Understanding Signal Sequences with Machine Learning

Jean-Luc Falcone[1,⋆], Renée Kreuter, Dominique Belin[2], and Bastien Chopard[1]

[1] Département d'informatique, Université de Genève, 1211 Genève 4, Switzerland
[2] Département de Pathologie et d'Immunologie, Université de Genève, Switzerland

Abstract. Protein translocation, the transport of newly synthesized proteins out of the cell, is a fundamental mechanism of life. We are interested in understanding how cells recognize the proteins that are to be exported and how the necessary information is encoded in the so called "Signal Sequences". In this paper, we address these problems by building a physico-chemical model of signal sequence recognition, using experimental data. This model was built using *decision trees*. In a first phase the classifier were built from a set of features derived from the current knowledge about signal sequences. It was then expanded by feature generation with *genetic algorithms*. The resulting predictors are efficient, achieving an accuracy of more than 99% with our wild-type proteins set. Furthermore the generated features can give us a biological insight about the export mechanism. Our tool is freely available through a web interface.

1 Introduction

1.1 Signal Sequences

Proteins synthesized in the cell must be transported to the correct cellular compartment so that they can achieve their role. This process is called protein targeting and is a fundamental aspect of cell protein metabolism [1]. For instance blood plasma proteins and polypeptidic hormones must be delivered to the extracellular space. We are interested in the secretion pathway, which involves the targeting and transport of the proteins out of the cell. The protein complex (called *translocon*) which exports the proteins varies from one species to another.

All the proteins that must be exported, carry a particular region of conserved function, the *signal sequence* (SS) or *signal peptide*, located in N-terminal extremity. The length varies slightly from 10 to 50 amino-acids (AA). The protein is exported before folding and the SS is usually cleaved after the export. The precise location where the cleavage occurs is called the cleavage site.

The most interesting feature of SS is their inter- and intra-species variability. Their sequence as well as their length vary. Thus, they do not carry any systematic consensus. However, three properties have been proposed as distinguishing

⋆ Supported by the Swiss National Science Foundation.

E. Marchiori, J.H. Moore, and J.C. Rajapakse (Eds.): EvoBIO 2007, LNCS 4447, pp. 57–67, 2007.
© Springer-Verlag Berlin Heidelberg 2007

features of SS [2]: (i) They begin with an N-terminal region which includes one or several positively charged lysine or arginine residues. This region is called the *N-Region*. (ii) Following the N-Region, SS contain a stretch of hydrophobic AA forming the so-called *H-Region*. (iii) In the majority of secreted proteins there is a third region, the *C-Region*, located between the H-Region and the cleavage site. It carries a weak consensus recognized by the leader-peptidase.

The above properties are too vague to easily determine whether or not a protein will be secreted. The hypothesis that these three regions, as defined above, characterize an exported protein is based on observations made on known SS. However there are no experiments that confirm that these three properties are sufficient and/or relevant for the recognition process. To address the problem of correctly discriminating secreted proteins from the other ones (cytosolic), artificial intelligence techniques have been considered.

1.2 Computer Predictions of Signal Sequences

Many methods for the recognition of SS have been proposed. For all these methods, the predictors are built by an algorithm based on supervised learning techniques. The SignalP method is currently considered as the best classification method [3] and is the most widely used. It consists of two feed-forward neural networks [4]. The first is trained to recognize the SS itself; the second recognizes the cleavage site.

SignalP cannot help us understand the physico-chemical properties recognized by the translocon. The problem is intrinsic to the nature of classical neural networks which does not allow the retrieval of high-level symbolic rules. Another weakness resides in the fact that SignalP uses the existence of a valid cleavage site as a strong classification criterion. Although it is true that most SS include such a site, there are proteins like ovalbumin which are exported but lack a cleavage site. Furthermore mutated SS are poorly recognized by existing predictors. Therefore there is a need to develop new approaches with better prediction scores on such proteins and which gives better insight into the mechanisms at work.

