Protein Motif Prediction by Grammatical Inference[⋆]

Piedachu Peris, Damián López, Marcelino Campos, and José M. Sempere

Departamento de Sistemas Informáticos y Computación.
Universidad Politécnica de Valencia.
Camino de Vera s/n
46071 Valencia (SPAIN)
{pperis, dlopez, mcampos, jsempere}@dsic.upv.es

Abstract. The rapid growth of protein sequence databases is exceeding the capacity of biochemically and structurally characterizing new proteins. Therefore, it is very important the development of tools to locate, within protein sequences, those subsequences with an associated function or specific feature. In our work, we propose a method to predict one of those functional motifs (coiled coil), related with protein interaction. Our approach uses even linear languages inference to obtain a transductor which will be used to label unknown sequences. The experiments carried out show that our method outperforms the results of previous approaches.

Keywords: Grammatical inference, bioinformatics, protein motif location.

1 Introduction

Processing of biological data is a key task in many applied fields. Recently, an explosion of papers apply Pattern Recognition techniques to bioinformatics tasks [1,2]. Formal Language Theory and Grammatical Inference (GI) are also playing an important role and it is expected that they could lead to good applied results [3,4]. Some works use GI techniques in order to address, among other tasks: secondary structure identification [5], protein motifs detection [6,7,8], optimal consensus sequence discovery [9,10] or gene prediction [11].

The selection of proteins with certain characteristics from genomic sequences is a central goal of computational biology. One aspect of this problem is to detect certain subsequences, known as domains or motifs, with some interesting functional features.

Coiled coil domains are of interest for molecular biologists studying a variety of processes such as protein transportation and interaction. It has been shown that coiled coil motif is implied in membrane fusion and the infection of cells by

[⋆] Work supported by the CICYT TIC2000-1153 and the Generalitat Valenciana GV06/068.

Y. Sakakibara et al. (Eds.): ICGI 2006, LNAI 4201, pp. 175–187, 2006.

viruses or parasites [12][13]. Predictions based on analysis of primary sequences suggest that approximately 2-3% of all protein residues form coiled coils [14].

The coiled coil motif consist of two α-helices wrapping around each other forming a supercoil. The sequences of coiled coils are made of seven-residue (amino acids) repeats which forms a pattern usually denoted $(abcdefg)_n$ where the position of each residue is noted from a to g. Within this pattern, called also heptad, generally an hydrophobic core occurs every four and then three residues apart, that is, at positions a and d. The interaction between two α-helices in a coiled coil involves these hydrophobic residues. The result is a highly versatile protein interaction mechanism (see Figure 1). Due to its simplicity and regularity, the coiled coil is one of the most extensively studied protein motifs.

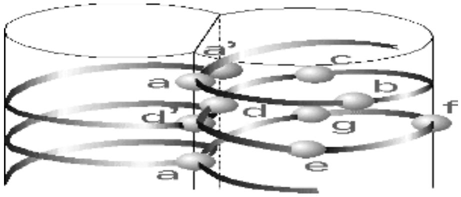

Fig. 1. A schematic coiled coil representation is shown. The relative position of amino acids in a characteristic coiled coil heptad repeat is marked with bullets. Residues at the a and d positions are predominantly hydrophobic. Due to the α-helical structure, residues at the a and d position are spatially close one each other. Both features (hydrophobicity and spatial arrangement), provide a versatile protein interaction mechanism.

Several programs for predicting coiled coil domains have been proposed. The most relevant to large-scale annotations are *coils* [15] (probably the most widely used), *paircoil* [16] and *multicoil* [14]. 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. Multicoil is the most specialized one, and aims to detect double or triple coiled coil domains. All of them are based on a *Position Specific Scoring Matrix* (PSSM) (also known as *Position Weighted Matrix*) approach [17]. This general scheme considers the probabilities of appearance of each possible residue in each position of the motif. These probabilities are obtained from sequences with confirmed motifs or considering multiple sequence alignments of functionally related sequences. This approach has also been widely used in gene-finding tasks.

