Protein Sequence Classification through Relevant Sequence Mining and Bayes Classifiers

Pedro Gabriel Ferreira[⋆] and Paulo J. Azevedo[⋆⋆]

University of Minho,
Department of Informatics
Campus of Gualtar, 4710-057 Braga, Portugal
{pedrogabriel, pja}@di.uminho.pt

Abstract. We tackle the problem of sequence classification using relevant subsequences found in a dataset of protein labelled sequences. A subsequence is *relevant* if it is frequent and has a minimal length. For each query sequence a vector of features is obtained. The features consist in the number and average length of the relevant subsequences shared with each of the protein families. Classification is performed by combining these features in a Bayes Classifier. The combination of these characteristics results in a multi-class and multi-domain method that is exempt of data transformation and background knowledge. We illustrate the performance of our method using three collections of protein datasets. The performed tests showed that the method has an equivalent performance to state of the art methods in protein classification.

1 Introduction

Concerning data where an order relation between atomic elements occurs, sequence data appears as a natural representation. An important and very useful operation to be done over sequence data is classification. The problem of classifying sequence data is to take a given set of class labelled sequences and build up a procedure to *a posteriori* assign labels to unlabelled sequences (queries). Many examples of the application of this task can be found in a variety of domains. Consider the case of biology/bioinformatics field where given a database of nucleotide sequences (DNA/RNA) or amino-acids sequences. Portions of the former sequences code for the latter through two mechanisms: *transcription* and *translation* [12, 6]. A sequence of amino-acids constitute a protein and is hereafter called as a protein sequence. A possible scenario would be the case where a biologist wants to find the respective family/domain or function of an unclassified sequence, for example a new synthesized protein. This problem is of critical

[⋆] Supported by a PhD Scholarship from Fundação para Ciência e Tecnologia, Ministério da Ciência e Ensino Superior of Portugal

[⋆⋆] Supported by POSI/2001/CLASS project, sponsored by Fundação Ciência e Tecnologia and European program FEDER

importance due to the exponential growth of newly generated sequence data in the last years, which demands for automatic methods for sequence classification. In the problem of sequence categorization/classification three types of methods can be distinguished:

- The *Direct Sequence Classifiers*, that exploit the sequential nature of data by directly comparing the similarity between sequences. Example of these type of classifiers is the k-Nearest Neighbour. In this method the class label of the k most similar sequences in respect to the query sequence are used to vote on a decision. Sequence similarity can be assessed through a method like FASTA [17] or BLAST [1].
- The *Feature based Sequence Classifiers*, that work by first extracting and model a set of features from the sequences and then adapt those features to accomplish with the traditional techniques, like decision trees, rule based classifiers, SVM's and many others. In [15, 16, 5, 4, 21] we have examples of these type of methods.
- The *Probabilistic Model Classifiers*, that work by simulating the sequence family under consideration. Typical probabilistic classifiers are the simple and k-order Markov Chain [19], Hidden Markon Models [14] and Probabilistic Suffix Trees [11].

Recently Probabilistic Suffix Trees (PSTs) [11] and Sparse Markov Transducers (SMTs) [8] have been applied in the protein classification problem, and have shown superior performance. A PST is essentially a variable length Markov Model, where the probability of a symbol in a sequence depends on the previous symbols. The number of previous considered symbols is variable and context dependent. The prediction of an input sequence is done symbol by symbol. The probability of a symbol is obtained by finding the longest subsequence that appears in the tree and ends just before the symbol. These probabilities are then combined to determine the overall probability of the sequence in respect to a database of sequences. One of the disadvantages of the PSTs is that the conditional probabilities of the symbols rely on exact subsequence matches. In protein family classification this becomes a limitation since substitutions of symbols by equivalent ones is often very frequent. SMTs are a generalization of PSTs that support wild-cards. A wild-card is a symbol that denotes a gap of size one and matches any symbol on the alphabet. In [11] an experimental evaluation has shown that PSTs perform much better than a typical BLAST search and as good as HMM. This is very interesting since the latter approach makes use of multiple alignments and the families are usually defined based on an HMM [9]. Additionally PSTs are a totally automotive method without prior knowledge (multiple alignments or score matrices) or any human intervention. In [8], SMTs have shown to outperform PSTs.

