ACTAS DE LA ACADEMIA NACIONAL DE CIENCIAS. (CORDOBA, ARGENTINA)
TOMO XIV, DICIEMBRE DE 2008

PATTERN RECOGNITION ON RANDOM TREES ASSOCIATED TO FUNCTIONALITY FAMILIES OF PROTEINS

Ana G. Flesia[1,*] Ricardo Fraiman[2] & Florencia Leonardi[3]

1: FaMAF-UNC y CIEM-CONICET. Ing. Medina Allende s/n, Ciudad Universitaria. CP 5000, Córdoba, Argentina.
2: Departamento de Matemática y Ciencias. Universidad de San Andrés, Argentina y Universidad de la República, Uruguay.
3: Instituto de Matemática e Estadística. Universidade de São Paulo, Brazil.
*E-mail: flesia@mate.uncor.edu

PATTERN RECOGNITION ON RANDOM TREES
Keywords: supervised learning, unsupervised learning, space of trees, protein classification.

RECONOCIMIENTO DE PATRONES EN ARBOLES ALEATORIOS
Palabras clave: clasificación, agrupamiento, espacio de árboles, clasificación de proteinas.

RECONOCIMIENTO DE PATRONES EN ÁRBOLES ALEATORIOS ASOCIADOS A FAMILIAS FUNCIONALES DE PROTEÍNAS

SYNOPSIS

In this paper, we address the problem of identifying protein functionality using the information contained in its aminoacid sequence. We propose a method to define sequence similarity relationships that can be used as input for classification and clustering via well known metric based statistical methods.

To obtain our measure of sequence similarity, we first fit a Variable Length Markov model (VLMC) to each sequence of our database, generating estimated context trees, and then we compute the BFFS distance (Balding *et al.* 2004) in tree space between each pair of trees. The BFFS distance takes into account the structure of each tree, that is directly related to the most relevant motifs of the sequence, and indirectly, to the probability of occurrence of each motif.

In our examples, we specifically address two problems of supervised and unsupervised learning in structural genomics via simple metric based techniques on the space of trees

- Unsupervised detection of functionality families via K-means clustering in the space of trees,
- Classification of new proteins into known families via k-nearest neighbour trees.

We found evidence that the similarity measure induced by our approach concentrates information for discrimination. Classification has the same high performance than others VLMC approaches. Clustering is a harder task, though, but our approach for clustering is alignment free and automatic.

.

SINOPSIS

En este trabajo estudiamos el problema de identificar la función de una proteína usando información contenida en su estructura primaria de aminoácidos. Proponemos un método que defina relaciones de similaridad entre sucesiones de aminoácidos, que puede ser utilizada para clasificación y determinación de grupos, usando métodos estadísticos clásicos.

Para obtener nuestra medida de similaridad primero ajustamos una cadena de Markov de alcance variable (VLMC) a cada sucesión en nuestra base de datos generando árboles de contexto, y luego calculamos la distancia BFFS (Balding *et al.* 2004) entre cada par de árboles. La distancia BFFS considera la estructura de cada árbol y en forma indirecta las probabilidades de transición asociadas a cada contexto.

En nuestros ejemplos nos referimos específicamente a dos problemas de aprendizaje supervisado y no supervisado de genómica estructural vía métodos clásicos basados en distancia en el espacio de árboles aleatorios.

- Detección no supervisada de familias funcionales mediante el método de K-medias
- Clasificación de nuevas proteínas en nuevas familias vía k-vecinos más cercanos.

Encontramos evidencia de que la medida de similaridad inducida por nuestra propuesta concentra información útil para discriminación. Nuestra propuesta de clasificación mantiene la tasa de error de los otros métodos basado en VLMC. Agrupamiento es un problema más difícil, y hemos obtenido resultados dispares, pero nuestra propuesta no precisa alinear las sucesiones primero y es totalmente automática.

INTRODUCTION

A central problem in functional genomics is to determine the function of a protein using only the information contained in its aminoacid chain, Karp (2002). It is well known that a protein functionality family is formed by proteins that perform the same function on different organisms and by proteins that come from the same organism that have been derived by genetic duplication or rearrangements, Dayhoff (1976), Hegyi (1999). Well characterized proteins within a family may help enhance the process of classification of family members whose functions are not well known or not well understood, Eisenberg (2000). Also, the features characterizing each functionality family may give information about common evolutionary history, Sasson *et al.* (2003).

Most used methods for proposing hypothesis over protein functionality are based on sequence alignment, Smith (1981). Exact sequence alignment has a quadratic computational complexity, which make them unfeasible for large databases. Heuristic methods like BLAST, (Altschul *et al.* 1997) or FASTA, (Pearson *et al.* 2000) are between the most common choices for comparing sequences in large data sets.

Recently, this problem has been addressed also with non alignment methods, that look for family models with parameters or characteristics that determine its functionality. An example of such body of work is the fitting with different markovian models, like Hidden Markov Models (Rabiner *et al.* 1986), Hidden Markov Transducers (Eskin *et al.* 2000), and Variable Length Markov Chains (VLMC) (Bejerano 2001, Apostolico *et al.,* 2000).

Hidden Markov Models (HMM) are very powerful tools for this task, but have the disadvantage of having too many parameters to fit, and even though, in practice, they do not guarantee an optimal choice of model.

Recently, Bejerano *et al.* (2001) proposed to apply Variable Length Markov Chains to the problem of classification of proteins into families. Some advantages of this model are the following: it does not depend of alignment, it has not as many parameters as HMM, and there are algorithms that can fit the model in linear time, (Apostolico *et al.,* 2000).

We are going to address two specific problems here:

- Detection of functionality families via sequence clustering,
- Classification of new proteins into known families.

These problems are directly related to the problem of detecting protein functionality, so they can be addressed by the same methods, but with great variations in performance.