The work by Lupas et al. [15] takes into account that even very short proteins have stable coiled coils containing four or five heptads, and analyzes the test sequences using a sliding window of 28 amino acids. A score for each amino acid in the sequence of the protein is obtained using the probabilities of the PSSM. Then, the score distributions for general globular proteins and coiled coil sequences are approximated with Gaussian curves used to obtain, for each amino acid of the protein, a probability of belonging to a coiled coil motif.

Although this approach is widely known by the biological community, it is known that the method leads to a significant number of *false positives*, some of them due to the continuous appearance of some frequent amino acids in coiled coil regions (for instance, $(Lys - Lys - Lys)_n$ scores highly though it is not a coiled coil). To solve this problem, Berger et al. [16] follow the same PSSM approach but taking into account the pairwise amino acid correlations in known coiled coils. The correlations and the size of the window used were empirically selected, and,

- the correlations between the pairs of amino acids placed in positions $(i, i+1)$ and $(i, i+4)$ were considered.
- the size of the sliding window was set to 30.

The authors claim that the approach is useful to discard false positives detected by the Lupas' approach. They carry out a wide experimentation to show the behaviour and present several examples of false positives detected.

Nevertheless, the problem of locating general coiled coil motifs still remains open. Several authors have noted several important coiled-proteins that are not detected when the previous methods are used (among others, fusion-membrane proteins of the human and simian inmunodeficiency virus or Ebola virus [18]). Thus, several other works propose solutions for more specific instances of the problem [19][18].

In our work, we use a grammatical approach to locate coiled coil motifs within protein sequences. Previous related works address the task of detecting protein structures: α-helix structures in protein sequences [20] or even the coiled coil motif [7,8].

We address the problem of predicting the coiled coil motifs of a given protein. Our approach considers the original sequence and an annotated reduced version which distinguish between coiled coil and non-coiled coil subsequences. This data is combined into an even linear structure and used to infer *Even Linear Languages* (ELL). The inferred languages are then used to build a transductor suitable to translate, that is, to distinguish coiled and non-coiled regions in problem sequences. The results of the experimentation carried out are compared with other existing approaches. Our work is organized as follows: Section 2 summarizes some definitions and the notation used; Section 3 explains our approach to the problem; Section 4 shows the experimental results and the indexes used to compare our results with previous ones; Finally, some conclusions and future lines of research end the paper.

2 Notation and Definitions

Let Σ be an alphabet and Σ^* the set of words over the alphabet. For any word $x \in \Sigma^*$ let x_i denote the i-th symbol of the sequence, let $|x|$ denote the length of the word and let x^r denote the reverse of x. Let also λ denote the empty word. A grammar is denoted by $G = (N, \Sigma, P, S)$ where N and Σ are the auxiliar and terminal alphabets, P is the set of productions and $S \in N$ is the initial symbol or axiom. The language generated by G is denoted by $L(G)$.

An *Even Linear Grammar (ELG)* is a context-free grammar [21] where the productions are of the forms:

$A \to xBy$ where $A, B \in N$, $x, y \in \Sigma^*$ and $|x| = |y|$
$A \to x$ where $A \in N$, $x \in \Sigma^*$

The class of Even linear Languages (ELL) is a subclass of the context free languages and includes properly the class of regular languages. Given an ELG, it is possible to obtain an equivalent one where the productions are of the form.

