Biosequence Processing through Grammatical Inference Methods

José M. Sempere
Research Group on Computation Models and Formal Languages
Universitat Politècnica de València

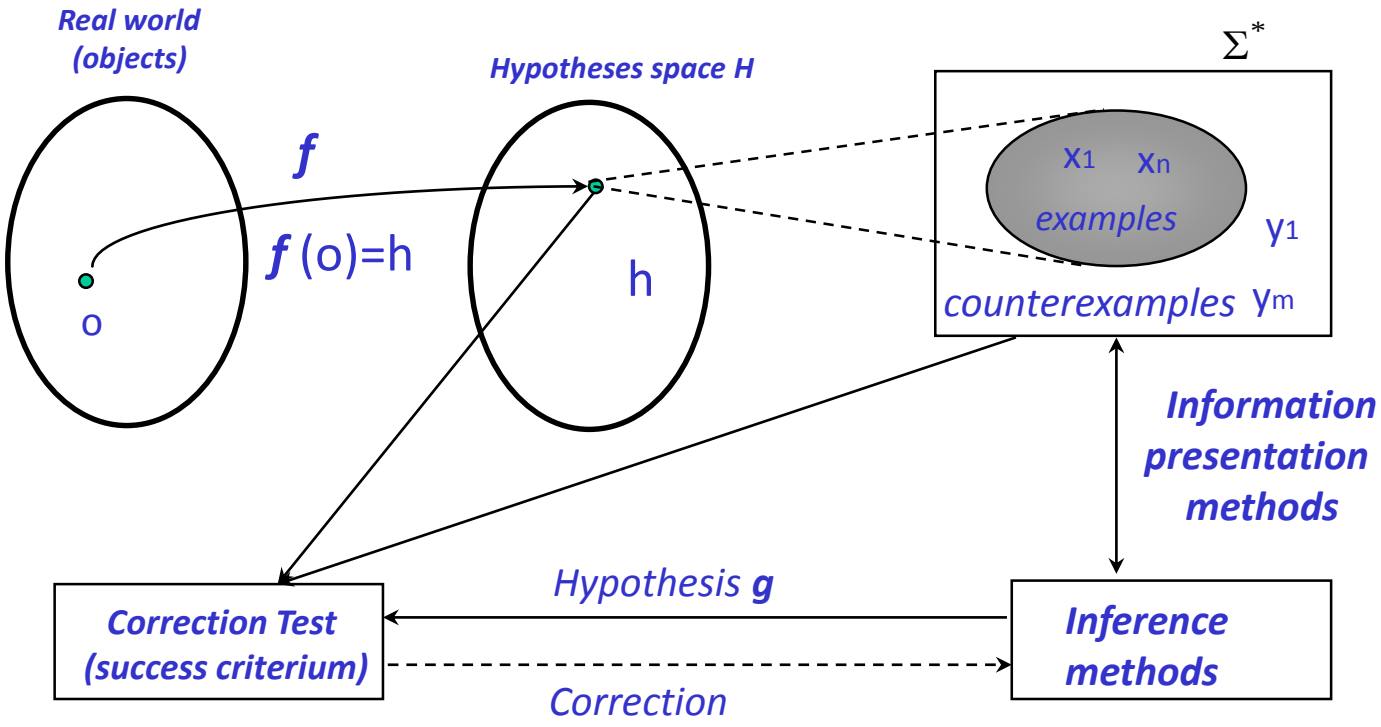
Biosequence Processing through Grammatical Inference Methods

Outline

1. **Basic Aspects of Grammatical Inference**
2. **The Biosequence Information**
3. **Making Protein motif prediction by GI: coiled-coil and transmembrane**

Basic aspects of Grammatical Inference

Grammatical Inference is an inductive inference technique to learn formal languages.



The definition of a Grammatical Inference approach implies

1º Define a method to assign hypotheses to the real objects

Acceptors
Generators

2º Define a method for information presentation (examples and counterexamples)

Text (examples) { **arbitrary** / **recursive** / **primitive-recursive**

Informant (examples + counterexamples, oracles) { **arbitrary** / **methodic** / **on request**

3º Define the inductive inference method (the algorithm)

4º Define a success criterium

Identification (… in the limit)
Aproximation (PAC)

Two taxonomies of GI methods

(a) Depending on the language class to be learned

Characterizable : The algorithm learns (identifies or aproximates) any language in a predefined language class. The method defines a search space and it gives algebraic properties to the language class

Heuristic : The algorithm learns (identifies or aproximates) languages which «some times» belong to a pre-defined language class

(b) Depending on the techniques to explore the search space

Enumerative : The algorithm defines an enumeration of all the hypotheses and tests everyone of them up to the target one.