From the mathematical point of view, clustering is an ill posed problem. The definition of functionality family is quite ambiguous, so it is very difficult to quantify it mathematically to obtain a unique objective function to optimize. As a result, computational clustering approaches differ in the representations of the proteins to be clustered, the definition of the optimization goals and also in the resulting partitions of the known protein space. Stability and heterogeneity of the resulting clusters are known problems that are shared by most methods, but still they help to build a big picture of the on going experimental structure which represents super-families. The goal of fully automated clustering methods becomes to give partial answers with respect to global organization of all protein sequences.

Sequence classification into families is a simpler task than unsupervised learning, but still has delicate problems as the one introduced by multi-domain proteins, and the accuracy of the labelling of the training set. A related problem is to classify sequences belonging to a family into subfamilies that are in most cases, defined by their evolutionary history. Most tailored methods rely on a multiple alignment of the family sequences as well as the phylogeny tree inferred from it. Indeed, when the resulting evolutionary tree can be reconstructed accurately, functional subtypes can often be identified with subtrees within it.

There are many clustering techniques that rely in pairwise similarity measures of protein sequences. ProtoMap, (Yona *et al.* 2000) ProtoNet, (Sasson *et al.* 2003), BioSpace (Yona and Levitt 2000), use a combination of the three most common measures of pairwise similarity, (Smith-Waterman, Fasta and Blast) followed by the construction of a weighted graph that has the resulting clusters as the most strong connected components. The evolution of the graph differs in each algorithm. Tribe-MCL (Enright *et al.,*2003) uses also BLAST to build up a dissimilarity matrix, converting it into a probability matrix which is used to simulate a flow that leads to the final graph. Each algorithm has evolved into complicated learning machines, to avoid the multi domain protein problem, and to generate a hierarchical view of protein space. In this paper we will work with a very simple clustering machine that also relies in pairwise similarity, the K-means algorithm, which is well suited to detect generative clusters when the underlying distributions are concentrated.

In the case of the non alignment methods, most of them rely in modelling protein functionality, so the more accurate the fit of the model, the better the results in clustering and classification. In this paper, we explore a hybrid approach for protein classification and clustering. We fit a Variable Length Markov Model to each protein sequence, and we use the architecture of their associated context trees to perform classification and clustering, considering a metric on the space of trees. The BFFS distance (Balding *et al.,* 2007) takes into account the structure of each tree, that is directly related to the most relevant motifs of the sequence, and indirectly, to the transition probabilities associated to these motifs. This approach is motivated by the idea that proteins that have the same functionality could be modelled with the same VLMC. In consequence, their estimated context trees are observations of the same random element, and should be close together in tree space. We are combining a model based technique with the classical similarity based statistical learning.

MATERIAL AND METHODS

Data handling

A FASTA file containing all sequences that are to be clustered o classified into families is assembled. The labels are only visible for evaluation and training purposes. This file is transformed via PST algorithm (Bejerano *et al.,* 2001) in trees of fixed depth 4.

Algorithm

The file containing all trees is compared against itself using the BFFS distance for trees. The all against all sequence similarities generated by this analysis are stored in a square matrix. The labels are reserved for evaluation and training purposes.

- **Unsupervised Clustering**
  We have chosen a very simple clustering technique that rely on distances, and that optimizes the within sum of distances in each cluster. The distance between two sequences is in fact the distance between two trees estimated via the PST algorithm, so we can obtain the partition C with

an alternating optimization procedure that first compute the cluster mean centroid trees of a given partition and then reassign the observations to the closest centroid tree, until the objective function is no longer decreased. This can be done because there is a notion of average tree or mean centroid tree that shares properties of the average in Euclidean space. We have adapted a Matlab code for K-means in order to handle trees in BFS format, and computing the BFFS distance and the mean centroid tree as needed.

- **Supervised Learning**
  We have approached classification also with one of the simple schemes that rely on distance. Given a new sequence, we classify it in the family that has more members between the k closest members of the database, with a standard code in MATLAB.

Availability

The original PST algorithm for Variable Length Markov Chain modelling can be adapted from (Bejerano, 2003). The additional modules for computing distances and mean trees, necessaries for protein sequence clustering and classification can be obtained from the authors upon request.

VARIABLE LENGTH MARKOV CHAIN MODELLING OF PROTEIN FUNCTIONALITY

The starting point of our approach to supervised and unsupervised learning is to learn the structure of the Variable Length Markov Chain that models each family. We claim in this paper that the estimator of the context tree that characterize the VLMC of each family is a random tree, a random object that produces trees following a distribution that also characterize the family. Classification and clustering is then carried out in a metric space of trees, and as it is well known, the success of it depends strongly in the concentration of the distribution of the family in tree space.

We compute estimates of the VLMC context tree of each family using the Probabilistic Suffix Trees algorithm.

VLMC and the BFFS space of trees

A Variable Length Markov Chain is a stochastic process introduced by Rissanen (1983) in information theory; see also Bühlmann and Wyner (1999). In this model the probability of occurrence of each symbol at a given time depends on a finite number of precedent symbols. The number of relevant precedent symbols may be variable and depends on each specific sub-sequence. More precisely, a VLMC is a stochastic process (X_n), with values on a finite alphabet A such that

$$P\left[X_n = . \mid X_{-\infty}^{n-1} = x_{-\infty}^{n-1}\right] = P\left[X_n = . \mid X_{n-k}^{n-1} = x_{n-k}^{n-1}\right]$$

where x_s^r represents the sequence $x_s, x_{s+1}, \ldots, x_r$ and k is a stopping time that depends on the sequence $x_{n-k}, \ldots, x_{n-1}$. As the process is homogeneous the relevant past sequences $x_{n-k}, \ldots, x_{n-1}$ do not depend on n and are denoted by $(x_{-k}, \ldots, x_{-1})$. Each relevant past $(x_{-k}, \ldots, x_{-1})$ is called a *context*. The set of contexts τ can be represented as a rooted tree t, where each complete path from the leaves to the root in t represents a context. Calling p the transition probabilities associated to each context in τ given by the former equation, the pair (τ,p), called *probabilistic context tree*, has all information relevant to the model, see Rissanen (1983) and Bühlmann *et al.* (1999).