$A \to aBb$ where $A, B \in N$, $a, b \in \Sigma$
$A \to a$ where $A \in N$, $a \in \Sigma \cup \{\lambda\}$

The learning of ELL can be reduced to the inference of regular languages [22]. The general algorithm consists in transforming the training strings through a function $\sigma : \Sigma^* \to [\Sigma \times \Sigma]^* \cup [\Sigma]^*$ defined as follows:

$\sigma(\lambda) = \lambda$
$\sigma(a) = [a]$ where $a \in \Sigma$
$\sigma(axb) = [ab]\sigma(x)$ where $a, b \in \Sigma$ and $x \in \Sigma^*$

Once applied the function σ, it is possible to use any regular language inference algorithm to learn a language over the alphabet $[\Sigma \times \Sigma]^* \cup [\Sigma]^*$ and then transform the productions of the obtained regular grammar to undo the transformation σ as follows:

$\forall A \to [ab]B \in P$ add the production $A \to aBb$ to the ELG
$\forall A \to [a] \in P$ add the production $A \to a$ to the ELG
$\forall A \to \lambda \in P$ add all these productions to the ELG

Obviously, whenever the GI algorithm identifies a subclass of regular languages, then a subclass of ELL is obtained.

A *finite state transducer* is defined by a system $\tau = (Q, \Sigma, \Delta, q_0, Q_F, E)$ where: Q is a set of states, Σ and Δ are respectively the input and output alphabets, q_0 is the initial state, $Q_F \subseteq Q$ is the set of final states and $E \subseteq (Q \times \Sigma^* \times \Delta^* \times Q)$ is the set of transitions of the transducer. A successful path in a transducer is a sequence of transitions $(q_0, x_1, y_1, q_1), (q_1, x_2, y_2, q_2), \ldots, (q_{n-1}, x_n, y_n, q_n)$ where $q_n \in Q_F$ and for $1 \leq i \leq n$: $q_i \in Q$, $x_i \in \Sigma^*$ and $y_i \in \Delta^*$. Note that a path can be denoted as $(q_0, x_1x_2 \ldots x_n, y_1y_2 \ldots y_n, q_n)$ whenever the sequence of states are not of particular concern. A transduction is defined as a function $t : \Sigma^* \to \Delta^*$ where $t(x) = y$ if and only if there exist a successful path (q_0, x, y, q_n). We refer the interested reader to [23].

3 Grammatical Inference Approach to Coiled Coil Prediction

Several methods have been proposed to solve the coiled coil motif location task. The most widely known are the PSSM-based methods by Lupas and Berger

[15,16], but also Hidden Markov Models have been used [24] as well as Neural Networks approaches [25]. This motif occurs always on an underlying α-helix protein structure. It is important to note, on the one hand, that the detection of α-helix structure has been successfully addressed by GI methods [20], and in the other hand, the biological regularity of the coiled coil pattern (that is, the characteristic repeated heptad). This two facts support our GI approach to tackle this task.

In our work we address the protein motif location problem as a transduction problem. In such a way that, given an amino acid sequence, we propose a method to obtain a sequence with the same length which distinguishes between those amino acids within a motif and those that are not. The inference of transducers has been widely studied by the GI community, in our work, we take into account the special features of our problem to propose a method based on inference of ELL. Our approach firstly transforms the available data to obtain a training set with even linear structure. This set was used to infer an ELL. The transducer is obtained using the structure of the ELG inferred. To do so, note that, given a ELG $G = (N, \Sigma, P, S)$ that does not contain productions of the form $A \rightarrow a$, $a \in \Sigma$, it is possible to obtain a transducer $\tau = (N, \Sigma, \Sigma, S, Q_F, E)$ where:

$$\begin{aligned} Q_F &= \{A \in N \ : \ (A \rightarrow \lambda) \in P\} \\ E &= \{(A, a, b, B) \ : \ (A \rightarrow aBb) \in P\} \end{aligned}$$

Example 1 shows how this transformation work.

Example 1 *Given the ELG $G = (N, \Sigma, P, S)$ with the productions:*

$$\begin{aligned} S &\rightarrow aS0 \mid bB1 \\ A &\rightarrow aA1 \mid bS0 \\ B &\rightarrow aA1 \mid bB1 \mid \lambda \end{aligned}$$

then, the transducer $\tau = (N, \Sigma, \Sigma, S, \{B\}, E)$ is obtained where:

$$E = \left\{ \begin{array}{l} (S, a, 0, S),\ (S, b, 1, B),\ (A, a, 1, A), \\ (A, b, 0, S),\ (B, a, 1, A),\ (B, b, 1, B) \end{array} \right\}$$

The resulting transducer is shown in Figure 2.

