highest similarity weight, respectively lowest dissimilarity is assigned to the query.

The weighted binary tree is built, which contains proteins in each leaf. We assign the known class labels to the proteins; all leaves except the unknown query will be labelled. Figure 1 shows a binary tree of proteins.

Tree-*k*NN algorithm is a weighted nearest neighbour method that applies as weight a dissimilarity measure (such as BLAST e-value) between the proteins constituting the tree.

We denote the length of the path between two leaves L_i and L_j by $p\ (L_i, L_j)$. Here p is an integer representing the number of edges along the path between L_i and L_j. We define the closest neighbourhood $Nei\ (L_q)$ of a query protein L_q as the set of leaves, which have the minimum number of edges from the query. For example, the closest neighbourhood of the query protein q in Fig. 1 are two members of class 1 where $p = 3$.

We assume n leaves from m different classes. An indicator function, I_f assigns class labels to the proteins represented by the leaves of the tree $I_f :\{ L_i... L_m\} \rightarrow \{1... m\}$ and $j \in \{1 ... m\}$. The distance $d\ (L_i, L_q)$ represents the dissimilarity measure between a given protein L_i and the query protein L_q.

From the set of the closest neighbourhood $Nei\ (L_q)$, Tree-*k*NN finds the neighbour protein L_i, which has the lowest dissimilarity measure. The query protein L_q is assigned the class label of the neighbour L_i as given by:

$$D\left(j, L_q\right) = \underset{L_i \in Nei\left(L_q\right) \wedge I_f\left(L_i\right) = j}{Min} d\left(L_i, L_q\right). \tag{1}$$

If we consider a similarity measure, such as a BLAST score, to compare protein sequences, the formula changes as follows:

$$S\left(j, L_q\right) = \underset{L_i \in Nei\left(L_q\right) \wedge I_f\left(L_i\right) = j}{Max} s\left(L_i, L_q\right). \tag{2}$$

B. Tree-based Method: TreeNN

The performance of Tree-*k*NN is compared to a tree-based algorithm, namely, TreeNN. As Tree-*k*NN, TreeNN is a weighted nearest neighbour method, which takes weights from a distance matrix of all pair-wise comparisons of protein sequences, construct a phylogenetic tree and infer the class label of a query protein based on the tree neighbourhoods of the query. The tree neighbourhoods, as defined by TreeNN, are proteins which have the minimum number of edges from the query protein in each class. Thus, from each class, TreeNN determines a different number of neighbourhoods. Instead, Tree-*k*NN takes only the first closest neighbour without considering all the classes. For example, the closest neighbourhood given by TreeNN for the query protein q in Fig. 1 are two members of class 1 (p = **3**), one member of class 2 (p= 4) and two members of class 3 (p= 5). Then, if a similarity measure is considered as edge weight, TreeNN assigns to the query protein the class label with the highest aggregate similarity measure given by:

$$R\left(j, L_q\right) = \underset{L_i \in Nei\left(L_q\right) \wedge I_f\left(L_i\right) = j}{\Theta} s\left(L_i, L_q\right), \tag{3}$$

where the aggregation operator Θ can be the sum, product or maximum.

If we consider a dissimilarity measure as edge weight, then the formula given in (3) can be transformed as follows:

$$R'\left(j, L_q\right) = \underset{L_i \in Nei\left(L_q\right) \wedge I_f\left(L_i\right) = j}{\Theta} d\left(L_i, L_q\right), \tag{4}$$

where the aggregation operator Θ can be the minimum.

Additionally, as described by the authors, TreeNN compares protein sequences using an all vs. all pair-wise alignments instead of multiple alignment. Thus, this work tests, in one hand, the classification performances of Tree-*k*NN and TreeNN algorithms, and on the other hand, evaluates the classification results of the two algorithms on each alignment approach.