As an example, take a binary alphabet $A = \{1,2\}$ and transition probabilities

$$P\left[X_n = 1 \mid X_{-\infty}^{n-1} = x_{-\infty}^{n-1}\right] = \begin{cases} 0.7, & if\ x_{n-1}=1, x_{n-2}=1 \\ 0.4, & if\ x_{n-1}=1, x_{n-2}=2 \\ 0.2, & if\ x_{n-1}=2. \end{cases}$$

$$P\left[X_n = 2 \mid X_{-\infty}^{n-1} = x_{-\infty}^{n-1}\right] = 1 - P\left[X_n = 1 \mid X_{-\infty}^{n-1} = x_{-\infty}^{n-1}\right]$$

so that, if $x_{n-1}=2$, then the stopping time is $k=1$ and $X_n=1$ with probability 0.2; otherwise the stopping time is $k=2$, and $X_n=1$ with probability 0.7 if both $x_{n-1}=x_{n-2}=1$, or $X_n=1$ with probability 0.4 if $x_{n-1}=1$ and $x_{n-2}=2$. The set of contexts is τ_x={11,21,2}, and it can be matched with a rooted tree τ_x with all its leaves in τ.

Another example over the same alphabet is given by the transition probabilities

$$P\left[Y_n = 1 \mid Y_{-\infty}^{n-1} = y_{-\infty}^{n-1}\right] = \begin{cases} 0.6,\ if\ y_{n-1}=1. \\ 0.4,\ if\ y_{n-1}=2, y_{n-2}=2 \\ 0.2,\ if\ y_{n-1}=2, y_{n-2}=1 \end{cases}$$

$$P\left[Y_n = 2 \mid Y_{-\infty}^{n-1} = y_{-\infty}^{n-1}\right] = 1 - P\left[Y_n = 1 \mid Y_{-\infty}^{n-1} = y_{-\infty}^{n-1}\right]$$

The set of contexts is τ_y={1,12,22}. The matched probabilistic context trees are represented in Figure 1.

We are going now to embed the set of all possible context trees that can be constructed for a given a fixed alphabet $\mathcal{A}$, into the compact metric space of rooted trees, $\mathcal{T}$, defined by Balding *et al.* (2004). The relationship is very simple, since a rooted tree can be thought as a subset of the nodes satisfying the condition "son present implies father present". In this kind of space, the natural sigma algebra $\mathcal{B}$ is the minimal one containing cylinders, that is, the sets of trees defined by the presence/absence of a finite number of nodes. The natural topology is the one generated by the cylinders as open sets. We can associate to this topology a family of distances that take values depending on matched presence/absence of nodes in the trees, including also the internal nodes.

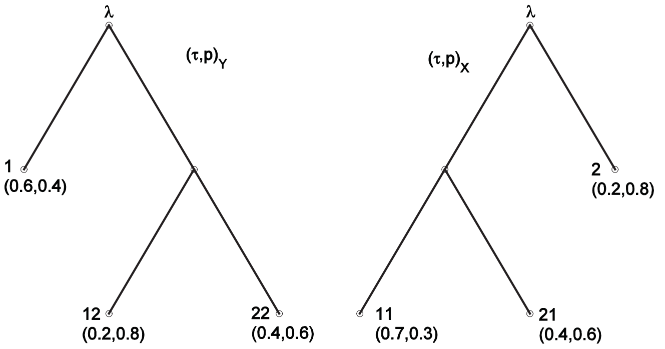


Figure 1: Two probabilistic context trees.

So let define $\mathcal{V}$ as the set of all possible sequences $(x_{-k},\ldots,x_{-1})$ over the alphabet $\mathcal{A}$, with all possible stopping time k, and define a tree t as a function t: $\mathcal{V}\rightarrow$\{0,1\} such that t only give value one to the set of contexts of the tree and the set of internal nodes of these contexts. With this definition, it is easy to define our distance as

$$d(t,y) = \sum_{v\in \mathrm{V}} \varphi(\mathrm{v})\,(t(v) - y(v))^2$$

where φ is a strictly positive function such

$$\sum_{v\in \mathrm{V}} \varphi(\mathrm{v}) < \infty$$

Let compute the distance between the two trees given in the preceding example,

$$\begin{aligned} d(t,y) = & \sum_{v\in V} \varphi(v)\,(t(v) - y(v))^2 \\ & \varphi(\lambda)\big(t(\lambda) - y(\lambda)\big) + \\ & \varphi(1)\big(t(1) - y(1)\big) + \varphi(2)\big(t(2) - y(2)\big) + \\ & \varphi(11)\big(t(11) - y(11)\big) + \varphi(12)\big(t(12) - y(12)\big) + \\ & \varphi(21)\big(t(21) - y(21)\big) + \varphi(22)\big(t(22) - y(22)\big) \\ = & \ 0 + 0 + 0 + \varphi(11) + \varphi(12) + \varphi(21) + \varphi(22) \end{aligned}$$

With this distance ($\mathcal{T}$, d) becomes a compact metric space, see details in Balding *et al.* (2004).

Following Balding *et al.* (2004), in our computational examples we use the function

$$\varphi(v) = z^{gen(v)}$$

with $0<z<1$, where *gen (v)* stands for the generation of the node, the number of symbols to reach the root. We found the choice of the parameter z not relevant in order to improve the performance of the classification and clustering methods, so we will fix its value in $z=0.1$

Random trees

A *random tree* with distribution η is a $\mathcal{F}$-measurable function T:Ω→ $\mathcal{T}$such that

$$\varphi(v) = z^{gen(v)}$$

for any Borel set A in $\mathcal{B}$, where (Ω,$\mathcal{F}$,P) is a probability space and η a probability on ($\mathcal{T}$, $\mathcal{B}$).