Our motivation to this work is to suggest a robust and adaptable classification method using a straightforward algorithm. We propose a multi-class sequence classification method which can be applied to data in many different domains, in particular to protein sequence data without requiring any type of data transformation, background knowledge or multiple alignment. Our proposal fits under the *direct sequence classifiers* type described before.

2 Preliminaries

Our method exploits global and local similarity of the sequences by extracting common subsequence patterns of different sizes that occur along the query sequence and the sequence families. These same patterns can then be used to interpret and understand the classification results.

Since our main concern is protein datasets we are only considering the alphabet of amino-acids. Each symbol of the sequence is generically called as an *event* and the distance between consecutive events as *gaps*. Considering the definition of a pattern as $A_1 - x(p_1, q_1) - A_2 - x(p_2, q_2) - \ldots - A_n$ where A_i are amino-acids and $-x-$ gaps greater than p_i and smaller than q_i, then two types of patterns can be distinguished:

- **Rigid Gap Patterns** only contain gaps with a fixed length, i.e. $p_i = q_i, \forall i$ The symbol "." is a wild-card symbol used to denote a gap of size one and it matches any symbol of the alphabet. Ex: $MN..A.CA$
- **Flexible Gap Patterns** allow a variable number of gaps between events of the sequence, i.e. $p_i \leq q_i, \forall i$. Ex: A-x(1,3)-C-x(4,6)-D.

The idea behind our method is that a sequence can be classified based on its *relevant* subsequences. A (sub)sequence is *relevant* if it is:

- *Frequent*, i.e. appears in at least an user defined number (*minimum support*) of database sequences.
- Has a *non trivial length*, i.e. satisfies a minimal defined length.

The problem that we address in this work can be formulated as follows: *Given a collection D of previously classified sequences, an unclassified query sequence, a minimum support (σ) and a minimal sequence length, determine the similarity of the query sequence w.r.t all the classes present in D.*

2.1 Sequence Patterns

Protein sequences of the same family typically share common subsequences, also called motifs. These subsequences are possibly implied in a structural or biological function of the family and have been preserved through the protein evolution. Thus, if a sequence shares patterns with other sequences it is expected that the sequences are biologically related.

Since the protein alphabet is small, many small patterns that express trivial local similarity may arise. Therefore, longer patterns are expected to express greater confidence in the sequences similarity.

Considering the two types of patterns, rigid gap patterns reveal better conserved regions of similarity. On the other hand, flexible gap patterns have a greater probability of occur by chance, having a smaller biological significance.

3 Method

Sequence pattern mining [20, 18, 2, 13] is the task of finding frequent patterns along the sequence data. A pattern is considered to be frequent if it occurs in the data sequences a number of times greater than a pre-determined threshold, called *support*. Besides providing valuable information about the data, these patterns have application in many areas like clustering or classification.

In this work we will make use of a method that consists in an adaptation of a sequence pattern mining algorithm [10] designed for the task of protein mining. The method reports all the frequent patterns occurring in a query sequence in respect to a user defined database. The query sequence is used to drive the mining process ensuring containment of the reported patterns. The algorithm allows a refined analysis by enumerating frequent patterns that eventually occur in a small subset of the database sequences. Two types of patterns (described in section 2), with variable or fixed length spacing between events, satisfying the user restrictions and associated options can be identified. The restrictions or constraints that the algorithm supports are:

- *Item Constraints*: restricts the set of the events (*excludedEventsSet*) that may appear in the sequence patterns,
- *Gap Constraints*: defines the (*minGap*) minimum distance or the maximum distance (*maxGap*) that may occur between two adjacent events in the sequence patterns,
- *Duration or Window Constraints*: defines the maximum distance (*window*) between the first and the last event of the sequence patterns.

Given a query sequence S and a collection of protein families D, applying the above algorithm, two parameters are obtained: *number of relevant patterns* and *average length of the patterns*. This information is then combined to determine the probability of S belonging to one of the families in D.