C. Implementations

The different steps of the Tree-*k*NN and TreeNN algorithms are described as follows:

1- Compute a distance matrix (from BLAST or ClustalX alignments) containing the all vs. all comparisons of a dataset, which consists of a query protein and an a priori classified dataset.
2- Construct a Neighbor-Joining tree from the distance matrix of Step 1.
3- Based on the tree structure of Step 2, the query protein is assigned a class label using Tree-*k*NN and TreeNN algorithms in the way described in Section 2.1 and 2.2, respectively.

For the calculation of the distance matrices in Step 1, we have applied the scoring scheme based on the dissimilarity measures. Using BLAST, we have considered the E-value distances with a cut-off of 10. When using ClustalX with default parameters, the alignment result is given to protdist program from the Phylip package 3.6 [9] to generate the corresponding distance matrix. Based on these matrices, Tree-*k*NN and TreeNN algorithms classify a query protein using the scoring scheme given in (1) and (4), respectively.

III. DATA SETS AND METHODS

A. Data sets and Classification Tasks

To assess the performance of our algorithm, we have used datasets with different degrees of sequence similarity. The Protein Classification Benchmark collection [10] provides standard datasets for testing machine learning methods. The collection contains datasets of sequences and structures with different classification tasks. A classification task is the subdivision of a dataset into positive train (+train), positive test (+test), negative train (-train) and negative test (–test) groups. Given such a subdivision, one can train a classifier and evaluate its performance. Here, we used protein sequences from 3PGK, COG, CATH95 and SCOP95 datasets described, respectively, in the following sections.

- Dataset#1: 3PGK

This dataset is constructed from the 3-phosphoglycerate kinase (3PGK) protein sequences, which represent various species of the Archaea, Bacteria and Eukaryota kingdom. The classification tasks are defined as follows. The positive set is taken from a given kingdom. One of the phyla, with at least 5 members is the +test while the remaining phyla of the kingdom is the +train. The negative set contains members of the other two kingdoms divided in such a way that members of one phylum can be either –test or –train.

- Dataset#2: COG

This is a subset of the Clusters of Orthologous Groups database (COG) [11]. Each COG cluster contains functionally related orthologous sequences belonging to prokaryotic and unicellular eukaryotic. The classification tasks are defined as follow. Only COG groups with at least 8 eukaryotic and 16 prokaryotic members were selected. The positive set is taken from a given COG cluster subdivided into eukaryotes (+test) and prokaryotes (+train). The rest of the COG database is divided in such a way that members of a COG group can be either –test or –train.

- Dataset#3: CATH95

The dataset is created from the protein sequences of CATH database [12] with sequence identity greater than 95%. The CATH database is a hierarchical classification of protein domain structures. The classification is achieved via a semi-automatic procedure. The protein domains are classified into four main levels: protein class (C), architecture (A), topology (T) and homologous superfamily (H) groups, based on similarity (S) groups.

The classification tasks are defined on this dataset in the following way. The positive set is created from a given H group. One of the S groups, with at least 5 members and at least 10 members outside the S group but within the same H group, is the +test. The remaining S groups of the selected H group are the +train. The rest of the dataset outside the H group is first divided in such a way that members of an S group can be either –test or –train. Then 10% of the resulting two sets where randomly selected to give the final -train and –test.

- Dataset#4: SCOP95

This dataset is constructed from the protein sequences of SCOP database (Structural Classification Of Proteins) [13] with sequence identity less than 95%. SCOP provides a detailed and comprehensive description of the structural and evolutionary relationships between proteins, and is constructed manually by visual inspection and comparison of structures. Like CATH, there is a hierarchy of four main levels of classification: class, fold, superfamily and family.

The classification tasks are defined in the following way. The positive examples are taken from a given superfamily. One of the families, with at least 5 members and at least 10 members outside the family but within the same superfamily, is the +test while the remaining families of the superfamily are used as the +train. The negative set contains members of the other superfamilies and is divided in such a way that members of a family can be either –train or –test. 10% of the resulting two sets where randomly selected to give the final -train and –test.