Given a sample of independent random trees $T_1,\ldots,T_n$ with identical distribution η on our compact metric space ($\mathcal{T}$, $\mathcal{B}$,), a measure of central tendency is a *sample centroid* defined as a tree (or set of trees) $\overline{T}$ in $\mathcal{T}$satisfying

$$\overline{T} = \arg\min_{t\in T} \frac{1}{n}\sum_{i=1}^{n} d(T_i, t)$$

This formula may show the problem as more difficult than what it really is, since it is calling for a search over the whole set of trees, that grows exponentially in the number of nodes. But it is easy to prove that the sample centroid (or mean tree) of a set of trees can be built by majority vote over the nodes. That means, at least one of the sample centroids (it does not need to be unique) can be defined as the tree whose nodes are present only if they are present in at least half of the sample, see Flesia *et al.* (2007) for details.

SUPERVISED AND UNSUPERVISED LEARNING ON THE SPACE OF TREES

In cluster analysis, the goal is to find an optimal partition for which observations or objects within each cluster are similar, but the clusters are dissimilar to each other. It differs fundamentally from classification analysis, where the observations are allocated to a known number of predefined groups or populations. Many techniques are based on a certain measure of similarity between pairs of observations.

K-means clustering

We are concerned with a particular cluster technique, called K-means clustering procedure, which generates the class labels trough the minimization of the "within cluster" point scatter, a dissimilarity based loss function defined by

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k}\left\|x_i - x_{i'}\right\|^2$$

This rule characterizes the extent to which observations assigned to the same cluster tend to be close to each other. It was initially intended for real valued quantitative variables, and the squared Euclidean distance was chosen as a measure of dissimilarity. In our case, we have objects belonging to a space of trees, so we choose the distance d as the dissimilarity measure, and the within-point scatter may be redefined as

$$W(C) = \sum_{k=1}^{K}\sum_{C(i)=k} d\left(t_i, \overline{t}_k\right)$$

where $\overline{t}_k$ is the sample centroid associated to the kth cluster. As in the case of the Euclidean space, an iterative descent algorithm for solving

$$C^* = \arg\min_{C}\sum_{k=1}^{K}\sum_{C(i)=k} d\left(t_i, \overline{t}_k\right)$$

may be obtained by noting that for any set of observed trees S

$$\overline{t}_S = \arg\min_{t}\sum_{t_i\in S} d\left(t_i, t\right)$$

by definition. Hence we can obtain C* with an alternating optimization procedure that first compute the cluster mean centroid trees and then reassign the observations to the closest centroid tree, until the objective function is no longer decreased. This is one of the most popular iterative descent algorithms that go by the name of K-means, and the one we use in our examples. It is not difficult to prove, following Pollard (1981) par example, that in the case of locally compact metric spaces the sample centroid trees converge to the population centroid tree when the sample increases. This important result ensures that if there is a partition of the population into K-clusters, and if we have enough data, with the proper initialization to avoid local minima, the K-means algorithm will give an accurate outline of the clusters.

As a final note we state that K-means algorithm could be used for classification purposes, following the next steps

- apply R means clustering to the training data in each of the K classes separately, selecting R prototypes per class (the R centroids),
- assign a class label to each of the set of R prototypes,
- classify a new feature t to the class of the closest prototype.

This is an example of the prototype methods of classification that can be also adapted to work on spaces of trees. The difference between this method and k-nearest neighbours is the fact that the prototypes are not part of the training samples, but the centroids of the partition of each training sample class.

k-nearest neighbours classification

Given a family of proteins $\mathcal{F}$ and a new sequence of amino acids *s,* the goal is to determine if *s* belongs to $\mathcal{F}$ or not. To answer that question, Bejerano (2003) and Leonardi (2007) estimate first a model for the family $\mathcal{F}$, using sequences classified inside the family. To determine the label of a new protein, they search for the family whose model has higher probability of having produced that sequence. The model constructed for the family $\mathcal{F}$ is a Variable Length Markov Chain, obtained estimating the probabilistic tree that matches the chain by means of the PST algorithm as in Bejerano (2003).

In this paper we fit the simplest model for classification, we consider that proteins from the same family are clustered tightly, measuring it with the BFFS distance for context trees, and we score the new protein with the rule of the k-nearest neighbours. We label the protein as belonging to the family that has more neighbours in this k-subjects neighbourhood.

The k-nearest neighbour rule is a very simple, distance based method for pattern recognition or data classification. This method relies on the intuitive concept that data points of the same class should have neighbours in the class (in distance) with high probability. As a result, for a given data point of an unknown class, we can simply compute the distance of this point to the training data, and assign the class determined using majority vote among the k neighbours of this data point. The algorithm is straightforward

- Given a new observation t we find the k training points $S=\{t_{(1)},\ldots,t_{(k)}\}$ closest in distance to t.
- Classification is made using majority vote among the k neighbours in S.
- Ties are broken at random.

The simplicity of the rule is extreme, and its success depend only in the ability of the measure to cluster families, and the ability of the VLMC modelling to successfully detect all the differences between the protein chains of the same family and resume them into its context tree.

Computational Examples

In this section we would like to assess the capability of VLMC methods to capture the essential structure of the family that would help the discrimination problem. Traditional PST classification methods choose a training set of sequences of a given family and estimates the context tree of the family concatenating all the sequences in that training set. Then classification is performed computing the probability that a given sequence would be produced by that context tree. The motivation of such approach is related to biological understanding of the evolution and composition of protein families. We suppose that a group of evolutionary related protein sequences should exhibit many identical short segments which have been either preserved by selection or have not diverged long enough from their common single ancestral sequence. The variable memory model is well equipped to pick up these locally conserved segments, showing them in the architecture of the context tree, (Bejerano, 2003).

Our approach diverges from the classical approach of VLMC methods since we do not use for classification the empirical probabilities associated to each context but the architecture of the context tree that is computed for each protein sequence in the family. Also, we do not collect all the samples to generate an estimation of the model, but we compute an estimate per sample sequence and look how they cluster together in tree space. The context tree built with all the collected sequences will show segments that are consistently repeated in most of the sequences, but context trees built with each sequence will show patterns inside each particular sequence, and the family bond will emerge as a relationship in tree space. The definition of the distance is thus fundamental for our approach.