3.1 Bayes Classifier

The naïve Bayes Classifier is a simple probabilistic classifier. It is based on a probabilistic model that requires strong independence assumptions of the variables involved. Nevertheless, even in the cases where the independence assumption is not strictly satisfied the classifier performs well on a variety of situations, including complex real world situations [7]. The goal of the classifier is to assign a probability to one of the classes in $\{C_1, C_2, \ldots, C_n\}$, based on a d-dimensional vector of observed parameters, $\overrightarrow{f} = f_1 \ldots f_m$. This can be expressed through a conditional probability relation in the form $P(C_i|\overrightarrow{f})$. Using the Bayes Theorem this can be written as:

$$P(C_i|\overrightarrow{f}) = \frac{P(C_i) \times P(\overrightarrow{f}|C_i)}{P(\overrightarrow{f})} \tag{1}$$

$P(C_i)$ is known as the apriori probability of the class and can be obtained through the relative occurrence of C_i in the data. Since P($\overrightarrow{f}$) is class independent its value can be expressed as a constant value. Thus, equation 1 can be rewritten as:

$$P(C_i|\overrightarrow{f}) = \alpha_i \times \prod_{j=1}^{n} P(f_j|C_i) \tag{2}$$

where α_i is a constant value for the respective class C_i. For our classification problem, the vector consist only of two parameters: total number and average length of the extracted relevant subsequences. We assume that they are statistically independent, although this is not entirely the case.

In our work three different models are studied and compared. These models are slightly variations of the model in 2. In the first model (A), the apriori probability of the class is not taken into account, thus in equation 2: $\alpha_i = 1 \ \forall_i$.

When a query sequence S is analyzed against a database D, it is naturally expected that the number of extracted patterns is proportional to the number of sequences in D. To avoid the bias due to the different databases length the probability is normalized by the length of the class. In equation 2, $\alpha_i = \frac{N}{|C_i|}$ where $N = \sum_{i=1}^{n} C_i$, i.e. it corresponds to the inverse of the apriori probability of C_i. Finally, in model C, the parameter "average length" is given a greater relative weight than the parameter "number of patterns" and in equation 2, $P(f = avgLength|C_i)$ is raised to a power of three.

Now for the three models, and given the feature vector $\overrightarrow{f}$ of a sequence S, the classification is simply given by:

$$max\{P(\overrightarrow{f}|C_i) \ \forall_i\} \tag{3}$$

4 Results and Discussion

To evaluate our method, we configured our query driven miner to extract rigid gap patterns. Only two types of constraints were applied: maxGap and Window, with a value of 15 and 20, respectively. These constraints allow a confinement of the search space and make possible the mining in interactive time. The minimal length of the extracted patterns is two. We used three collections of protein families. A smaller collection was obtained from Pfam [9] version 17.0. Most of the proteins in this collection were taken from the top twenty list of April 2005. This collection gave us the first insights in the performance of the method. The second collection is composed of 50 sequences, obtained from Pfam version 1.0, and can be downloaded from [9]. This set of families was already used in [8] and will allow a direct comparison with the PSTs and SMTs. Due to the constant refinement in the topology of the PFam database we should note that there are significant differences in the families common to the two collections. The third dataset consist on 27 families from the receptors group entries on the Prosite [3] database.

All the methods are assessed based on the precision rate (PR) measure:

$$PR = \frac{NumCorrect}{NumTested} \times 100\% \tag{4}$$

The method was evaluated using the "leave-one-out" methodology[1]. The classification result is determined by equation 4. The evaluation in [11] and [8] was different from ours. They used 4/5 of the family sequences to build a model for the respective family and evaluated the model with the remaining 1/5 of the sequences. Unfortunately, since there is no indication on how the folds were created, their experiment could not be totally recreated.

The only parameter required by our model is the support value. Since we do not have a way to apriori define this value, it was determined empirically. For each family we measured the average time to mine the largest, the smallest and two medium size sequences of the respective family. If the average time was approximately below one minute than that support value was used for that family. The reason for the use of this criterion is that the performance of the mining process directly depends on the support value and on the density (similarity between the sequences) of the family. Thus, if the support is set to low values in the more dense families the mining process becomes very time consuming.

All the experiments were performed on 1.5GHz Intel Centrino machine with 512MB of main memory, running windows XP Professional. The mining application was written in C++ language.