TABLE I. BENCHMARK DATASETS

Dataset Name	*No. of sequences*	*Average Seq. Length*	*PID*
3PGK	131	411	55.63
COG	3239	352	55.53
CATH95	1263	147	35.48
SCOP95	1284	170	32.52

The benchmark represents various degrees of difficulty. The sequences in orthologous groups of the COG database are closely related to each other within the group, while there are relatively weak similarities between the groups. Protein families of SCOP or homology groups of CATH are less closely related to each other in terms of sequence similarity and the similarities between groups are also weak. Sequences in the 3PGK dataset, divided into taxonomic groups represent a case where both the within-group and between-group similarities are high. The details of the datasets are described in Table I. The classification tasks used are shown in Table II.

TABLE II. BENCHMARK CLASSIFICATION TASKS

Dataset Name	*ID*	*+test*	*+train*	*-test*	*-train*
3PGK	Archaea_Crenarchaeota	4	11	53	63
COG	[J] COG0008 Glutamyl- and glutaminyl-tRNA synthetases	8	103	1601	1527
CATH95	1.10.10.10._1.10.10.10.103.	5	137	554	567
SCOP95	a.1.1._a.1.1.1.	5	97	598	584

Values indicate the number of proteins in each set.

B. Alignment programs

Sequence alignment approaches are classified as either global or local and pair-wise or multiple. Global alignment methods optimize the overall alignment of sequences, which may include large stretches of low similarity. Local similarity algorithms seek only relatively conserved subsequences, and a single comparison may yield several distinct subsequence alignments. Pair-wise alignment methods compare two sequences, while multiple alignment take three or more sequences. In order to test the influence of the alignment approach in classification performance, we used BLAST 2.0 and ClustalX 2.0.11 [14] for pair-wise and multiple alignment of protein sequences, respectively.

- BLAST

BLAST (Basic Local Alignment Search Tool) is a widely used tool for searching protein databases for sequence similarities. BLAST is based on pair-wise comparison that approximates alignments, which optimize a measure of local similarity, the maximal segment pair (MSP) score. Blast compares a query protein with a database of sequences, and identifies library of sequences with locally MSP that resemble the query above a certain threshold.

- ClustalX

ClustalX is a windows interface for the widely-used progressive multiple sequence alignment program ClustalW [15]. It is a progressive alignment method, which aims to find a multiple global alignment. The method generates all pair-wise sequence alignments in order to compute a distance matrix showing the divergence of each pair of sequences. Then, a tree is constructed from the distance matrix in order to guide the final multiple alignment. Finally, a progressive procedure is used in order to align the sequences following the branching order of the guide tree. At this stage, the pairs of sequences are aligned from the tips of the rooted tree towards the root.

C. Tree-construction method

There are two main classes of phylogenetic tree construction methods: distance-based method such as Neighbor-Joining [16] and character-based methods such as maximum parsimony, maximum likelihood and Bayesian approaches. Distances based methods, compute a matrix of pair-wise distances between sequences in an alignment, and thereafter, a greedy algorithm predicts an evolutionary tree based on progressively adding the next most-alike sequence, or set of sequences, as an additional branch to an existing tree. The computational advantage of distance-based methods over character-based methods makes them more popular and also amenable to bootstrap analysis [17] for very large trees. In this work, we used Neighbor-Joining method from the Phylip package 3.6 to construct phylogenetic trees.

IV. RESULTS

A. Assessment of classification performance

To assess the classification performance, we calculate the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) as follows:

- TP: the number of proteins predicted to belong to positive class and the actual value is positive.
- FP: the number of proteins predicted to belong to positive class and the actual value is negative.
- FN: the number of proteins predicted to belong to negative class and the actual value is positive.
- TN: the number of proteins predicted to belong to negative class and the actual value is negative.

The following measures are used to assess the performance of Tree-*k*NN and TreeNN using, in one hand, BLAST program, and on the other hand, the ClustalX program. The error rate is given by

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN}, \qquad (5)$$

it measures how many errors are made when query proteins are classified. The accuracy given as follows

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (6)$$

gives the proportion of correctly assigned proteins.

B. Comparison of Tree-kNN versus TreeNN

The performance of Tree-*k*NN is evaluated by error and accuracy calculations in the way described in the previous section. For comparison we have also include the results obtained by a given tree-based algorithm namely, TreeNN.