We have illustrated these ideas with some small examples, following the methodology of Bejerano (2003), since we want to compare our approach with the traditionally PST approach, and indirectly, with other classification methods. Our reference is the Pfam database that is based on Hidden Markov Models trained in a smaller database of manually curated well known proteins.

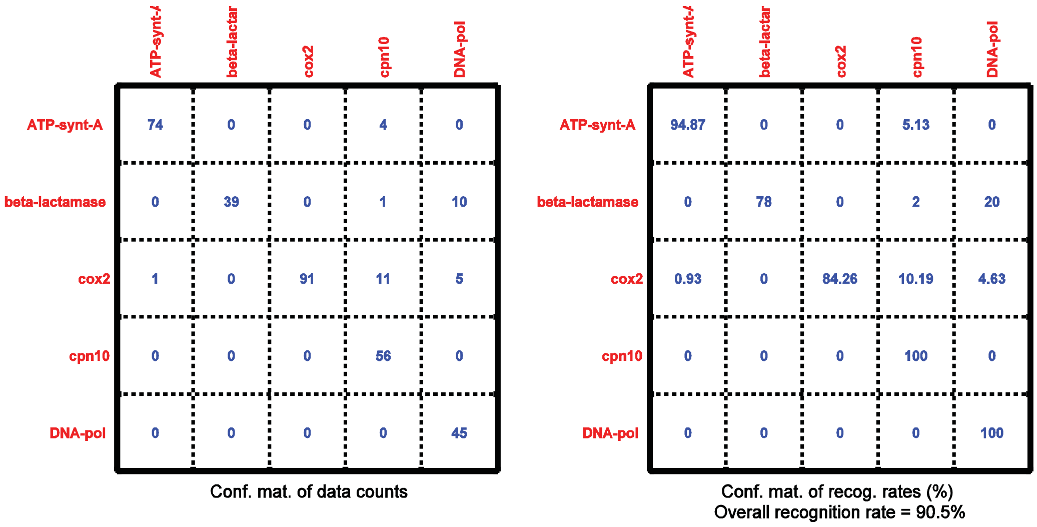
K-means

Clustering is a very hard problem, since there is no help of ground truth to shape the families. Most databases have been constructed clustering sequences with statistical methods, but with the help of a core of manually curated well known proteins. We do not aim to show a method that would accurately cluster the whole Pfam database in seconds, but to discuss the potential that our model could have for checking coherence and relationships among families. We should notice too that K-means is an algorithm that considers similarities in a pairwise fashion, methods that consider global similarities could help cluster assignment.

In order to assess the ability of the K-means procedure to cluster proteins in tree space using the BFFS distance, we have started transforming 5 families of proteins selected from the Pfam database, labelled ‘ATP-synt-A’, ‘beta-lactamase’,’cox2’, ‘cpn10’, ‘DNA-pol’. The overall performance is 90.5%. In Figure 2 we have plot two matrices, one of counts and other with percentage of well classified and misplaced proteins.

We notice that the two families that are correctly clustered, showing a high dgree of coherence, also attract members from other families. as it is shown in Figure 2.

In our second example we added 6 more families, ‘7tm-1’, ’actin’, ’adh-short’, ’adh-zinc’,’ank’, and ‘efhand’, and the overall recognition rate drop to 64%, but the confusion matrix plot in Figure 3 show us that proteins are not scattered around but are misplaced in specific families. This feature could be very interesting at the time to determine the coherence of the family and the relationships between different families. For example, from the ten proteins of the beta lactamase family that have been incorrect-



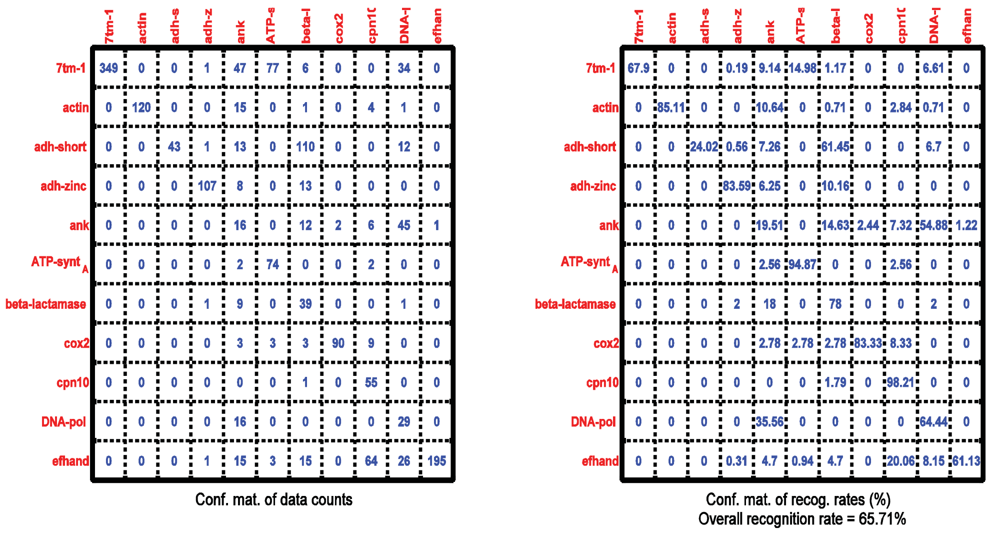
	ATP-synt-A	beta-lactar	cox2	cpn10	DNA-pol
ATP-synt-A	74	0	0	4	0
beta-lactamase	0	39	0	1	10
cox2	1	0	91	11	5
cpn10	0	0	0	56	0
DNA-pol	0	0	0	0	45

Conf. mat. of data counts

	ATP-synt-A	beta-lactar	cox2	cpn10	DNA-pol
ATP-synt-A	94.87	0	0	5.13	0
beta-lactamase	0	78	0	2	20
cox2	0.93	0	84.26	10.19	4.63
cpn10	0	0	0	100	0
DNA-pol	0	0	0	0	100