1.3 Decision Trees

In this paper we propose a novel approach based on *decision tree* classifiers to understand how SS are recognized. Decision trees are classification programs that can classify objects according to properties (or features). Different algorithms exist to build such trees from a list of properties and a set of example objects representative of the different classes. As it is the case with neural networks, the learning strategy is supervised, i.e. based on a training set of sequences. The output of this algorithm is a tree in which the non-terminal nodes are evaluations based on the properties characterizing the objects being classified. The leaves of the tree (the terminal nodes) are the possible classes.

An important advantage of decision tree building algorithms is that only the properties necessary for the classification are retained. The most discriminant properties appear at the root of the tree and the less discriminant ones are near

the leaves. This allows us to identify the classification mechanisms explicitly, from the most relevant to the least important one.

1.4 Machine Learning to Investigate Signal Sequences

Our goal was to apply the decision tree method to the problem of SS recognition. We started our investigation using only the N- and H-regions, because the C-region is recognized by the cleavage enzyme after export but probably not by the translocon itself.

Note that since these regions were only described qualitatively, it is necessary to specify a procedure to measure them from a given protein sequence. We propose in Sec. 2.2 a way to compute them.

The purpose of building a decision tree is not only the construction of a new and better SS predictor, but also to show if one or more of the properties mentioned above are relevant and sufficient. If we can reach good efficiency using only these properties we can conclude that they are indeed sufficient. If not, an extended set of features needs to be considered.

Hence, to generate these extra features (or to refine the parameters of existing ones) we have considered feature generation with genetic algorithms. The evolution will optimize individuals whose genes are potential new features. Those features are mathematical functions modeling the various physico-chemical interactions between translocon and the N-terminus of protein. For this purpose, they compute a score for each protein according to the physico-chemical properties of the AA. To evaluate the fitness of the individuals, we add its gene (the new features) to the existing features described above. The fitness is then the performance of the resulting decision tree. This method can lead us not only to improve our efficiency but to suggest new biological criteria and new wet-lab experiments to confirm them.

In a first phase we focus our research on *E. coli*. This choice is motivated by the large number of experimentally known secreted proteins and the existence of a collection of mutant SS.

2 Decision Trees for Signal Sequences

2.1 Datasets

We used four datasets: (i) 104 wild-type *E. Coli* proven signal sequence proteins dataset ; (ii) 160 wild-type *E. Coli* cytoplasmic proteins; (iii) a collection of 17 signal-defective mutants of the *phoA*, *malE* and *rbsB* genes; (iv) a collection of 145 maspin "gain-of-function" mutants and derived constructions.

The datasets (i) and (ii) were obtained from the UniRef100 database as follows: the set (i) was composed only the proteins with proven signal sequence annotations after removal putative TAT proteins with the TatP predictor [5]; the set (ii) was composed of proteins containing the line: `CC -!- SUBCELLULAR LOCATION: Cytoplasm`. The sequences of all datasets were truncated to the first

70 AA. The final dataset was made by pooling together the four datasets for a total of 427 instances (130 SS and 297 cytoplasmic proteins).

We lack space to describe properly the mutants collection, but all the datasets, their descriptions and references are available online on [6].

2.2 Computation of the Signal Sequence Features

In order to build our decision tree and to assign specific values to the SS features (properties), it is necessary to find a way to define the N- and H-regions and to attribute them a score.

H-Region Features: The H-region score quantify the presence of a stretch of hydrophobic AA. The difficulties in scoring the H-region are (i) to identify the relevant stretch, (ii) to choose the most appropriate hydrophobicity scale and (iii) to find the proper way to combine the hydrophobicity of each residue along the stretch.

We compute four different H-region features, mixing two different hydrophobicity scales with two different calculations. The two scales are (i) AA hydropathy [7]; (ii) percentage of buried residues [8]. The first is a composite index obtained from different measurements and modified to match biological insight. This scale is the most widely used in bioinformatics experiments. The second derives from statistical structural data and only takes the distribution of a residue at the surface or in the core of a set of globular proteins into account.