The evaluation results of Tree-*k*NN and TreeNN on the four datasets are given in Table III. The best scores are given in bold. When using ClustalX program, we note that our algorithm outperforms TreeNN on all datasets used and with respect to all criteria. When protein sequences are closely related as in the 3PGK dataset, Tree-*k*NN recognizes all actual positives and negatives. However, TreeNN generates slightly higher error rate and lower accuracy than Tree-*k*NN. For less closely related protein sequences, Tree-*k*NN outperforms TreeNN with 3.91%, 8.41% and 21.4% lower classification errors and higher accuracy than TreeNN on COG, CATH95 and SCOP95 datasets, respectively.

When using BLAST alignment program on 3PGK and COG datasets, Tree-*k*NN shows slightly better results than TreeNN. However, on CATH95 and SCOP95 datasets, the performance of TreeNN is slightly better than Tree-*k*NN with 4.11% and 0.67% difference with respect to error rate and accuracy, respectively.

A. Comparison of BLAST versus ClustalX

Protein sequence alignments using BLAST or ClustalX program give different distance matrices. Thus, estimated phylogenetic trees from these alignments are also different. This motivates our work to evaluate the performance of each tree-based algorithm using different alignment approaches.

Using BLAST or ClustalX program changes the classification results. Our aim is to show which alignment program gives the best classification performance. Table IV show the results of Tree-*k*NN and TreeNN algorithms using BLAST and ClustalX. We can denote that the classification results using BLAST are better than those using ClustalX, with respect to all criteria.

For closely related sequences as given in the 3PGK dataset, pair-wise and multiple alignment approaches work similarly. However, on COG, CATH95 and SCOP95 datasets, BLAST program shows higher performance than ClustalX. With Tree-*k*NN, BLAST gives 0.13%, 9.3% and 0.49% better error rate and accuracy than ClustalX on COG, CATH95 and SCOP95 datasets, respectively. In addition, with TreeNN, BLAST gives better performance than ClustalX with 3.97%, 21.82% and 22.56% lower error rate and higher accuracy on COG, CATH95 and SCOP95 datasets, respectively.

TABLE III. CLASSIFICATION PERFORMANCES OF TREE-KNN AND TREENN USING CLUSTALX AND BLAST ON THE FOUR DATASETS

	Tree-kNN				**TreeNN**			
	ClustalX		**Blast**		**ClustalX**		**Blast**	
	Error rate	**Accuracy**	**Error rate**	**Accuracy**	**Error rate**	**Accuracy**	**Error rate**	**Accuracy**
3PGK	**0.00**	**100.00**	**0.00**	**100.00**	1.75	98.25	1.75	98.25
COG	**0.25**	**99.75**	**0.12**	**99.88**	4.16	95.84	0.19	99.81
CATH95	**18.96**	**81.04**	9.66	90.34	27.37	72.63	**5.55**	**94.45**
SCOP95	**3.81**	**96.19**	3.32	96.68	25.21	74.79	**2.65**	**97.35**

The best values are shown in bold.

TABLE IV. CLASSIFICATION PERFORMANCES OF BLAST AND CLUSTALX USING TREE-KNN AND TREENN ON THE FOUR DATASETS

	Blast				**ClustalX**			
	Tree-kNN		**TreeNN**		**Tree-kNN**		**TreeNN**	
	Error rate	**Accuracy**	**Error rate**	**Accuracy**	**Error rate**	**Accuracy**	**Error rate**	**Accuracy**
3PGK	**0.00**	**100.00**	**1.75**	**98.25**	**0.00**	**100.00**	**1.75**	**98.25**
COG	**0.12**	**99.88**	**0.19**	**99.81**	0.25	99.75	4.16	95.84
CATH95	**9.66**	**90.34**	**5.55**	**94.45**	18.96	81.04	27.37	72.63
SCOP95	**3.32**	**96.68**	**2.65**	**97.35**	3.81	96.19	25.21	74.79

The best values are shown in bold.