In table 1 we present the classification results for the collection of 26 protein families. In the left columns we have the name, the number and the average length of the sequences in the family. The intra similarity of the family is also presented (see [9]). The fifth column shows the support used to mine the respective family. In the right side of the table, the precision rate for the three probabilistic models of our method is presented. The last column shows the average time that it takes to mine each sequence of the family. This value has to be multiplied by the total number of families to give the total amount of time spent mining the sequence against all the families. From the presented results we can see that the prediction rate is around or above the 80%, except for the PPR and the TPR-1 family. In these cases, the number of missed sequences is extremely large. These results can be explained due to a combination of small intra-similarity, low average length and a large family size, resulting in a relative small number of common patterns shared by the sequences of the family. On the other hand, the small length of the sequences leads the query sequences to share more patterns with the families with a greater average length. Model B gives particularly bad results for these cases since the multiplication factor imposes a big penalty in the classification probability. Table 2 shows the average classification results for table 1, when all the families are considered (row 2), when the family PPR is left out (row 3) and when PPR and TPR are both left out of the classification (row 4).

[1] The complete set of the results and the datasets can be obtained from the authors.

Name	Size	AvgLen	Intra-Sim	Supp.	A(%)	B(%)	C(%)	Time(secs)
7tm-1	64	269	19	2	100.0	100.0	100.0	0.18
7tm-2	33	263	25	2	100.0	100.0	100.0	2.65
7tm-3	30	256	27	2	100.0	100.0	100.0	3.34
AAA	245	194	25	2	95.3	88.4	97.7	0.26
ABC-tran	65	191	26	2	100.0	100.0	100.0	0.22
ATP-synt-A	30	162	52	2	78.6	96.4	89.3	0.20
ATP-synt-ab	157	232	54	6	98.9	98.9	98.9	1.29
ATP-synt-C	35	69	49	2	93.9	97.0	97.0	0.09
c2	409	76	23	2	89.2	46.7	91.3	0.13
CLP-protease	88	182	41	2	91.8	85.9	94.1	0.14
COesterase	129	541	27	2	86.5	58.7	87.3	1.73
cox1	24	461	48	2	90.9	90.9	90.9	0.41
cox2	32	117	60	2	96.7	100.0	100.0	0.43
Cys-knot	24	103	37	2	95.5	100.0	100.0	0.09
Cytochrom-B-C	9	101	74	2	100.0	100.0	100.0	1.61
Cytochrom-B-N	8	199	69	2	85.7	100.0	85.7	0.15
HCV-NS1	10	347	51	2	100.0	100.0	100.0	6.90
Oxidored-q1	33	284	28	2	90.3	93.5	93.5	0.26
Pkinase	54	274	24	2	98.1	90.4	96.2	0.21
PPR	558	36	20	2	11.9	0.0	22.6	0.03
RuBisCO-large	17	310	79	10	81.3	87.5	81.3	0.72
rvt-1	164	219	74	4	76.5	75.5	79.6	0.23
RVT-thumb	42	71	89	4	88.2	97.1	91.2	2.84
TPR-1	569	35	18	2	43.6	10.5	54.9	0.04
zf-C2H2	196	25	37	2	83.6	64.0	88.4	0.03
zf-CCHC	208	19	51	2	100.0	100.0	100.0	0.04

Table 1. Classification results of the three models, for the collection of 26 sequences from Pfam 17.0.

In table 3 we compare the three probabilistic models with the results from the PSTs and SMTs published in [11, 8]. In the last row of the table we present the average classification results. We can see that the precision rates of all classifiers are above the 90% threshold and that SMT ranks at the top. We should remind that this is a raw comparison since different evaluation methods were used. For our method all the sequences were evaluated, in this sense our evaluation provides more confidence on the presented results. Besides, at the cost of an extra computational work, the precision of a class can be increased by setting the support of that class to a lower value.

We applied a two-tailed signed rank test [22] to study if the medians of the classifiers C, PST and SMT are statistically equal. It was tested as a null hypothesis that medians for the pairs of classifiers C and PST, C and SMT are equal. For the first pair, the null hypothesis is rejected, thus the medians of the classifiers are significantly different. In the second case the null hypothesis is accepted, consequently there is no significant difference between the medians for

PrecisionRate	A(%)	B(%)	C(%)
All	87.6	83.9	90.0
without PPR	90.6	87.3	92.7
without PPR and TPR	92.5	90.5	94.3

Table 2. Average classification results for the collection of 26 proteins from Pfam 17.0

the classifiers C and SMT, with a level of significance of 0.05 and a p-value of 0.34.

