Sequence Motif Identification and Protein Family Classification Using Probabilistic Trees[⋆]

Florencia Leonardi and Antonio Galves

Instituto de Matemática e Estatística, Universidade de São Paulo

Abstract. Efficient family classification of newly discovered protein sequences is a central problem in bioinformatics. We present a new algorithm, using *Probabilistic Suffix Trees*, which identifies equivalences between the amino acids in different positions of a motif for each family. We also show that better classification can be achieved identifying representative fingerprints in the amino acid chains.

1 Introduction

A central problem in genomics is to determine the function of a new discovered protein using the information contained in its amino acid sequence [1]. Nowadays, the most popular methods to generate a hypothesis about the function of a protein are BLAST and Hidden Markov Models (HMM).

Probabilistic Suffix Trees (PST) were first introduced in [2] as a universal model for data compression. A major advantage of PST is its capacity of extracting structural information from the sequences under analysis. Recently, an implementation of PST has been successfully used in protein classification [3], even though its performance decreases with less conserved families. Better results have been obtained using mixtures of PST models for sparse sequences [4, 5]. A major drawback of these algorithms is their high complexity, which makes problematic their application in very large databases.

We present a new algorithm to estimate *Sparse Probabilistic Suffix Trees* (SPST). We also show that the identification of sub-sequences of maximal mean probability (*fingerprints*) increases the classification rates of the SPST algorithm. This is the basis of our F-SPST algorithm.

2 Variable Length Markov Models

It was suggested in the literature to use PST models to fit protein families. A PST is a Variable Length Markov Model (VLMC), that is, a stochastic chain $(X_0, X_1, \ldots)$ taking values on a finite alphabet $\mathcal{A}$ and characterized by two elements. The first element is the set of all contexts. A context $X_{n-\ell}, \ldots, X_{n-1}$

[⋆] This work is partially supported by CAPES and is part of PRONEX/FAPESP's Project *Stochastic behavior, critical phenomena and rhythmic pattern identification in natural languages* (grant number 03/09930-9).

J.C. Setubal and S. Verjovski-Almeida (Eds.): BSB 2005, LNBI 3594, pp. 190–193, 2005.

is the finite portion of the past $X_0, \ldots, X_{n-1}$, for each time, which is relevant to predict the next symbol X_n. The second element is a family of probability transitions associated to each context. Given a context, its associated probability transition gives the distribution of occurrence of the next symbol immediately after the context.

In a PST the set of contexts has the *suffix property*: looking from the present to the past no context is a suffix of another context. This makes it possible to define without ambiguity the probability distribution of the next symbol. The suffix property makes it possible to represent the set of contexts as a tree. In this tree, each context $c = (c_{-k}, \ldots, c_{-1})$ is represented by a complete branch, in which the first node on top is c_{-1} and so on until the last element c_{-k} which is represented by the terminal node of the branch.

In a PST model for a protein family, the alphabet $\mathcal{A}$ represents the set of twenty amino acids and the stochastic chains $(X_0, X_1, \ldots)$ are the sequences of amino acids belonging to the family.

A *Sparse Probabilistic Suffix Tree* (SPST) is a PST in which some contexts are grouped together in an equivalence class. More precisely, the contexts of a SPST model are sequences of the form $A_{n-\ell}, \ldots, A_{n-1}$, with $A_i \subset \mathcal{A}$ for each i. This feature makes SPST models more suitable for sparse sequences like amino acids chains.

3 The SPST and the F-SPST Algorithms

The SPST algorithm works as follows. It starts with a tree consisting of a single root node. At each step, for every terminal node t with depth less than L and for every symbol x, the leaf x is added to t, if the sequence xt appears in the training sequences at least $N_{\min}$ times. For every pair of new leaves of a node, we test their *equivalence* using a log-likelihood ratio test and choose the pair that realizes the minimum between all the tests. If this minimum belongs to the acceptance region, the leaves are merged together in a single leaf. The procedure is iterated with the new set of leaves. It stops when no more leaves can be merged. The acceptance region is defined by $\{c < r_{max}\}$, where c is the value of the test. Clearly, taking the minimum between the tests ensures the independence of the order in which the tests are performed.

To conclude the construction of the SPST we assign to each leaf a transition probability estimated by the usual maximum likelihood procedure. In order to avoid non zero probabilities, the distributions associated to each leaf (context) are smoothed by a constant γ_{min}.

After the construction of the model, we want to decide if a given sequence of amino acids belongs to the family or not. To do this, we calculate the log probability of the sequence in the family model and divide this value by the length of the sequence. If this value is greater than a predefined threshold, the protein is identified as a member of the family.