Conf. mat. of recog. rates (%)
Overall recognition rate = 90.5%

Figure 2: Confusion matrices of data counts and percentage of well classified and misplaced proteins



	7tm-1	actin	adh-s	adh-z	ank	ATP-s	beta-l	cox2	cpn10	DNA-	efhan
7tm-1	349	0	0	1	47	77	6	0	0	34	0
actin	0	120	0	0	15	0	1	0	4	1	0
adh-short	0	0	43	1	13	0	110	0	0	12	0
adh-zinc	0	0	0	107	8	0	13	0	0	0	0
ank	0	0	0	0	16	0	12	2	6	45	1
ATP-synt$_A$	0	0	0	0	2	74	0	0	2	0	0
beta-lactamase	0	0	0	1	9	0	39	0	0	1	0
cox2	0	0	0	0	3	3	3	90	9	0	0
cpn10	0	0	0	0	0	0	1	0	55	0	0
DNA-pol	0	0	0	0	16	0	0	0	0	29	0
efhand	0	0	0	1	15	3	15	0	64	26	195

Conf. mat. of data counts

	7tm-1	actin	adh-s	adh-z	ank	ATP-s	beta-l	cox2	cpn10	DNA-	efhan
7tm-1	67.9	0	0	0.19	9.14	14.98	1.17	0	0	6.61	0
actin	0	85.11	0	0	10.64	0	0.71	0	2.84	0.71	0
adh-short	0	0	24.02	0.56	7.26	0	61.45	0	0	6.7	0
adh-zinc	0	0	0	83.59	6.25	0	10.16	0	0	0	0
ank	0	0	0	0	19.51	0	14.63	2.44	7.32	54.88	1.22
ATP-synt$_A$	0	0	0	0	2.56	94.87	0	0	2.56	0	0
beta-lactamase	0	0	0	2	18	0	78	0	0	2	0
cox2	0	0	0	0	2.78	2.78	2.78	83.33	8.33	0	0
cpn10	0	0	0	0	0	0	1.79	0	98.21	0	0
DNA-pol	0	0	0	0	35.56	0	0	0	0	64.44	0
efhand	0	0	0	0.31	4.7	0.94	4.7	0	20.06	8.15	61.13

Conf. mat. of recog. rates (%)
Overall recognition rate = 65.71%

Figure 3: Confusion matrices of data counts and percentage of well classified and misplaced proteins

ly assigned to the 'Dna-pol' family, 9 of them have been reassigned now to the 'ank' family, but 50% of the ank proteins have been assigned to the 'Dna-pol' family, showing that these three families are close in tree space.

k-nearest neighbours

Now we consider the same 11 families from the previous example, '7tm-1', 'actin', 'adh-short','adh-zinc','ank','ATP-synt-A','beta lactamase', 'cox2', 'cpn10', 'DNA-pol', 'efhand' but we easy the difficulty of the clustering problem allowing the help of a training set. We selected randomly 80% of the whole set of proteins, and set it aside as training set. We could refine later the example, choosing randomly 80% *of each family* as a training set. Then we classify the whole set of 1700 proteins using the training set to generate neighbourhoods of different sizes, and plot in blue the percentage of true positives detected as a function of the number of neighbours considered. Secondly, we compute the percentage of true positives detected but considering only the 20% of proteins that are not included in the training set, and plot it in red as a function of the

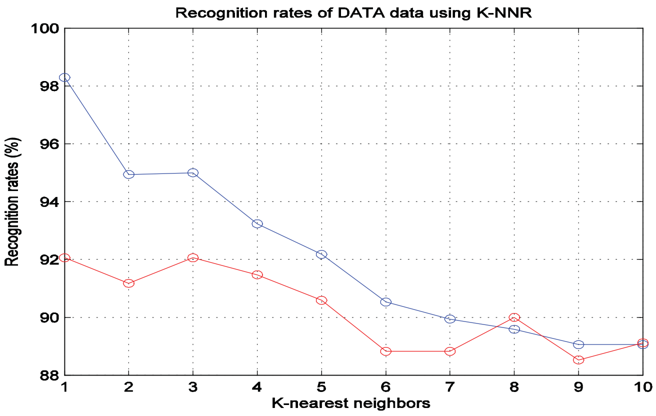


Figure 4: In blue, we plot the overall recognition rates of k-nearest neighbours rule as a function of k, considering a training set of 80% of the total of proteins, classifying the 100%. In red, we plot the overall recognition rates computed classifying only the 20% set aside from the training set.

number of neighbours considered, see Figure 4. The last rate computed is a better indicator of the performance of the rule, since any rule should do well in the train set. In fact, usually, one nearest neighbour achieves 100% detection in the train set. Even though, we want to compare with the PST classification rule itself, and the first rate is the one reported in Bejerano (2003).

The first thing to report is that the 98% of the whole set (counting the training set) has been well classified using the one nearest neighbour rule, against a 60% of true positives given by K-means. If we do not count the training set, the 92% of the test set is well classified with one nearest neighbour rule. It is interesting to notice that if we allow more than three points in the neighbourhood, we have less accuracy in the classification, giving evidence to the idea that the differences between the trees are very subtle. We have written in Table 1 the percentage of true positives per family as reported in Bejerano (2003). We should point out that we are using very short trees, and still we achieve the same rates of classification. The first three columns KNN1, KNN2 y KNN3 are the percentage of good classification training with 1, 2 and 3 neighbours, computed using a training set of 80% random proteins and scoring 100%. The last three columns show the same variables, but computed only over the 20% of samples that are not part of the training set. The percentages are reduced in some cases, but not very much, but the interpretability of the results and credibility of the experiment has been reinforced. We only compute these first three columns to compare with PST values, which are computed in this fashion.