In terms of computational demands, our method has low requirements. Since the mining algorithm only counts and does not collect the frequent patterns, it required a maximum of 5 MB of memory usage. This is in contrast with HMMs, PSTs and SMTs which are known to have high memory requirements.

As a last experiment we selected a set of proteins that match the patterns in the group of Receptors from the PROSITE [3] database. This group contains 27 entries matching a total of 13458 protein sequences. Table 4 contains the name and the size of each group of sequences and the respective support used. Next, we randomly selected 30% of the sequences of each group. Based on the percentage of the true positives (main diagonal) and false negatives we built a similarity matrix for the 27 groups of sequences. Figure 1 displays the similarity matrix, where each row and column represent the entries listed in table 4. Dark areas represent a higher number of class hits.

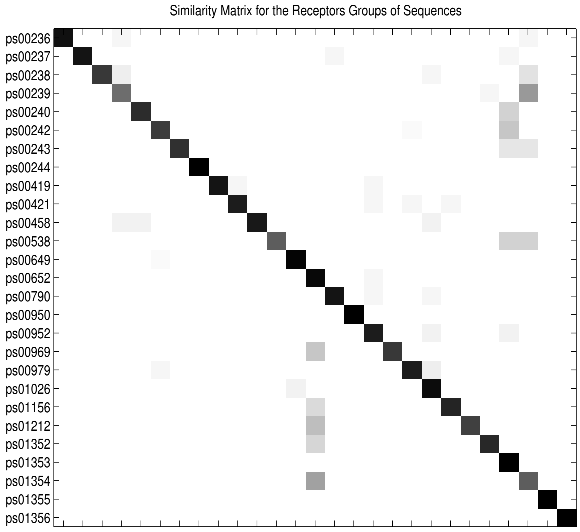

Fig. 1. Similarity Matrix for the classification performed on the 27 families of the set of sequences that match the entries in the Prosite Database.