These two scales can be used in two different ways to define the H-region feature score: we scan the sequence searching for the most hydrophobic stretch of AA, or the longest one. A stretch is defined as a contiguous substring in which all AA have an hydrophobic score larger than a given threshold (with respect to the chosen scale). The threshold used is given by the average hydrophobicity of the 20 AA minus the standard deviation. Among all the stretches, we select the best one. In the first calculation method we compute the sum of the hydrophobicity of all its AA for each stretch. The H-region is then defined as the stretch with highest sum. The final score is this largest sum. In the second calculation method, the H-region is defined as the longest stretch. The final score is its length.

We chose the following terminology for these four features: H_{LK} is the score based on the Kyte-Doolittle scale and returning the longest stretch. H_{VK} is based on the same scale but returns the value of the most hydrophobic stretch. We use the same notation scheme for the score derived from the percentage of buried residues: H_{LB} and H_{VB}.

N-Region Features: In order to determine the N-region, we used the fact that the N-region terminates where the H-region starts. Thus, the H-region had to be computed first, as explained above. We observe that the four possible ways to extract the H-Region give the same N-region. This is due to the fact that the two hydrophobicity scales discussed above select the same AA as hydrophobic, although they return different final values.

Once the N-region has been identified, we compute its score. We again propose two different features. The first one is the average charge $N_Q(p)$ of protein p and the second one is the lysine-arginine density $N_R(p)$:

$$N_Q(p) = \frac{1}{n(p)} \sum_{i=2}^{n(p)+1} q(p_i)\,, \qquad N_R(p) = \frac{1}{n(p)} \sum_{i=2}^{n(p)+1} r(p_i)$$

where $n(p)$ is the length of the N-Region, p_i the ith AA of the protein and $q(p_i)$ is the charge of AA p_i. $r(p_i)$ is equal to 1 if p_i is a lysine or arginine, 0 otherwise. The difference between these two features is that charged aspartate and glutamate residues are neglected in N_R.

2.3 Tree Building Algorithm and Test Procedure

We want to be able to classify a protein according to its location inside or outside the cell, from the six features described above. Here we consider the *C4.5* tree building algorithm [9] because of the good open-source Weka 3 [10] implementation. Several parameters of the *C4.5* algorithm were tested, but the best results were obtained by setting the *pruning confidence* parameter to 0.15 and the *minimum number of instances* parameter to 2.

We were mainly interested in the accuracy of the classifier: $A = \frac{a+d}{N}$ where A is the accuracy, a the number of true positives, d the number of true negatives and N the total number of instances.

To detect over training we used stratified cross-validation tests. This technique works as follows: the dataset (both signal and cytosolic sequences) is split in 10 subsets S_1,...,S_{10} using random sampling preserving the ratio between positive and negative instances. At each step $1 \leq i \leq 10$, a tree T_i is built with all S_j such that $j \neq i$. The accuracy A_i of step i is computed by applying tree T_i to S_i. The final score is the average of A_i. Note that the final tree is the one obtained with the full dataset. Thus, the stratified cross-validation method tests the robustness of the tree building algorithm with respect to the dataset.

2.4 Results

The resulting tree, called SigTree, was built according to all properties discussed above, and is presented in fig. 1. After a first score-node corresponding to the N_Q score, the tree splits in two main branches. The first one (low N_Q score) corresponds to 91.2% of all 297 cytosolic sequences, whereas the second branch, corresponds to 77.7% of all 130 SS.

SigTree achieves an accuracy of 98.2% on the whole dataset, and a cross-validation accuracy of 89.9% which is significant. We can now test separately the wild type proteins (set i and ii) and the two mutants collections on the resulting tree. The results are summarized on tab. 1.

