Detection of Functional Motifs in Biosequences: A Grammatical Inference Approach

Damián López[1], Antonio Cano[1], Manuel Vázquez de Parga[1], Belén Calles[2], José M. Sempere[1], Tomás Pérez[1], José Ruiz[1], and Pedro García[1]

[1] Departamento de Sistemas Informáticos y Computación, Universidad Politénica de Valencia, Camino de Vera s/n, E-46022-Valencia

[2] Instituto Investigaciones Citológicas, Fundación Valenciana de Investigaciones Biomédicas, Amadeo de Saboya 4, E-46010 Valencia

1 Introduction

The selection of proteins with certain characteristics from genomic sequences is a goal of computational biology. One aspect of this problem is to detect certain subsequences (domains or motifs) with some funtional features.

Proteins with *coiled coil* domains are of interest for molecular biologists studying a variety of processes such as protein transport and membrane fusions and the infection of cells by parasites [1, 2].

The coiled coil is a ubiquitous protein folding and assembly motif made of α-helices wrapping around each other forming a supercoil. The sequences of coiled coils are made of seven-residue repeats $(abcdefg)_n$, called heptads, in which generally hydrophobic core occurs at positions a and d. The interaction between two α-helices in a coiled coil involves these hydrophobic residues, as well as the predominantly charged residues at the e and g positions [3]. The result is a highly versatile protein interaction mechanism.

Several programs for predicting coiled coil domains have been described [4, 5] All these programs are based on the probability of appearance of every amino acid in each position of the characteristic heptad, extracted from known coiled coil motifs. In them, a protein sequence is analized using a sliding window of 28 amino acids. For each residue in a test sequence, all these methods output a membership probability to a coiled coil motif.

Recently, Delorenzi and Speed [6] propose a *Hidden Markov Model* (HMM) based approach. In their work, they take profit of the good characterization of the domain to design the net of states of the HMM. The experimentation show good performance.

Nevertheless, the problem of locating general coiled coil motifs is far of being solved. Several authors have noted several important coiled-proteins that are not detected when the previous approaches are used (i.e., [7]).

In our work, we use a grammatical approach to decide whether a protein contains a coiled coil motif or not. As far as we know, this approach to the processing of biosequences has been used only by Yokomori et al. [8] to a preliminary *alpha*-helix detection task.

We tackle the problem of determining if a given protein contains at least a coiled coil motif.

We will use grammatical inference algorithms in order to model the classes involved in the problem. We will consider the whole sequence of the protein instead using the subsequences that contain such motifs. The experimental results obtained show that the performance of our method is comparable to those described above. Whenever convenient data were available, this approach could be extended for detection of other less characterized motifs.

2 Database and methods

All the protein sequences were extracted from SwissProt (release 40, April 2003). All the entries in the database are annotated with the known domains (motifs). We extracted from the database the sequences of those proteins with non-potential annotated coiled coil domains (not obtained by homology). The resulting set of 350 sequences will be referred to from now on as M_c in the sequel. We also randomly extracted a bigger set of 3500 sequences from de database. All the sequences in this set correspond to proteins without references to coiled coil domains. We will refer to this set as M_{nc}. Note that given a domain, the absence of annotations for the domain does not mean that the protein does not contain it, given that it is possible that the protein has not been studied in that way.

The resulting sequences could be seen as strings in an alphabet over 22 symbols (20 amino acids plus the glutamic and aspartic acids). In order to reduce the alphabet as much as possible without loss of information, three different codifications were considered. The first one considered the hidrophobic properties of the amino acids, classifying them in two classes: hidrophobic and polar. The second one extends the first codification considering also charge properties. The third codification is due to Dayhoff and has been used previously in a related work [8]. This codification is based on some physical-chemical properties of the aminoacids (acidity, aromaticity, hidrophobicity, among others). Although the original one considers six classes (from a to f) we have extended it to consider the glutamic and aspartic acids as a new class.

In order to obtain grammatical models for the presence and absence of the domain we used *grammatical inference* (GI) algorithms. These algorithms obtain, from a given set of samples (strings), the decription of a formal language, that is, the description of a potentially infinite set of related strings. GI algorithms could be obtained by considering some desiderable algebraic properties of the words in the resulting language. Briefly, the *k-testable in the strict sense* languages (*k-TSS*) [9] could be obtained in this way by considering segments of certain length. The inference algorithm for k-TSS languages has been used successfully in several pattern recognition tasks (i.e. [10]).

Another approach to grammatical inference is based on the consideration of some features of the training set. One example of this approach is the *Error-Correcting Grammatical Inference* algorithm (ECGI) by Rulot [11]. Briefly, the algorithm uses a distance measure in order to extend the initial (empty) language. The resulting language is not characterizable, nevertheless, it accepts many closely related words to those in the training set. This algorithm has also been succesfully used in pattern recognition tasks (i.e., [12]).

In this work, we used both algorithms to infer both the class of proteins that contains a coiled coil motif and the class of proteins that does not contain such motifs. Once the classes were obtained, we used the Viterbi algorithm (i.e , [13]) in order to obtain, for a given protein sequence, the probability of belonging to each model.

