Project INF442-2

Prediction of signal peptide cleavage site using supervised learning

difficulty **

contact: Philippe.Chassignet@polytechnique.edu

1 Introduction

Protein targeting is the mechanism by which some proteins are directed to their appropriate destinations in the cell or outside it. The delivery process is carried out based on information contained in the protein itself. The information is often identified as a sequence of amino acids located near the N-terminal side of the protein. This sequence, known as the *signal peptide*, extends more or less toward the C-terminal side of the protein, up to a position named the *cleavage site*. Once a protein had been properly directed, its N-terminal part is generaly cleaved out at this position, while the C-terminal part leads to the protein as observed in its mature form. Correct targeting of proteins is crucial for life and the identification of signal peptids and their cleavage site is of interest to the design of new drugs.

2 Assignment

We consider the problem of predicting the position of cleavage site in proteins, relying on patterns learned from a training set. Here a protein is known by its *primary sequence*, i.e. the sequence of its amino acids, which is given starting from the N-terminal side. Each amino acid is denoted by a letter code and, as there are 20 standard amino acids, plus some peculiarities, most of upper case letters are used in this encoding.

For instance, the following sequence is the beginning of a protein, where the cleavage site is marked as the underlined letter :

$$\texttt{MASKATLLLAFTLLFATCIA}\underline{\texttt{R}}\texttt{HQQRQQQQNQCQLQNIEALEPIEVIQAEA...}$$

Then, for the purpose of this project, one will simply work on such sequences of letters. An annex web page [1] gives a few sets of many sequences, for which the cleavage position is also given. One has now to program some learning algorithms, tune them, and evaluate their performance. A simple statistical model, will be used first to establish a reference. One will then try to improve accuracy with some specific kernel functions for Support Vector Machines. Guidance follow.

3 Localization of a cleavage site as a classification problem

It has been shown in [2] that the cleavage site may be characterized by amino acids in its close neighborhood. Then, define this neighborhood as the p amino acids before and the q after. Note that the cleavage site is not an amino acid but the bond between two consecutive amino acids. So, values p and q define a neighborhood of $p + q$ amino acids in length. The work in reference suggests up to $p = 13$ and $q = 2$, but different values may also be experimented as meta-parameters of the problem.

For a binary classification problem, given any sequence $S = (a_i)_{i=0,\ldots,\ell-1}$, and any position j , $p \leq j \leq \ell - q$, the word $a_{j-p}a_{j-p+1}\ldots a_{j-1}a_j\ldots a_{j+q-1}$, should then be enough to decide whether bond at position j, between a_{j-1} and a_j, is a cleavage site or not.

3.1 Simple statistical model using a Position Specific Scoring Matrix

Given a set of N sequences, with a known cleavage site for each, one can first compute $N(a,i)$, the number of occurences of each amino acid $a \in \{\mathtt{A},\ldots,\mathtt{Z}\}$, at every position $i \in \{-p,\ldots,q-1\}$, relative to the corresponding cleavage site. Then, for each a and i, define $f(a,i) = N(a,i)/N$, the observed frequency of amino acid a at the relative position i.

In a same way, by counting over the whole length of given sequences, one can compute the observed general frequency $g(a)$ of amino acid a in the given set, regardless of their position. However, it must be noticed that the very first amino acid at the beginning of a sequence is almost always an `M`, because it corresponds to the transcription of the *start* codon. Also, one will not count letters on this first position to avoid a bias.

These frequencies will be used as estimated probabilities to compute the probability of a given word to be located at a cleavage site, under an independent model. We rather use the log of probabilities to go on additive calculations.

Then, $\forall\, a \in \{\mathtt{A},\ldots,\mathtt{Z}\}, \forall\, i \in \{-p,\ldots,q-1\}$, define $W(a,i) = \log(f(a,i)) - \log(g(a))$. Also, as zero counts may occur, pseudocounts [3] must be used. Finally, for any word $w = a_0a_1\ldots a_{p+q-1}$, the score defined as $\sum_{i=-p}^{q-1} W(a_{p+i}, i)$ may tell whether w is the neighborhood of a cleavage site or not. A simple thresholding (to be tuned) is then enough to define a binary classifier.

3.2 Some kernels for SVM

Tutorial [4] is a good introduction. At least, one must implement and experiment some classification algorithms with SVM around the following ideas. It should be easy to achieve better results than with the previous statistical model.

A direct approach for using SVM on fixed length words, consists in encoding each word as a point in a euclidian space. A simple way to encode a single letter is to form a vector of length 26, containing only *zeros*, but a *one* at the i-th position when encoding the i-th letter of the alphabet. Then the encoding of every word of n letters is the vector of length $26.n$, obtained by concatenating the n vectors for the letters of this word. Such an encoding allows a straight linear SVM, or one of the popular polynomials and Gaussian RBF kernels.

One can now observe that the dot product of two vectors, as defined by the previous encoding, simply counts the number of common letters between the two corresponding words. This allows a dedicated implementation, without any vector encoding, with a kernel defined as a similarity function of two strings. Then, one should retrieve exactly the same results, as for the extensive encoding.

For a more flexible criterion, one can use the similarity defined by a substitution matrix [5]. Substitution matrices are built on the observed mutations of amino acids within proteins having a similar function. This yields to scores of similarity $M(a,b)$ between any pair (a,b) of amino acids. The annex web page [1] gives access to many examples of such matrices, with a gradation in the similarity.

Then, to get a score between two words, $a_0\ldots a_{n-1}$ and $b_0\ldots b_{n-1}$, one may uses the sum $\sum_{i=0}^{n-1} M(a_i, b_i)$ of the letters scores. It is a simplified alignment algorithm, as no insertion/deletion is considered. Note that such a score is not related to a distance and that it does not satisfy Mercer's conditions. But it is safe to use it as the dot product to define a RBF kernel, for instance.

4 Expected work

This project consists first of learning from a given set and then predicting whether a given word is located at a cleavage site or not. A program (or many), must allow the choice of a kernel and the tuning of meta-parameters, as the size of the neighborhood around the clivage site, the C parameter of SVM and any other parameter, for example the parameter of a RBF kernel.

One can then slide a window onto a given sequence, examining in turn each position as a potential cleavage site, to locate the real one. A second program must implement your *best* classifier, already tuned, take a series of sequences as an input file, and output the predicted positions of cleavage site (all those classified as positive, possibly more that one for a sequence). It will be evaluated and discussed during the defense, on an entry that is kept secret for now, of course. Execution time is also important.

Références

[1] `http://www.enseignement.polytechnique.fr/profs/informatique/Philippe.Chassignet/18-19/CLEAVAGE/index.html`

[2] Gunnar von Heijne, *A new method for predicting signal sequence cleavage sites*, Nucleic Acids Research, Vol. 14, No. 11, pp. 4683–4690, 1986.

[3] `https://en.wikipedia.org/wiki/Additive_smoothing`

[4] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf and Gunnar Rätsch, *Support Vector Machines and Kernels for Computational Biology*

[5] `https://en.wikipedia.org/wiki/Substitution_matrix`