As we said above, the detection of a cleavage site may not be a good indicator of protein secretion. To verify this assumption, we built another tree by adding a cleavage site feature as described in [11]. The results were almost identical to

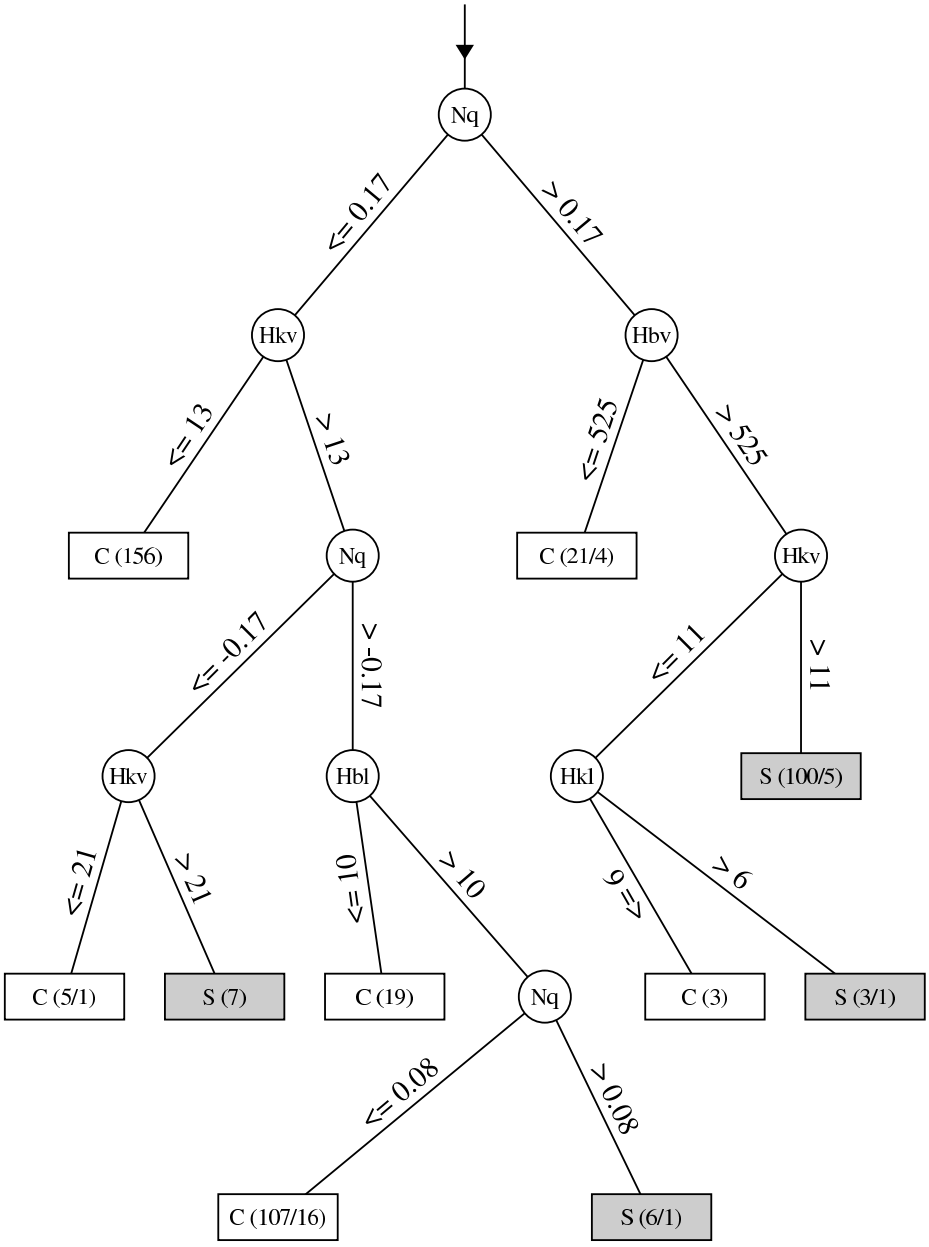

Fig. 1. SigTree. The circle nodes represent the features defined above. The square leaves are the predicted classes. The first number between parenthesis represents the total number of instances classified in the leaves; the second number is the number of dataset sequences wrongly classified (it is omitted if all the sequences are well classified).