3 Discussion

The same process is applied for each codification of the database. We extracted a training set from M_c and M_{nc} and inferred a language using these sets (namely TR_c and TR_{nc}) and both algorithms exposed above. Test sets (TS_c and TS_{nc}) were extracted from the database, and membership probabilities were obtained by using Viterbi's algorithm. The proteins were classified using a maximum probability criterion.

Each experiment involved the set M_c and one subset of M_{nc} of 350 samples. Therefore, we carried out ten rounds with disjoint sets of non-coiled proteins. In order to obtain as much statistical relevance of the results as possible, seven different balanced partitions of the data were considered in each round. The influence of the parameter k was also studied.

In order to compare our results with the most known prediction algorithms [4, 5], it is very important to note that the database contains annotations only for those proteins that contain a coiled coil region. Our approach looks for modelling the presence and absence of the coiled coil domain. Therefore, we have an important drawback because of the lack of non-coiled annotations in the database. Thus, we think that the importance of the error rate has to be somewhat weighted. The error rate when classifying coiled coil proteins should to be considered as *true* error, but the error obtained when classifying non-coiled coil proteins should be considered as a partially valid measure.

We run available versions of the algorithms ([14, 15] respectively). We fixed a threshold in 0.5 and consider every protein that contains any amino acid with a probability over that threshold as a coiled one.

	error rate			
	Lupas alg.	Berger alg.	k-TSS	ECGI
M_c	18% (63/350)	23.14% (81/350)	11.8%	15.5%
M_{nc}	11.14% (390/3500)	0.3% (11/3500)	20.2%	37.8%

Fig. 1. Comparison between algorithms. The k-TSS languages were inferred with $k = 12$ and the codification of four symbols. The ECGI approach considered the two symbols alphabet.

The best results were obtained when the k-TSS inference algorithm was used. The lower value of the k parameter, the worse results obtained. This behaviour could be explained by the main known feature of the coiled-coil domain (sequence of heptads).

The ECGI algorithm obtained worse results than the k-TSS inference algorithm. The ECGI algorithm is based on an error-correcting analysis, therefore, it is posible that the error-correcting phase leaded to align coiled subsequences with other non-coiled ones.

Figure 1 show the results of the experimentation. Notice that the k-TSS inference algorithm lead to best classification of the M_c set, but with worse behaviour when the set M_{nc} is considered. ECGI algorithm produces worse results in any case.

Taking into account the above discussion, we consider the results very promising and comparable to those obtained by traditional approaches. Of course, the availability of non-coiled protein sequences should improve our results.

Coiled coil is a well characterized motif. Its structure is the key stone of the most used prediction algorithms [4–6]. This work permits us to conjecture that other motifs, whose structure is poorly known, could be detected by using grammatical inference techniques.

References

1. Skehel, J.J., Wiley, D.C.: Coiled coils in both intracellular vesicle and viral membrane fusion. Cell **95** (1998) 871–874
2. Chan, D.C., Kim, P.S.: HIV entry and its inhibition. Cell **93** (1998) 681–684
3. O'Shea, E.K., Klemm, J.D , Kim, P.S., Alber, T.: X-ray structure of the GCN4 leucine zipper, a two-stranded, parallel coiled coil. Science **254** (1991) 539–544
4. Lupas, A., Dyke, M.V., Stock, J.: Predicting coiled coild from protein sequences. Science **252** (1991) 1162–1164
5. Berger, B , Wilson, D.B., Wolf, E., Tonchev, T., Milla, M., Kim, P.S.: Predicting coiled coils by use of pairwise residue correlation. Proc. Natl. Acac. Sci. **92** (1995) 8259–8263
6. Delorenzi, M., Speed, T.: An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. Bioinformatics **18** (2002) 617–625
7. Singh, M., Berger, B., Kim, P.: LearnCoil-VMF: Computational evidence for coiled-coil-like motifs in many viral membrane fusion proteins. J. Mol. Biol. **290** (1999) 1031–1041
8. Yokomori, T., Ishida, N., Kobayashi, S.: Learning local languages and its application to protein α-chain identification. In: Proc. Twenty-Seventh Annual Hawaii International Conference on System Sciences, IEEE (1994) 113–122
9. García, P., Vidal, E.: Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. IEEE Trans. Pattern Analysis and Machine Intelligence (1990) 920–925
10. Torres, I., Varona, A.: k-TSS language models in speech recognition. Computational Speech and Language **15** (2001) 127–149
11. Rulot, H.: ECGI, un algoritmo de inferencia gramatical mediante corrección de errores. PhD thesis, Depto. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (1992)
12. Prieto, N., Sanchis, E., Palmero, L.: Continuous speech understanding based on automatic learning of acoustic and semantic models. In: Proceedings of the 1994 International Conference on Spoken Language Processing, ICSLP (1994)
13. Amengual, J.C., Sanchis, A., Vidal, E., Benedí, J.: Language simplification through error-correcting and grammatical inference techniques. Machine Leaning **44** (2001) 143–159
14. Source Code NCOILS. (`http://www.russell.embl.de/cgi-bin/coils-svr.pl`)
15. Source Code PAIRCOIL. (`http://theory.lcs.mit.edu/~bab/computing`)