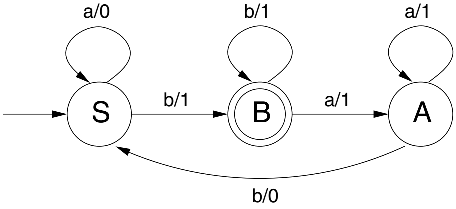

Fig. 2. A three states transducer example. A label x/y denotes that the transition symbol is x with output y. For instance, the transduction of *baabaab* is 1110001

□

As we stated before, the learning problem for ELL can be reduced to the problem of learning regular languages. In our work, in order to learn the ELL, we use an algorithm to infer *k-testable in the strict sense* (k-TSS) languages [26,27,5]. The class of k-TSS languages is contained into the regular languages class and it is characterized by the set of segments of length k that appear in the words of the language. The characteristic coiled coil heptad lead us to consider this algorithm as a first suitable candidate.

Our approach considered a set of protein sequences P with known coiled coil motifs and another set L of strings over $\Delta = \{c, n\}$ with, a labeled sequence l_x for each sequence x in P. The labeled sequence was obtained in such a way that distinguish between the amino acids corresponding to coiled and non-coiled regions. That is, given the string $x = x_1 x_2 \ldots x_n \in P$ and its corresponding labeled string $l_x = l_1 l_2 \ldots l_n \in L$, $l_i = c$ whenever x_i correspond to a coiled coil motif, otherwise $l_i = n$.

These sets were combined to obtain another set M with the strings xl_x^r. Note that the strings in this set have an even linear structure and even length. The set M was used to obtain a transducer by ELL inference. The general method is summarized in Algorithm 3.1.

Algorithm 3.1. Coiled coil Grammatical Inference approach.

```
Input:
    – A set $P$ of amino acid sequences with known coiled coil motifs.
    – A set $L$ of motif labeled sequences. Each string $x$ in $P$ has its corresponding
      string $l_x$ in $L$.
Output:
    – A transducer to locate coiled coil motifs.
Method:
    – Combine the sets $P$ and $L$ to obtain the training set $M$ with strings $xl_x^r$
    – Apply to the strings in $M$ the transformation function $\sigma$
    – Apply a GI algorithm for (a subclass of) regular languages
    – Undo the transformation $\sigma$ to obtain the ELG from the regular language
    – Return the transducer obtained from the ELG
EndMethod.
```

The returned transducer can be used to analyze problem sequences to obtain the corresponding transduction. Note that the transducer may be non-deterministic and the test sequences may not belong to the language accepted by the transducer. Therefore, an error-correcting parser (for instance Viterbi's algorithm) is necessary to analyze the test sequences. We used a standard configuration of Viterbi's algorithm when a GI approach is applied to pattern recognition tasks (i.e. [8]). We considered the number of times each transition of the transducer is used to probabilize it. The error-correcting analysis considered only low probability substitution errors for edit operations.

4 Experimental Results

In order to test our approach we considered two different datasets: The first one contains sequences extracted from SwissProt Database (release 40, April 2003) [28]. All the sequences selected contain a non-potential coiled coil annotation. Potential annotations are those based mainly on homology results. Potential motifs were not included in the database because the function of potential domains has not been yet assured. The resulting 350 sequences database has been previously used by [7,8]. The second coiled coil dataset was build by Delorenzi and Speed [24]. This dataset considered sequences of *Protein Data Bank* [29]. This database contains structural information of the tertiary structure of the proteins and it is more suitable to obtain confident information. From the information stored in the database, two sets were built: one with those coiled coil sequences with experimental confirmation (397 sequences), and another with sequences from which coiled coils motifs were eliminated (1525 sequences).