Table 1. SigTree and SigTreeGA accuracy on the different datasets (computed on the entire sets)

Dataset	SigTree	SigTreeGA
i+ii	97.4%	99.2%
iii	64.7%	70.6%
iv	89.7%	97.2%

the tree presented above. The cross-validation was only slightly better: 90.4% instead of 89.9%. Therefore we decided to omit this feature in our predictor.

3 Features Generation with Genetic Algorithms

We now want to extend our set of six initial features using Genetic Algorithms (GA). An individual in the GA population will code for a set of new features. These features represent measures of interactions of the N-terminus of proteins with the translocon. Since we do not know how many new features are necessary to improve the classifiers, we used variable size individuals. Those indivuals ranged between one and seven genes and were initialized with four random genes.

As those interactions are depending on the AA physico-chemical properties we consider for each AA the values of its physico-chemical properties. However the large number of AA properties compiled in the database *AAIndex* [12] (more than 500) is impracticable. Thus we have reduced the existing set with the method described in [13] to a set of 45 clusters. This set is available on-line on [6]. A feature is now defined by two components: (i) one of the 45 physico-chemical properties P; (ii) a reduction function which combines the value of P for the AA of the given protein. Each reduction function represents a possible type of interaction between a newly synthesized protein and the translocon.

3.1 Genetic Algorithm Components

Genes: Each gene of the individuals is a possible feature. When a feature is applied on a protein, we first generate a numeric array by replacing each AA with its corresponding value of the coded property. The the reduction function is then applied on this array and returns one number: the feature score for the protein.

We have designed four types of reduction functions defined by an algorithm and structural parameters. Those parameters are optimized by the GA evolution. The reduction functions we chose and their associated parameters are:

Sliding Windows: A sliding window reduction returns the minimum or the maximum sum of all substring of l AA along the protein.

Parameters: the window size l, a Boolean indicating whether the max or min of the sums is chosen.

Stretch operator: This reduction works as the *H-region* feature of Sect. 2.2 but now with an adjustable threshold value.

Parameters: a number representing the threshold, a Boolean indicating whether the highest stretch or the lowest one is returned.

Helix Window: This function assumes that the SS has an helical structure (not necessarily an α-helix) of period λ AA. For all λ possible offsets, it computes the sum of the AA along the SS, with stride λ.

Parameters: an integer λ representing the helix period in AA, a Boolean indicating whether the max or min of all the sums is chosen.

Sliding Helix Window: This reduction function is a combination of the *sliding window* and the *helix window*. It proceeds like the latter but sums up only l AA at time. For example, for $\lambda = 3$ and $l = 4$, we will first add AA with indices $0, 3, 6, 9$, then AA with indices $1, 4, 7, 10$ and so on.

Parameters: an integer λ representing the helix period, the window size l, a Boolean indicating whether the max or the min is taken.

Individuals and Fitness: The fitness is computed by first including the features of the individual to the previous set of six features described above and then by training a decision tree with this extended feature set. The cross-validation of the resulting classifier is computed on the whole dataset and returned. Note that if the same feature appears more than once, the tree building algorithm will use only one.

GA Operators: We chose the classical one-point crossover and point mutation operators but adapted for variable size individuals. For the crossover we simply draw a different crossing point for each mate. For the mutation, we choose randomly for each individual either to add a single gene or to delete an existing one. We used the n-genes, evolutionnary computing environment described in [14] and available on [15].

3.2 Results

New Features. Different runs of our system converge to the same 3 new features. The first two are interpretable: (i) F_1: a Sliding Helix Window which minimizes an hydrophobicity property with $l = 4$ and $\lambda = 4$; (ii) F_2: an Helical window of period 3 which maximizes a turn/coil-propensity. We point out that the periods found are good approximation of the α-helix period (3.6 AA). These two new features are consistent with previous claims about the secondary-structure of the signal-sequence [16].