Name	Supp	Size	A(%)	B(%)	C(%)	PST(%)	SMT(%)	Time(secs)
7tm-1	14	530	90.2	41.3	90.4	93.0	97.0	0.34
7tm-2	3	36	97.2	97.2	97.2	94.4	97.2	1.32
7tm-3	3	12	100.0	100.0	100.0	83.3	100.0	0.57
AAA	15	79	87.3	89.9	89.9	87.9	94.9	0.37
ABC-tran	20	330	92.1	62.7	94.8	83.6	93.3	0.43
actin	100	160	86.3	86.3	86.9	97.2	97.5	0.80
ATP-synt-A	7	79	82.3	82.3	83.5	92.4	94.9	0.09
ATP-synt-ab	12	183	88.5	84.2	90.7	91.9	96.8	15.68
ATP-synt-C	14	62	96.8	98.4	100.0	96.7	100.0	0.10
c2	15	101	95.0	95.0	94.1	92.3	96.0	0.08
COesterase	5	62	91.9	93.5	88.7	91.7	90.3	0.35
cox1	4	80	100.0	100.0	100.0	83.8	97.5	0.17
cox2	10	114	91.2	91.2	93.0	98.2	95.6	1.27
Cys-Knot	2	61	86.9	91.8	91.8	93.4	100.0	0.07
Cys-protease	4	95	94.7	94.7	94.7	87.9	95.1	4.94
DAG-PE-bind	2	108	97.2	97.2	99.1	89.7	95.4	5.82
DNA-methylase	2	57	86.0	100.0	89.5	83.3	91.2	0.71
DNA-pol	4	51	88.2	98.0	94.1	80.4	88.2	0.35
E1-E2-ATPase	20	117	94.9	80.3	94.0	93.1	94.0	0.18
EGF	2	676	99.6	97.8	99.6	89.3	98.8	0.05
FGF	7	39	100.0	100.0	100.0	97.4	100.0	2.53
GATase	2	69	94.2	100.0	95.7	88.4	94.2	0.12
GTP-EFTU	40	184	91.3	82.6	92.4	91.8	98.4	0.23
HLH	3	133	97.7	95.5	98.5	94.7	98.5	0.05
HPS70	25	171	88.9	80.7	94.7	95.7	98.2	0.35
HSP20	40	132	97.0	95.5	97.0	94.6	96.2	0.14
HTH-1	2	101	100.0	100.0	100.0	84.2	85.1	0.11
HTH-2	2	65	89.2	93.8	90.8	85.7	81.5	0.09
KH-domain	2	51	88.2	90.2	88.2	88.9	84.0	2.30
Kunitz-BPTI	2	79	98.7	100.0	100.0	90.9	92.3	7.30
MCP-signal	7	24	100.0	100.0	100.0	83.3	100.0	0.20
MHC-I	125	151	97.4	96.7	98.0	98.0	100.0	0.60
NADHdh	3	61	96.7	98.4	96.7	93.0	98.4	0.18
PGK	10	51	90.2	98.0	98.0	94.1	98.0	0.46
PH	5	77	93.5	94.8	96.1	93.3	83.1	4.36
Pribosyltran	4	45	86.7	93.3	88.9	88.9	95.6	0.11
RIP	3	37	86.5	91.9	89.2	94.6	91.9	0.13
RuBisCO-large	250	311	99.7	98.7	99.7	98.7	99.7	9.79
RuBisCO-small	20	107	98.1	97.2	99.1	97.0	99.1	4.86
s4	28	54	87.0	92.6	90.7	92.6	96.3	0.12
s12	12	60	85.0	90.0	90.0	96.7	100.0	1.20
SH2	3	150	100.0	98.0	100.0	96.1	98.7	0.10
SH3	2	161	98.8	98.1	98.8	88.3	96.9	12.31
STphosphatase	15	88	100.0	100.0	100.0	94.2	97.7	6.37
TGF-beta	15	79	92.4	92.4	92.4	92.4	98.7	0.14
TIM	4	42	95.2	100.0	97.6	92.5	100.0	0.15
TNFc6	2	91	79.1	89.0	81.3	86.2	93.4	0.04
UPAR-c6	2	18	94.4	100.0	94.4	85.7	94.4	0.68
Y-phosphatase	8	122	87.7	83.6	89.3	91.3	96.7	0.04
Zn-clus	2	54	96.3	100.0	96.3	81.5	90.7	0.05
Avg. PR(%)			93.1	93.3	94.5	91.0	95.4	

Table 3. Classification results of the three models, PSTs and SMTs, for the collection of 50 sequences from Pfam 1.0.

Dataset	Size	support(%)
ps00236	772	0.03
ps00237	8522	0.017
ps00238	862	0.03
ps00239	157	0.05
ps00240	119	0.06
ps00242	170	0.03
ps00243	116	0.3
ps00244	22	0.99
ps00419	521	0.18
ps00421	226	0.05
ps00458	20	0.5
ps00538	44	0.2
ps00649	233	0.04
ps00652	372	0.014

Dataset	Size	support(%)
ps00790	101	0.2
ps00950	121	0.2
ps00952	52	0.35
ps00969	89	0.05
ps00979	82	0.12
ps01026	36	0.08
ps01156	417	0.025
ps01212	82	0.15
ps01352	104	0.11
ps01353	66	0.03
ps01354	48	0.05
ps01355	71	0.08
ps01356	33	0.2

Table 4. Identifier of the Prosite entries (Dataset), number of sequences in the Swiss-Prot database that match the respective entry and the relative support value used for sequence classification.

4.1 Factors that affect the performance of the method

Although we do not have the exact values, we verified that all the protein families in the second dataset, have a high intra-similarity. This explains the high precision values of the three methods in the second evaluation.

From the three probabilistic models of our method, the one with higher precision in the performed evaluations is model C. It seems that the average length of the patterns has a bigger impact in the sequence classification. As naturally expected, the lower it is the support value the higher the precision rate of the models is. Lower support values allow finding more common patterns between the query sequence and smaller subsets of families sequences. The support values used for the two evaluations establish a reasonable trade-off between the performance and the precision of the method. A very important aspect of this method is that the extracted motifs reveal local and global similarity. One of the aspects pointed in [8] for the success of SMTs is the use of common short subsequences patterns that contain wild-cards. These wild-cards allow to describe the positions of the patterns that can be occupied by two or more amino-acids. Our method incorporates this feature through the rigid gap patterns. We do not include any type of biological information. We believe that the introduction of Equivalent/ Substitution sets of amino-acids can further improve the precision of the method. These sets will permit that during the mining process an event can be substituted by another event belonging to the same set without lost of equivalence. Additionally, the introduction of a discrimination score for the most biological or statistical relevant patterns may result in another improvement.