Protein sequences can be considered as strings in an 22 symbols alphabet (20 amino acids plus the glutamic and aspartic acids[1]). In order to reduce the alphabet size without loss of information, two different codifications were considered. The first one is due to Dayhoff and is based on some properties of the amino acids. This codification has been previously used in some GI papers [20,7,8]. Second codification used considers only two symbols which distinguish between hydrophobic and polar amino acids. This codification was used because this feature is key in the coiled coil motif. Figure shows the correspondence of each amino acid for both codifications.

amino acid	P/H	Dayhoff
C	p	a
R, H, K,	p	d
D, E	p	c
N, Q	p	c
B, Z	p	g
Y	p	f
G	p	b
S, T	p	b
A, P	h	b
F, W	h	f
L, V, M, I	h	e

Fig. 3. Amino acid codifications

Several measures are suitable to evaluate the results. Some of them are reviewed in [30] under a scope of gene-finding problems. Nevertheless, they are

[1] Some sequences also contain the symbol X. This happens whenever it is not clear which amino acids occupy a certain position. In this work, we did not consider such sequences (just one sequences in the first dataset and two sequences in the second).

suitable to be applied to motif location tasks. Among all the measures proposed, *Sensitivity* and *Specificity* are probably the most used. Sensitivity (Sn) measures the probability of predict those symbols inside a motif. Specificity (Sp) measures the probability of predicted segments to be actually motifs.

These measures are computed using the following partial results:

True Positives (TP): symbols of the sequence inside a motif that are correctly annotated.

True Negatives (TN): symbols of the sequence outside a motif that are correctly annotated.

False Positives (FP): symbols of the sequence outside a motif that are annotated as they were inside one.

False Negatives (FN): symbols of the sequence inside a motif that are not correctly annotated.

Using these measures, both Sn and Sp can be computed as follows:

$$Sn = \frac{TP}{TP + FN} \qquad Sp = \frac{TP}{TP + FP}$$

Note that neither Sn nor Sp alone constitute a good measure. The *Correlation Coefficient* (CC) is defined in order to use a single value that summarizes both results. It can be computed as follows:

$$CC = \frac{(TP \cdot TN) - (FN \cdot FP)}{\sqrt{(TP + FN) \cdot (TN + FP) \cdot (TP + FP) \cdot (TN + FN)}}$$

Although CC has some statistical properties [30] it has also an undesirable drawback. It is not defined when any factor of the root is zero. Some measures have been defined to overcome this inconvenient, we will use the *Approximate Correlation* (AC) which is defined as follows:

$$AC = \left\{ \frac{1}{4} \left[\frac{TP}{TP + FN} + \frac{TP}{TP + FP} + \frac{TN}{TN + FP} + \frac{TN}{TN + FN} \right] - 0.5 \right\} \cdot 2$$

In order to evaluate the results, it has to be noted that, for some samples, it was not possible to calculate Sp and CC, and therefore, these samples were not taken into account. The Approximate Correlation AC considers all samples, including those for which it was not possible to calculate CC or Sp. This can explain why in some cases the difference between AC and CC is relevant. This fact makes AC the most reliable measure in order to evaluate the global performance of our approach.

In order to test our approach, both datasets were processed using the same scheme. We considered several values of the GI algorithm k parameter, and performed a leaving-one-out experiment (all the sequences but one were used to infer the transducer and the remaining one to test the performance. This process is iterated to consider the whole dataset).

Our results are also compared with the output of coils and paircoil methods. Note that these methods are based on the physic-chemical properties of the coiled coil motif, therefore, no training is needed. Public versions of these programs are available at [31] and [32] respectively. Both approaches use a default probability threshold of 0.5. This threshold has to be reached to consider an amino acid as belonging to a coiled coil motif. Note that to lower this threshold implies an increasing of the sensibility and a decreasing of the specificity levels. In the same way, to higher the default parameter implies an increasing of the sensibility but a decreasing of the sensibility levels. We are interested in the general behaviour, therefore, we will consider the default threshold value.