The *Fingerprint-SPST* algorithm estimates the context tree and the transition probabilities in the same way as the SPST algorithm. However, to classify a new sequence of amino acids, F-SPST starts by identifying fingerprints defined as follows. Given a new sequence of amino acids, we look for the sub-sequence of length M with maximal probability, where M is a parameter which depends on the size of the domains in each family. If this maximum is bigger than a pre-defined threshold, the protein is identified as a member of the family.

4 Statistical Results

In order to test our algorithms and to compare them with PST published results [3] we use protein families of the Pfam database [6] release 1.0. This database contains 175 families derived from the SWISSPROT 33 database [7]. We trained both SPST and F-SPST with 4/5 of the sequences in each family, and then we applied the resulting models to classify all the sequences in the SWISSPROT 33 database. To establish the family membership threshold, we used the **equivalence number criterion** [8]. This method sets the threshold at the point where the number of false positives equals the number of false negatives. The quality of the model is measured by the number of true positives detected relative to the total number of proteins in the family.

Table 1 summarizes the classification rates obtained with our SPST and F-SPST algorithms together with the published results obtained with the PST algorithm [3]. We emphasize that these are preliminary results as no attempt was made to optimize the choice of the parameters. It is clear that SPST and

Table 1. Performance comparison between PST, SPST and F-SPST. The parameters in the SPST and F-PST algorithms where: $L = 20$, $N_{\min} = 2$, $\gamma_{\min} = 0.001$ and $r_{\max} = 3.8$. The length of the fingerprint in the F-SPST algorithm was $M = 80$ for all families

Family	Size	PST	SPST	F-SPST
7tm_1	515	93.0%	96.3%	97.7%
7tm_2	36	94.4%	97.2%	100.0%
7tm_3	12	83.3%	100.0%	100.0%
AAA	66	87.9%	90.9%	93.9%
ABC_tran	269	83.6%	85.9%	89.3%
actin	142	97.2%	97.2%	99.3%
adh_short	180	88.9%	89.4%	92.8%
adh_zinc	129	95.3%	91.5%	95.3%
aldedh	69	87.0%	89.9%	92.8%
alpha-amylase	114	87.7%	91.2%	94.7%
aminotran	63	88.9%	88.9%	90.5%
ank	83	88.0%	86.8%	86.6%
arf	43	90.7%	93.0%	93.0%
asp	72	83.3%	90.3%	91.7%
ATP-synt_A	79	92.4%	94.9%	97.5%

F-SPST improves PST classification rates in all cases except for the *Ankyrin repeat* family. It is interesting to note that this family consists of very short domains (with mean length equal to 28.12), and this could explain the reduction in the classification rate.

Another very interesting feature of SPST appears when we compare the equivalence classes in the estimated trees with the classes obtained by grouping the amino acids by their physical and chemical properties. For instance, the estimated tree for the AAA family identifies as equivalence class the set of amino acids $\{I, V, L\}$ which corresponds exactly to the group of aliphatic amino acids. For more details see `http://www.ime.usp.br/~leonardi/spst/`.

5 Conclusion

The preliminary results presented in this paper strongly suggest that these new algorithms can improve in a significant way the classification rates obtained with the PST algorithm. We are presently applying our algorithms to more families in the Pfam database to confirm this initial encouraging results.

Nevertheless, even at this preliminary stage, it is alredy clear that a Sparse Probabilistic Tree fits protein families well. This is probably due to the fact that the sparse model mimics well the sparse nature of relevant domains in the amino acids chains. It is also worth observing that the complexity of the SPST and F-SPST algorithms is smaller than the complexity of previously presented algorithms for sparse sequences [4, 5].

References

1. Karp, R.M.: Mathematical challenges from genomics and molecular biology. Notices Amer. Math. Soc. **49** (2002) 544–553
2. Rissanen, J.: A universal data compression system. IEEE Trans. Inform. Theory **29** (1983) 656–664
3. Bejerano, G., Yona, G.: Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. Bioinformatics **17** (2001) 23–43
4. Eskin, E., Grundy, W.N., Singer, Y.: Protein family classification using sparse markov transducers. In: Proc. Int'l Conf. Intell. Syst. Mol. Biol. Volume 8. (2000) 134–145
5. Bourguignon, P.Y., Robelin, D.: Modèles de Markov parcimonieux: sélection de modèle et estimation. manuscript (2004)
6. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L., Studholme, D.J., Yeats, C., Eddy, S.R.: The Pfam protein families database. Nucl. Acids Res. **32** (2004) D138–141
7. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucl. Acids Res. **31** (2003) 365–370
8. Pearson, W.R.: Comparison of methods for searching protein sequence databases. Protein Sci **4** (1995) 1145–1160