FINAL REMARKS

Pattern recognition is an active field of research within engineering and computer science communities. Its main goal is to develop automatic methods for recognizing patterns in data. It encloses sub disciplines like discriminant analysis, feature extraction, error estimation and cluster analysis, among others. There are two specific methods that are well known in the literature of statistical pattern recognition that we have refereed here. They are

- K-means clustering
- k-nearest neighbours classification

We have addressed these two methods not in the conventional setting of a Euclidean space, but when the databases we have to classify and cluster consist of finite trees. We have considered a compact metric space of trees as the natural space where our database lies, and we have shown extensions of the two aforementioned procedures that apply in this new space. An interesting example of such database of trees is the one obtained when a general set of codified strings is modelled with a Variable Length Markov Chain. The VLMC is represented by its context tree, which can be estimated from each string using an algorithm like PST (Bejerano, 2003) or Context (Rissanen, 1983) leading to the final database of estimated trees. *If the codification is correct,*

Table 1: Family name, size: number of proteins in it, percentage of correct classification of PST method, percentage of classification of KNN1, KNN2 and KNN3 using 80% of the samples as a training test, and 100% as a test set, and 20% aside from training as test set.

Family	Size	PST	KNN1	KNN2	KNN3	KNN1	KNN2	KNN3
7tm_1	515	93	99.8	99.4	98.8	99.0	99.0	100.0
7tm_2	36	94.4	100	100	94.2	100	100	100
7tm_3	12	83.3	100	100	100	100	100	100
AAA	66	87.9	98.4	90.7	89.2	92.3	100	100
ABC_tran	269	83.6	95.5	87.6	88.4	77.7	74.0	74.0
actin	142	97.2	100	98.5	98.5	100	100	100
adh_short	180	88.9	95.5	87.1	89.9	77.7	75.0	80.5
adh_zinc	129	95.3	97.6	90.6	91.4	92.3	92.3	88.4
aldedh	69	87.0	98.5	95.5	92.6	92.8	85.7	92.8
Alpha-amylase	114	87.7	98.2	92.9	92.0	91.3	78.2	86.9
aminotran	63	88.9	95.1	74.1	77.4	76.9	46.1	69.2
ank	83	88.0	92.6	74.3	63.4	64.7	58.8	47.0
arf	43	90.7	100	97.6	100	100	100	100
asp	72	83.3	97.1	97.1	94.3	86.6	93.3	86.6
ATPsynt_A	79	92.4	96.1	100	96.1	100	100	100
ATPsynt_ab	180	96.7	100	97.7	99.4	100	100	100
ATPsynt_C	62	91.9	100	96.7	98.3	100	100	100
Betalactamase	51	86.3	98	92	90	90	90	90
bZIP	95	89.5	96.8	86.1	85.1	84.2	78.9	78.9
C2	78	92.3	94.8	66.2	83.1	75	75	68.7
cadherin	31	87.1	100	93.3	100	100	100	100
cellulase	40	85.0	92.3	76.9	76.9	62.5	50.0	62.5
cNMP_binding	42	92.9	100	90.2	87.8	100	100	100
COesterase	60	91.7	98.3	93.2	93.2	91.6	83.3	75.0
connexin	40	97.5	100	100	100	100	100	100
copper-bind	61	95.1	100	98.3	100	100	100	100
COX1	80	83.8	100	100	96.2	100	100	100
COX2	109	98.2	99.0	97.2	96.3	95.4	90.9	90.9
cpn10	57	93.0	100	98.2	98.2	100	100	100
cpn60	84	94.0	100	100	100	100	100	100
crystall	53	98.1	100	100	100	100	100	100
cyclin	80	88.8	94.9	89.8	87.3	75	75	68.7
Cys_knot	61	93.4	96.6	98.3	95	83.3	83.3	91.6
Cys-protease	91	87.9	95.5	94.4	91.1	77.7	77.7	77.7
cystatin	53	92.5	98.0	90.3	82.7	90.9	81.8	72.7
cytochrome_b_C	130	79.2	85.2	65.8	74.4	26.9	50.0	46.1
cytochrome_b_N	170	98.2	45.5	62.7	53.8	26.4	32.3	29.4
cytochrome_c	175	93.7	96.5	95.4	95.4	88.5	85.7	88.5
DAG_PE-bind	68	89.7	58.2	70.1	62.6	42.8	57.1	50.0
DNA_methylase	48	83.3	93.6	85.1	87.2	70.0	60.0	80.0
DNA_pol	46	80.4	97.7	91.1	91.1	88.8	77.7	77.7
dsrm	14	85.7	92.3	69.2	76.9	66.6	33.3	33.3
E1-E2_ATPase	102	93.1	96.0	90.0	87.1	80.9	76.1	76.1
efhand	320	92.2	95.9	94.6	92.7	92.1	92.1	92.1
EGF	169	89.3	92.8	84.5	85.7	85.2	85.2	70.5
enolase	40	100	97.4	97.4	97.4	87.5	87.5	87.5
fer2	88	94.5	97.7	93.1	90.8	88.8	88.8	94.4
fer4	152	88.2	97.3	90.7	90.7	87.0	70.9	83.8
fer4_NifH	49	95.9	100	97.9	97.9	100	100	100
FGF	39	97.4	100	97.3	100	100	100	100

we claim that the context tree of the chain will have all the information that is needed for discrimination by metric based methods.

In functional genomics, proteins are codified as strings of amino acids, and VLMC models are naturally fitted to functional families of such strings. Amino Acid chains are natural candidates for this type of modelling, but any suitable codification of an object with a finite alphabet will make this model arise, so other problems besides functional genomics could make profit of this type of approach. In classification of written reports, codification of the reports is usually made in order to reduce dimensionality or to extract features, leading to a database of strings, see Jeske and Liu (2004) and Jeske and Liu (2006). In these papers, the codification is derived carefully to ensure discrimination into "bad reports" or "good reports". But also, the codification could represent a conjecture made over style or prosody of speech or written text as it has been done by Veilleux *et al.* (1990) in a general setting, and Dorea *et al.* (1997), and Frota *et al.* (2001) for the case of detecting differences between Brazilian and European Portuguese. In this case, successful discrimination may give evidence to support the linguistic conjecture. We believe that our context can be of great importance for addressing several problems of Computational Linguistics, Natural Language Modelling and Speech Processing.