The two symbols codification did not obtain significant results, thus, we will show only the best configuration performance for this codification. The results obtained by the subset of SwissProt database are shown in Table 1.

Table 1. Experimental results when the coiled subset of SwissProt was used. Note the improvement of the results obtained by our method. Although bigger k parameter values lead to higher sensitivity, best results are obtained by using $k = 8$.

Method		**Sn**	**Sp**	**CC**	**AC**
coils		0.4568	0.8022	0.4897	0.4155
paircoil		0.4996	0.8209	0.5676	0.4806
IGcoils (Dayhoff coding)	$k = 2$	0.7865	0.7226	0.6355	0.5480
	$k = 3$	0.8287	0.7610	0.6799	0.6365
	$k = 4$	0.8095	0.8491	0.7547	0.7164
	$k = 5$	0.7688	0.9563	0.8741	0.7728
	$k = 6$	0.8527	0.9804	0.9291	0.8638
	$k = 7$	0.9180	0.9701	0.9420	0.9085
	$k = 8$	0.9506	0.9673	0.9529	0.9338
	$k = 9$	0.9696	0.9614	0.9498	0.9428
	$k = 10$	0.9710	0.9624	0.9479	0.9457
IGcoils (P/H coding)	$k = 8$	0.6526	0.7887	0.6113	0.5174

The experimental results when the Delorenzi database was used are shown in Table 2. Note that in this experiment, the lower values of the inference parameter, the worse values of the sensitivity and specificity. Nevertheless, considering the correlation coefficient (or the approximate correlation as well), our approach improves the results.

One of the most important drawbacks of the Lupas' method is the number of false positives that it produces. Berger et al. considered this fact as their main motivation to develop their approach. In order to compare the performance of our method when non-coiled sequences are to be tested, we carried out the following experiment: two transducers were inferred, each one considering all the sequences of the two different datasets (i.e. the coiled coil SwissProt subset and the Delorenzi dataset). The sequences of the non-coiled dataset where

Table 2. Experimental results when the coiled Delorenzi's database was used. Best results were obtained for $k = 7$ and $k = 8$. This result is consistent with the heptad-based biological characterization of the motif.

Method		**Sn**	**Sp**	**CC**	**AC**
coils		0.6688	0.8552	0.6372	0.6222
paircoil		0.6511	0.8489	0.6693	0.5972
IGcoils (Dayhoff coding)	$k = 2$	0.6269	0.7420	0.6501	0.5058
	$k = 3$	0.6353	0.7616	0.6730	0.5342
	$k = 4$	0.6275	0.7778	0.6793	0.5654
	$k = 5$	0.5709	0.8249	0.6842	0.5729
	$k = 6$	0.5262	0.8730	0.7395	0.5692
	$k = 7$	0.5465	0.9212	0.8128	0.5952
	$k = 8$	0.6058	0.9002	0.8036	0.6356
IGcoils (P/H coding)	$k = 8$	0.4012	0.8001	0.6020	0.3883

Table 3. Experimental results when the non-coiled dataset was processed. Upper two rows show the percentage of symbols predicted inside a coiled coil motif (error rate). Lower rows show the number of sequences in the non-coiled dataset with any erroneous prediction. All this results were obtained with the Dayhoff coding.

		IGcoils			**Coils**	**Paircoil**
		$k = 6$	$k = 7$	$k = 8$		
% error rate	SwissProt dataset	0.0118	0.0175	0.0254	0.0058	0.0023
	Delorenzi dataset	0.0036	0.0030	0.0016		
# of erroneous sequences	SwissProt dataset	104	123	140	57	12
	Delorenzi dataset	26	18	12		

then independently tested considering this two transducers, and two measures were work out: the error percentage and the number of sequences with a motif predicted. Note that these sequences were processed to delete all the coiled coil motifs, therefore, all the coiled coil predictions are erroneous. The results obtained with the two symbols codification were not significant, therefore, we only show the results (summarized in table 3) obtained when the Dayhoff coding was used.