V. DISCUSSION

The standard methods based on pair-wise comparison for protein sequence classification such as BLAST, transfer the class label of a database hit to a query sequence based on predicted similarities, have been shown to prone to systematic errors. The top hit in a sequence database may have a different function to the query due to function evolution stemming mainly from gene duplication. These errors have been propagated in databases by the application of homology-based annotation transfer [18] and [19].

Phylogenomic approach has been shown to enable the highest accuracy in prediction of protein molecular function as shown by Sjölander [20] and Brown and Sjölander [21] but the computational complexity has limited its use. Phylogenomic methods are used for distant similarities that cannot be treated by simple comparison tools like BLAST. Trees are often used in phylogenomics to find the taxonomic relationships between proteins. In this work, we have employed trees as a simple and computationally inexpensive formalism for protein classification.

We have proposed a tree-based algorithm, which searches for the most closely related sequence to the query protein based on the tree structure. The proposed algorithm is tested on four protein classification benchmark datasets. For comparison we also include the results obtained by an existing tree-based method in the literature. We have showed that our algorithm outperforms in terms of error rate and accuracy using four datasets.

It is well known that the multiple alignment method used as a first step for tree construction has a great impact on tree accuracy and, consequently on classification results. In a previous study given by Essoussi et al. [22] multiple alignment methods were evaluated based on the quality of the trees produced from the alignment results of each method. The study has demonstrated a statistically significant difference on alignment accuracies although, the statistical differences observed in the trees produced are not significant on the datasets used.

However, this study has evaluated the effect of pair-wise and multiple alignment approaches on trees accuracies and consequently, on classification performance. The multiple alignment approach is widely used in phylogenomic analyses to compare homologous sequences as a whole. However, we have demonstrated that comparing protein sequences by pairs and not as a whole provides a greatest classification performance.

As a future work, a deep evaluation of pair-wise and multiple alignment approaches on different phylogenomic methods should be conducted. This evaluation may provide interesting insights to phylogenomic researchers on use of multiple alignment approach. Another challenge would be to evaluate distance-based and character-based methods for tree construction on classification performance of phylogenomic methods. A work has been established by Lazareva-Ulitsky et al. [23] in this context. However, the authors have not compared phylogenomic methods, but they have proposed a measure to evaluate classification accuracy of trees estimated

from distance-based methods and hierarchical clustering using different protein similarity measures.

VI. CONCLUSION

We presented Tree-*k*NN a tree-based algorithm for protein sequence classification that exceeds the performance of TreeNN. We demonstrate the competence of our algorithm on four benchmark datasets and two classification measures. Moreover, in sequence alignment step of phylogenomic classification, we show that the two tree-based algorithms using pair-wise alignment approach provide more powerful results than using multiple alignment approach.

REFERENCES

[1] R. Busa-Fekete, A. Kocsor, and S. Pongor "Tree-Based Algorithms for Protein Classification," *Studies in Computational Intelligence*, vol. 94, pp. 165-182, 2008.

[2] J.A. Eisen "Phylogenomics: improving functional prediction for uncharacterized genes by evolutionary analysis," *Genome Research*, vol. 8, no. 3, pp. 163-167, 1998.

[3] S.E. Brenner "Errors in genome annotation," *Trends in Genetics*, vol. 15, pp. 132–133, 1999.

[4] M.Y. Galperin, and E.V. Koonin "Sources of systematic error in functional annotation of genomes: Domain rearrangement, non-orthologous gene displacement and operon disruption," *In Silico Biology*, vol. 1, pp. 55–67, 1998.

[5] L.B. Koski, and G.B. Golding "The closest BLAST hit is often not the nearest neighbour," *Journal of Molecular Evolution*, vol. 52, pp. 540–542, 2001.

[6] R.F. Doolittle "The multiplicity of domains in proteins," *Annu. Rev. Biochem.*, vol. 64, pp. 287–314, 1995.

[7] R.F. Doolittle, and P. Bork "Evolutionarily mobile modules in proteins," *Sci. Am.*, vol. 269, pp. 50–56, 1993.