THANKS

AGF was partially supported by PICT 2005-31659. FGL is supported by FAPESP (grant 06/56980-0). We would like to thank Pablo Ferrari and Antonio Galves for many interesting discussions about metrics in the space of trees and modelling protein functionality.

BIBLIOGRAPHY

Altschul, S.F.; Madden, T.L.; Schaeffer, A.A.; Zhang, J.; Zhang, Z.; Miller, W. & Lipman, D.J. 1997. Gapped BLAST and PSI BLAST : a new generation of protein database search programs. *Nucleic Acids Research.* 25, 3389-3402.

Apostolico & Bejerano, G. 2000. Optimal amnesic probabilistic automata or how to learn and classify proteins in linear time and space, Proc. Int'l Conf. Computational *Molecular Biology,* 4, 25-32

Bejerano, G. 2003. Automata learning and stochastic modelling for bio sequence analysis, PhD thesis. Hebrew University.

---------- & Yona, G. 2001. Variations on probabilistic suffix trees: statistical modelling and prediction of protein families. *Bioinformatics,* vol.17:1, 2001, 23-43.

Balding, D.; Ferrari, P.; Fraiman, R. & Sued, M. 2008. Limit theorems for sequences of random trees. *Accepted at TEST. ArXiv: math.* PR/0406280.

Bateman, A.; Coin, L.; Durbin, R.; Finn, R.D.; Hollich, V.; Griffiths-Jones, S.; Khanna, A.; Marshall, M.; Moxon, S.; Sonnhammer, E.L.; Studholme, D.J.; Yeats, C. & Eddy, S.R. 2004. The Pfam protein families database. *Nucl. Acids Res.* 32, 90001, D138-141.

Dayhoff, M.O. 1976. The origen and evolution of protein superfamilies. *Fed. Proc.* 35, 2132-2138.

Dorea, C.; Galves, A.; Kira, E.; & Alencar, P. 1997. Markovian Modeling of the Stress Contours of Brazilian and European Portuguese, *Brazilian Journal in Probability and Statistics,* vol 11:2,161-175.

Einsenberg, D; Marcotte, E.M.; Xenarios, I. & Yeates, T.O. 2000. Protein function in the post genomic era. *Nature*, 405, 823-826.

Flesia, A. G. & Fraiman, R. 2007. A distance based test on random trees, Mecánica Computacional. Vol XXVI. pp 2016-2025. *S. Elaskar, E. Pilotta, G. Torres. (Eds),* [arXiv: 0708.1733v1] [math.ST]

Frota, S. & Vigario, M. 2001. On the correlates of rhythmic distinctions: The European/Brazilian Portuguese case. *Probus,* vol 13:2, 247-266.

Hefyi, H. & Gerstein, M. 1999. The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *J. Mol. Biol.* 288, 147-164.

Jeske, D. & Liu, R. 2004. Mining massive text data and developing tracking statistics, Classification, Clustering and Data Mining Applications, ed. D. Banks, *et al., Springer,* pp. 495-510.

---------- & ----------. 2006. Mining and Tracking Useful Features in Massive Text Data with Applications to Risk Management, *To Appear in Technometrics.*

Karp, R.M. 2002. Mathematical challenges from genomics and molecular biology, *Notices Amer. Math. Soc.,* 49(5),544-553.

Krause, Stoye & Vingron. 2000. The SYSTERS protein sequence cluster setr. *Nucleic Acids Res.* 28(1)270:272.

Leonardi, F.G. 2007. Sparse Stochastic Chains with application to classification and phylogeny of protein sequences. *PhD thesis Bioinformatics program.* Universidade de Sao Paulo.

----------. 2006. A generalization of the PST algorithm: modelling the sparse nature of protein sequences, *Bioinformatics,* vol 22:11, 1302-1307

Enright, A.J.; Van Dongen, S. & Ouzonis, C.A. 2002. An efficient algorithm for large scale detection of protein families. *Nucleic Acids Res.* vol.30,7,1575-1584.

Eskin,E.; Noble,W. & Singer, Y. 2000. Protein family classification using Sparse Markov Transducers. *Proceedings of the Eight International Conference on Intelligent Systems for Molecular Biology* (ISMB-2000). San Diego, California. August 20-23 2000.

Pearson, W.R. 2000. Flexible sequence similarity searching with FASTA 3 program package. Methods. *Mol. Biol.* 132:185-219.

Rabiner, L.R. Tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE,* 77:257-286.

Rissanen, J. 1983. A universal data compression system. *IEEE Trans. Inform. Theory* Vol. 29(5), 656-664.

Pollard, D. 1981. Strong Consistency of k-means clustering. *Annals of Statistics* Vol. 9 No. 1 135-140.

Sverdrup-Thygeson, H. 1981. Strong law of large numbers for measures of central tendency and dispersion of random variables in compact metric spaces. *Annals of Statistics* Vol. 9 No. 141-143.

Sasson, O.; Vaakin, Fleischer H.; Portugaly, E.; Bilu, Y.; Lineal, N. & Linial, M. 2003. ProtoNet: Hierachical classification of protein space. *Nucleic acid res.* 31(1):348—352.

Smith, O.; Annau, T.M. & Chandrasegaran. S. 1990. Finding Sequence Motifs in Groups of Functionally Related Proteins, *PNAS*, 87, (2), 826-830.

Van Dongen, S. 2000. Graph Clustering by Flow Simulation. *PhD thesis,* University of Utrecht.

Yona, G.; Lineal, N. & Lineal, M. 2000. ProtoMap: automatic classification of protein sequences and hierarchy of protein families. *Nucleic Acid Res.* 28(1):49-55.

---------- & Levitt, M. 2000. Towards a complete map of protein space based on a unified sequence and structure analysis of all known proteins. *In Proc. Int. Conf. Intell. Sys. Mol. Biol,* Vol 8, 395-406.