The results obtained by transducers inferred with the SwissProt dataset are not comparable to previous methods. Nevertheless, those transducers inferred using the Delorenzi dataset obtained better results than Lupas' and Berger's methods. It could be argued that some homology between the Delorenzi's coiled coil and non-coiled datasets somewhat biases the results, but this can be refused by noting that the datasets were built considering low homology between the sequences and that there are many more sequences in the non-coiled dataset than in the coiled one.

5 Conclusions and Future Work

This work addresses the task of protein motif prediction by applying GI techniques. Previous methods are based on the physical-chemical characterization of the motif. This allow to use a Position Weighted Matrices approach to predict new motifs. The results obtained lead us to conjecture that it is not necessary to biologically characterize a new motif in order to develop prediction tools. It is also to note that these results are feasible to be extended to other bioinformatic tasks.

In all cases, the results obtained with the Dayhoff coding were much better than those obtained with the two symbols codification. Therefore, in what follows we will refer only to Dayhoff codification.

The results obtained using the first corpus (coiled sequences from SwissProt) show that our method outperforms previous approaches. Nevertheless this results are somewhat misleading because the transducers inferred lead to a high number of false positives (both error rate and number of sequences with erroneous predictions) when non-coiled sequences are tested.

When the Delorenzi's dataset was considered, our approach gave better results to those obtained by previous methods. This is mainly due to an increase of the specificity levels. This fact is specially motivating in order to apply new prediction methods, because it is very important to reduce the number of false positives. The experiments involving non-coiled sequences confirmed the good performance of our approach, which obtains lower error rate with the same number of erroneous sequences.

Future lines of work should consider, the consideration of other inference algorithms. Specially interesting are the learning of synchronized and non-synchronized ELL [33]. Bigger datasets (modeling coiled coil motif or other biologically interesting motifs) should also be considered. The comparison between GI and NN or HMM approaches to protein motif location is left also as future work.

References

1. Editorial. The fundamental role of pattern recognition for gene-expresion/microarray data in bioinformatics. *Pattern Recognition*, 38:2226–2228, 2005.
2. A.W-C. Liew, H. Yan, and M. Yang. Pattern recognition techniques for the emerging field of bioinformatics: A review. *Pattern Recognition*, 38:2055–2073, 2005.
3. D.B. Searls. The language of genes. *Nature*, 420:211–217, 2002.
4. Y. Sakakibara. Grammatical inference in bioinformatics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1051–1062, 2005.
5. T. Yokomori and S. Kobayashi. Learning local languages and their application to dna sequence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1067–1079, 1998.
6. S. Arikawa, S. Kuhara, S. Miyano, A. Shinohara, and T. Shinohara. A learning algorithm for elementary formal systems and its experiments on identification of transmembrane domains. In *Proceedings of the 25th Hawaii Intl. Conf. on System Sciences*. IEEE, 1992. ISBN: 0-8186-2420-0.