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

[9] J. Felsenstein "PHYLIP: Phylogeny Inference Package," Version 3.6. University of Washington, Seattle, WA, 2002.

[10] P. Sonego, M. Pacurar, S. Dhir, A. Kertesz-Farkas, A. Kocsor, Z. Gaspari, J.A.M. Leunissen, and S. Pongor "A Protein Classification Benchmark collection for machine learning," *Nucleic Acids Research*, vol. 35, D232-D236, 2007.

[11] R.L. Tatusov, N.D. Fedorova, J.D. Jackson, A.R. Jacobs, B. Kiryutin, E.V. Koonin, D.M. Krylov, R. Mazumder, S.L. Mekhedov, A.N. Nikolskaya, B.S. Rao, S. Smirnov, A.V Sverdlov, S. Vasudevan, Y.I. Wolf, J.J. Yin, and D.A. Natale "The COG database: an updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, no. 41, 2003.

[12] C.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, no. 8, pp. 1093–1108, 1997.

[13] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, pp.536–540, 1995.

[14] J.D. Thompson, T.J. Gibson, F. Plewniak, F. Jeanmougin, and D.G. Higgins "The CLUSTALX windows interface: Flexible strategies for multiple sequence alignment aided by quality analysis tool," *Nucleic. Acids. Research*, vol. 25, pp. 4876-4882, 1997.

[15] J.D. Thompson, T.J. Gibson, and D.G. Higgins "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673-4680, 1994

[16] N. Saitou, and M. Nei "The Neighbor-joining method: a new method for reconstructing Phylogenetic trees," *Molecular Biology and Evolution*, vol. 4, pp. 406-425, 1987.

[17] J. Felsenstein "Confidence limits on phylogenies: an approach using the bootstrap,"*Evolution*, vol. 39, pp. 783–791, 1985.

[18] W.R. Gilks,B. Audit, D. De Angelis, S. Tsoka, and C.A.Ouzounis "Modeling the percolation of annotation errors in a database of protein sequences," *Bioinformatics*, vol. 18, pp. 1641–1649, 2002.

[19] W.R. Gilks, B. Audit, D. De Angelis, S. Tsoka, and C.A.Ouzounis "Percolation of annotation errors through hierarchically structured protein sequence databases," *Math Biosci*, vol. 193, pp. 223-234, 2005.

[20] K. Sjölander K. "Phylogenomic inference of protein molecular function: advances and challenges," *Bioinformatics*, vol. 20, pp. 170-179, 2004.

[21] D. Brown, K. Sjölander, K. "Functional classification using phylogenomic inference," *PLoS Comput Biol*, 2, e77, 2006.

[22] N. Essoussi, K. Boujenfa, and M. Limam "A comparison of MSA tools," *Bioinformation*, vol. 2, no. 9, pp. 452-455, 2008.

[23] B. Lazareva-Ulitsky, K. Diemer, and P.D. Thomas "On the quality of tree-based protein classification,"*Bioinformatics*, vol. 21, no. 9, pp. 1876-1890, 2005.

AUTHORS PROFILE

Khaddouja Boujenfa is a PhD student at the High Institute of Management. She is pursuing her thesis work at LARODEC under the supervision of Professor Mohamed Limam. Her research focuses on classification of protein structures..

Nadia Essoussi received her Ph.D in Computer Science from Faculty of sciences of Tunis. She is associate professor in computer science at ISG, University of Tunis. Her research interests include Data Integration and Data Mining of complex data such as biological data. She is the co-author of several papers published in Bio data mining and Bioinformation. She is member of the Larodec Laboratory and the Tunisian Association of Statistics and its Applications (TASA).

Mohamed Limam received a PhD in Statistics from Oregon State University, he teaches in ISG at the University of Tunis, and his research interests in applied statistics, Data Mining and Quality Control. He is the author of many research studies published in JASA, Machine Learning, Bioinformation, Bio data mining, Communications in Statistics, Quantitative Finance, International Journal of Production Research, Quality and Reliability Engineering International. He is a co-founder of the Tunisian Management Science Society, and founder of the TASA.