5 Conclusions and Future Work

In this article we presented a method of straightforward implementation to perform multi-class and multi-domain sequence classification. The method does not require background knowledge or any change in the sequence representation. It extracts two features, number and average length of the frequent patterns, which the query sequence has in relation to the sequences families. These features are then combined through a Bayesian classifier. We present three probabilistic models based on this classifier. When compared to two state-of-the-art methods, PSTs and SMTs, for protein sequence classification the method shows very promising results. Our method performs better than PSTs and has an equivalent performance to SMTs.

As a future work, we plan to evaluate and compare our method with collections of protein families with lower homology. We believe that our method may have superior performance to other methods in those cases. We are also seeking for a way to apriori determine the support value to be used for each family. Finally, we aim to extend our pattern mining process to determine the biological significance of the patterns. This will allow discriminating the weight of each pattern in the overall classification results.

References

1. Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schaeffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.
2. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the 8th International Conference of Knowledge Discovery and Data Mining SIGKDD, S. Francisco, July 2002.*, pages 429–435, 2002.
3. A. Bairoch. Prosite: a dictionary of sites and patterns in proteins. *Nucleic Acids Res*, 25(19):2241–2245, 1991.
4. A. Ben-Hur and D. Brutlag. Remote homology detection:a motif based approach. *Bioinformatics*, 19(1):26–33, 2003.
5. A. Ben-Hur and D. Brutlag. Sequence motifs: highly predictive features of protein function. In *In Proceeding of Workshop on Feature Selection, NIPS - Neural Information Processing Systems*, December 2003.
6. Necia Grant Cooper. *The Human Genome Project, Dechiphering the blueprint of heredity*, volume 1. University Science Books, 1994.
7. P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.
8. E. Eskin, W. N. Grundy, and Y. Singer. Biological sequence analysis: Probabilistic models of proteins and nucleic acids. *Journal of Computational Biology 10(2)*, pages 187–214, 2003.
9. A. Bateman et al. The pfam protein families database. *Nucleic Acids Research*, vol 32, Database issue, October 2003.

10. Pedro Ferreira and Paulo Azevedo. Protein sequence pattern mining with constraints. In *Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Porto, October 2005.
11. Bejerano G. and Yona G. Modeling protein families using probabilistic suffix trees. In ACM press, editor, *In the proceedings of RECOMB1999*, pages 15–24, 1999.
12. Lawrence Hunter. Molecular biology for computer scientists (artificial intelligence & molecular biology).
13. A.Floratos I. Rigoutsos. Combinatorial pattern discovery in biological sequences: the teiresias algorithm. *Bioinformatics*, 1(14), January 1998.
14. Mian Sojlander Krogh, Brown and Haussler. Hidden markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, (235):1501–1531, 1994.
15. Daniel Kudenko and Haym Hirsh. Feature generation for sequence categorization. In *AAAI/IAAI*, pages 733–738, 1998.
16. Neal Lesh, Mohammed J. Zaki, and Mitsunori Ogihara. Mining features for sequence classification. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 342–346. ACM Press, 1999.
17. R.W. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proceedings Natl. Academy Sciences USA*, 5:2444–2448, 1998.
18. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In *Proceedings Int. Conf. Data Engineering (ICDE'01), Heidelberg, Germany, April 2001*, pages 215–226, 2001.
19. Durbin R. and Eddy S. R. Biological sequence analysis: Probabilistic models of proteins and nucleic acids. *Cambridge University Press*, 1998.
20. Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, editors, *Proc. 5th Int. Conf. Extending Database Technology, EDBT*, volume 1057, pages 3–17. Springer-Verlag, 25–29 1996.
21. N.M. Zaki, R.M. Ilias, and S. Derus. A comparative analysis of protein homology detection methods. *Journal of Theoretics*, 5.
22. J. H. Zar. *Biostatistical Analysis 3rd Edition.* Prentice Hall, 1999.