The third feature, F_3, is a stretch operator which minimizes a property which results from a cluster (see Sect. 3) whose meaning is difficult to understand.

Resulting Tree. The resulting tree (SigTreeGA) is similar to SigTree in its structure and the distribution of the instances. The performance is significantly improved with a cross validation of 93.9%. The value on the specific datasets

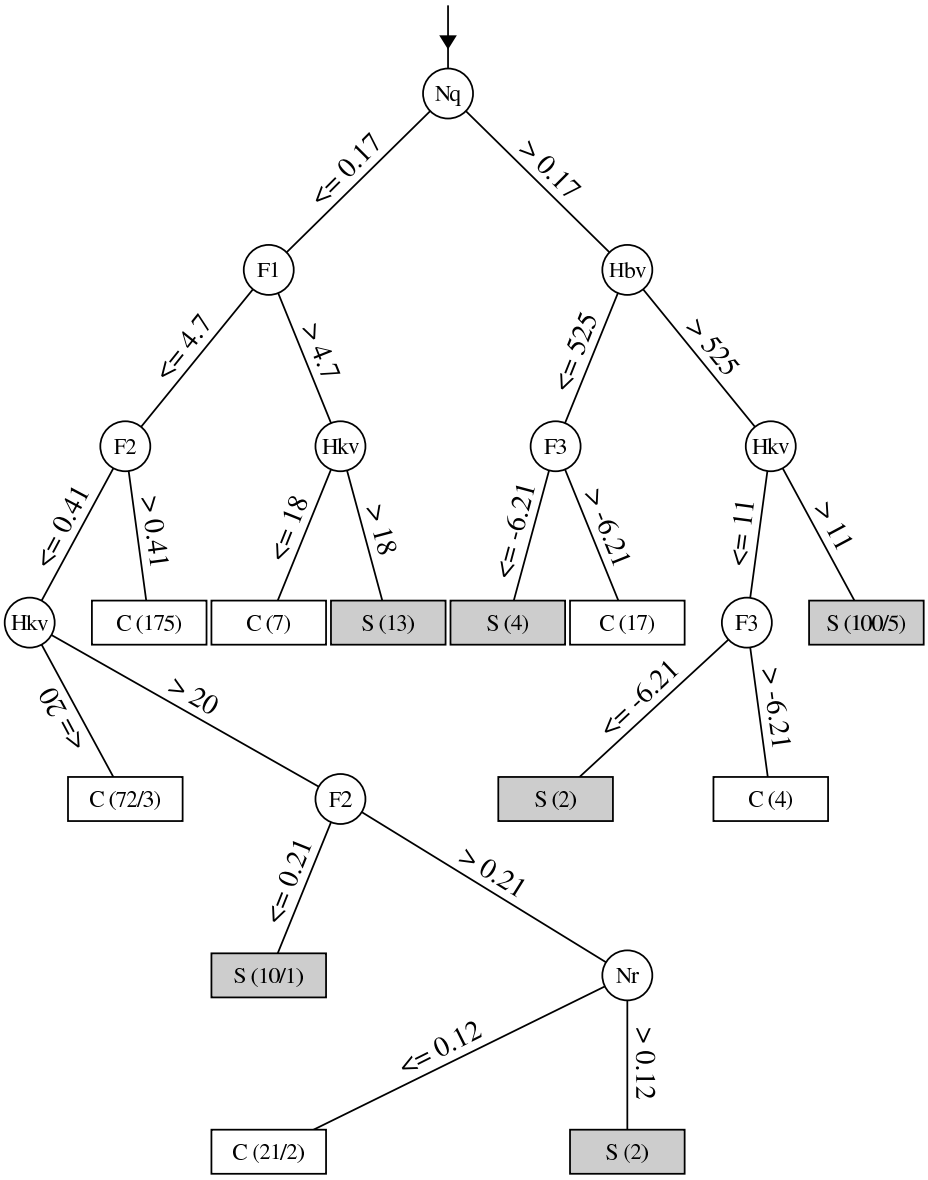

Fig. 2. SigTreeGA. The circle nodes represent the features defined above. The square leaves are the predicted classes. The first number between parenthesis represents the total number of instances classified in the leaves; the second number is the number of dataset sequences wrongly classified (it is omitted if all the sequences are well classified).

are better too. On the entire wild-type protein dataset, we reach an impressive accuracy score of 99.2% and the recognition of mutants is improved. The results are summarized in tab. 1. This predictor is available on [6].