7. D. Lopez, A. Cano, M. Vazquez de Parga, B. Calles, J.M. Sempere, T. Perez, J. Ruiz, and P. Garcia. Detection of functional motifs in biosequences: A grammatical inference approach. In *Proceedings of the 5th Annual Spanish Bioinformatics Conference*, pages 72–75. Univ. Politècnica de Catalunya, 2004. ISBN: 84-7653-863-4.
8. D. López, A. Cano, M. Vázquez de Parga, B. Calles, J. M. Sempere, T. Pérez, M. Campos, J. Ruiz, and P. García. Motif discovery by k-tss grammatical inference. In G. Paliouras C. de la Higuera, T. Oates and M. Van Zaanen, editors, *IJCAI-05 Workshop on Grammatical Inference Applications: Successes and Future Challenges*, 2005. Working Notes.
9. A. Brazma, I. Johansen, J. Vilo, and E. Ukkonen. Pattern discovery in biosequences. *LNAI*, 1433:257–270, 1998. 4th Intl. Colloquium, ICGI'98.
10. H. Arimura, A. Wataki, R. Fujino, and S. Arikawa. A fast algorithm for discovery optimal string patterns in large databases. *LNAI*, 1501:247–261, 1998. 9th Intl. Conference, ALT'98.
11. P. Peris, D. López, M. Campos, and J.M. Sempere. Gene-finding by grammatical inference. *(submitted manuscript)*.
12. J.J. Skehel and D.C. Wiley. Coiled coils in both intracellular vesicle and viral membrane fusion. *Cell*, 95:871–874, 1998.
13. D.C. Chan and P.S. Kim. Hiv entry and its inhibition. *Cell*, 93:681–684, 1998.
14. E. Wolf, P.S. Kim, and B. Berger. Multicoil: a program for predicting two- and three-stranded coiled coils. *Protein Science*, 6:1179–1189, 1997.
15. A. Lupas, M. Van Dyke, and J. Stock. Predicting coiled coild from protein sequences. *Science*, 252:1162–1164, 1991.
16. B. Berger, D.B. Wilson, E. Wolf, T. Tonchev, M. Milla, and P. S. Kim. Predicting coiled coils by use of pairwise residue correlation. *Proc. Natl. Acad. Sci.*, 92:8259–8263, 1995.
17. C. Mathé, M.F. Sagot, T. Schiex, and P. Rouzé. Current methods of gene prediction, their strengths and weakenesses. *Nucleic Acid Research*, 30(19):4103–4117, 2002.
18. M. Singh, B. Berger, and P.S. Kim. Learncoil-vmf: Computational evidence for coiled-coil-like motifs in many viral membrane fusion proteins. *J. Mol. Biol.*, 290:1031–1041, 1999.
19. M. Singh, B. Berger, P.S. Kim, J.M. Berger, and A.G. Cochran. Computational learning reveals coiled coil-like motifs in histidine kinase linker domains. *Proc. Natl. Acad. Sci.*, 95:2738–2743, 1998.
20. T. Yokomori, N. Ishida, and S. Kobayashi. Learning local languages and its application to protein α-chain identification. In *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*, pages 113–122. IEEE, 1994.
21. J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publishing Company, 1979.
22. J.M. Sempere and P. García. A characterization of even linear languages and its application to the learning problem. *LNAI*, 862:38–44, 1994.
23. J. Berstel. *Transductions and context-free languages*. Teubner Studienbücher, 1979.
24. M. Delorenzi and T. Speed. An hmm model for coiled-coil domains and a comparison with pssm-based predictions. *Bioinformatics*, 18(4):617–625, 2002.
25. M. Campos and D. López. Neural network approach to locate motifs in biosequences. 3773:214–221, 2005. 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005.

26. T. Knuutila. *Advances in Structural and Syntactic Pattern Recognition: Proc. of the International Workshop*, chapter Inference of k-Testable Tree Languages, pages 109–120. World Scientific, 1992.
27. P. García. Learning k-testable tree sets from positive data. Technical Report DSIC/II/46/1993, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, 1993. Available on: `http://www.dsic.upv.es/users/tlcc/tlcc.html`.
28. Swiss-Prot groups at SIB and at EBI. Uniprot database (swissprot and trembl). http://www.expasy.ch/sprot/.
29. Protein data bank. http://www.rcsb.org/pdb/Welcome.do.
30. M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.
31. Source Code NCOILS, 1999. http://www.russell.embl.de/cgi-bin/coils-svr.pl.
32. PAIRCOIL implementation by the authors, 1995. http://theory.lcs.mit.edu/ bab/computing.
33. J.M. Sempere and P. García. Learning locally testable even linear languages form positive data. *LNAI*, 2484:225–236, 2002.