If we analyze now the place of the new features in the tree, we can see that these new features are placed below the first set of operators, allowing to refine the result of the first classifier. The first two new features are placed in the low N_Q branch where the bulk of the non-secreted proteins are classified (271 out of 297). The extra features and their usage in the tree indicate an important role of secondary structure in SS recognition. Not only two of new features are related to helical motives, but also the role of F_2 in the tree can be viewed as eliminating proteins whose propensity of coil/turn formation is to high.

4 Discussion

The first tree produced (SigTree) shows good performances on our datasets. Further, the fact we can omit detection of a valid cleavage site is an indication that our approach is not fooled by another signal which is only relevant to a process taking place after the export.

However, the addition of new features generated by GA allow us to produce a better classifier (SigTreeGA). This improvement justifies our approach and emphasizes the need to provide a more complete theoretical description of SS. Moreover the generated feature F_1 and F_2 give weight to the hypothesis of SS (α-)helical structure.

Most of the mutants in our datasets are not explainable by the established theory. We hope that our new features will eventually reveal how these mutations have changed the export.

In the future we plan to extend our set of reduction functions to model other possible interactions between the translocon and the SS. Furthermore, we will train decision trees on taxonomic groups other than *E. coli*. Since our trees are in fact biological models, the decisions trees of other species/groups can give us insight into the biological differences between their translocons. Finally, this method can be applied to other proteomic challenges, like the prediction of N-terminal acetylation.

References

1. Stryer, L.: Biochemistry. Fourth edn. W. H. Freeman and Company, New-York (1995)
2. von Heijne, G.: The signal peptide. J. Membrane Biology **115** (1990) 195–201
3. Menne, K.M., Hermjakob, H., Apweiler, R.: A comparison of signal sequence prediction methods using a test set of signal peptides. Bioinformatics **16**(8) (2000) 741–2
4. Nielsen, H., Engelbrecht, J., Brunak, S., von Heijne, G.: Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Engineering **10**(1) (1997) 1–6
5. Bendtsen, J.D., Nielsen, H., Widdick, D., Palmer, T., Brunak, S.: Prediction of twin-arginine signal peptides. BMC bioinformatics **6**(167) (2005)
6. Falcone, J.L.: SigTree website. `http://cui.unige.ch/spc/tools/sigtree/` (2007)

7. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. J Mol Biol **157**(1) (1982) 105–32
8. Janin, J., Chothia, C.: Role of hydrophobicity in the binding of coenzymes. appendix. translational and rotational contribution to the free energy of dissociation. Biochemistry **17**(15) (1978) 2943–8
9. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann San Mateo (1993)
10. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
11. von Heijne, G.: A new method for predicting signal sequence clevage site. Nucleic Acid Res. **14** (1986) 4683–4690
12. Kawashima, S., Ogata, H., Kanehisa, M.: Aaindex: amino acid index database. Nucleic Acids Res. (27) (1999) 368–369
13. Falcone, J.L., Albuquerque, P.: Agrégation des propriétés physico-chimiques des acides aminés. IEEE Proc. of CCECE'04 **4** (2004) 1881–1884
14. Falcone, J.L.: Decoding the Signal Sequence. PhD thesis, University of Geneva, Switzerland (2007) (to be published).
15. Fontignie, J., Falcone, J.L.: n-genes website. `http://cui.unige.ch/spc/tools/n-genes/` (2005)
16. Izard, J., Kendall, D.: Signal peptides: exquisitely designed transport promoters. Mol Microbiol. **13**(5) (1994) 765–773