Constructive : The algorithm builds a new hypothesis from the information received during the learning process

- **Incremental** : The new hypothesis is built from the previous one and the additional information.
- **Non-incremental** : The new hypothesis is built, every time, from the information received so far.

Identification in the Limit [Gold 67]

$g_t = G(i_1, i_2, \ldots, i_t)$, where

- g_t is the guess of the name of unknown language from class L at time t
- G is the guessing/learning algorithm
- $i_1, i_2, \ldots, i_t$ is the information sequence received up to time t

G identifies L in the Limit if

$\exists t$ (t is finite) $g_t = g_{t+1} = \ldots = g_\infty$ is correct

Probably Approximately Correct (PAC) Learning [Valiant 84]

Alg A PAC-learns concept class C by hypothesis class H if for any target f in C, any distrib D over X (sample space), any $\varepsilon, \delta > 0$,

- A uses at most poly(n,1/ε,1/δ,size(f)) examples and running time.
- With probability 1-δ, A produces h in H of error at most ε.

A negative result for Grammatical Inference

No superfinite language class can be inferred from only positive data

Example $L = a^* \cup P_{fin}(a^*)$

Some language classes which can be inferred from positive data

- **kTS(S) (for every k ≥ 2)**
- **kREV (for every k ≥0)**
- **k-piecewise TS(S)**
- **TDRL**
- **Reductions to the previous ones**

Some language classes which can be inferred from complete data

- **RE** — **Enumerative**
- **CF** — **Enumerative or structural**
- **LIN** — **Enumerative or structural**
- **REG** — **Constructive in polynomial time**

Learning k-TSS languages

For any set of positive data (text) S

$\Sigma(S)$ is the minimal alphabet to define the words in S

$I_k(S)=\{u \mid uv\in S, |u|=k-1, v\in S(S)^*\}\cup\{x\in S \mid x|<k-1\}$

$F_k(S)=\{v \mid uv\in S, |v|=k-1, v\in S(S)^*\}\cup\{x\in S \mid x|<k-1\}$

$T_k(S)=\{v \mid uvw\in S, |v|=k, u,w\in S(S)^*\}$

Input: $k\geq 2$, S

Output: DFA $A_k=(Q,\Sigma,\delta,q_0,Q_f)$ // $Q\subseteq\Sigma^{\leq k-1}$, $\delta\subseteq(Q\times\Sigma\times Q)$ //

$(\Sigma,I,F,T)= (\Sigma(S),I_k(S),F_k(S),T_k(S))$;

$Q=\{\lambda\}$; $\delta=\varnothing$; $q_0=\lambda$;

$\forall a_1...a_m\in I$

For j=1 to m

$Q=Q\cup\{a_1...a_j\}$; $\delta=\delta\cup\{\{a_1...a_{j-1},a_j, a_1...a_j\}\}$;

End For

end $\forall$

$\forall a_1...a_k\in T$

$Q=Q\cup\{a_2...a_k\}$; $\delta=\delta\cup\{\{a_1...a_{k-1},a_k, a_2...a_k\}\}$;

End $\forall$

$Q_f=F$;

$A_k=(Q,\Sigma,\delta,q_0,Q_f)$

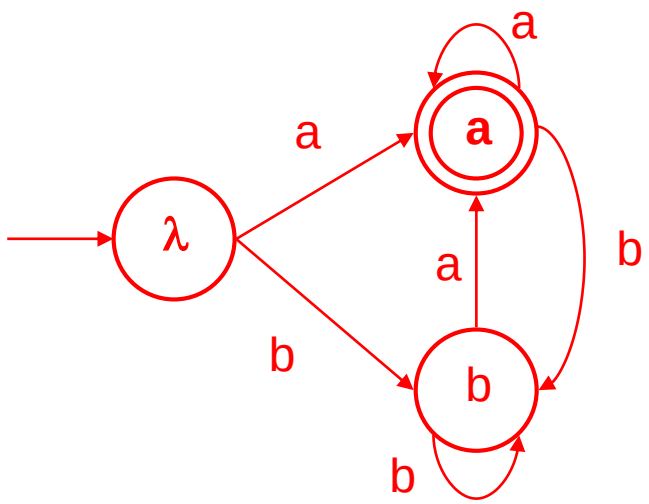
Learning k-TSS languages

Example 1

k = 2
S = { abba, aaabba, bbaaa, bba }

Σ= { a, b }
I = { a, b }
T = {ab, bb, ba, aa }
F = { a }

A:



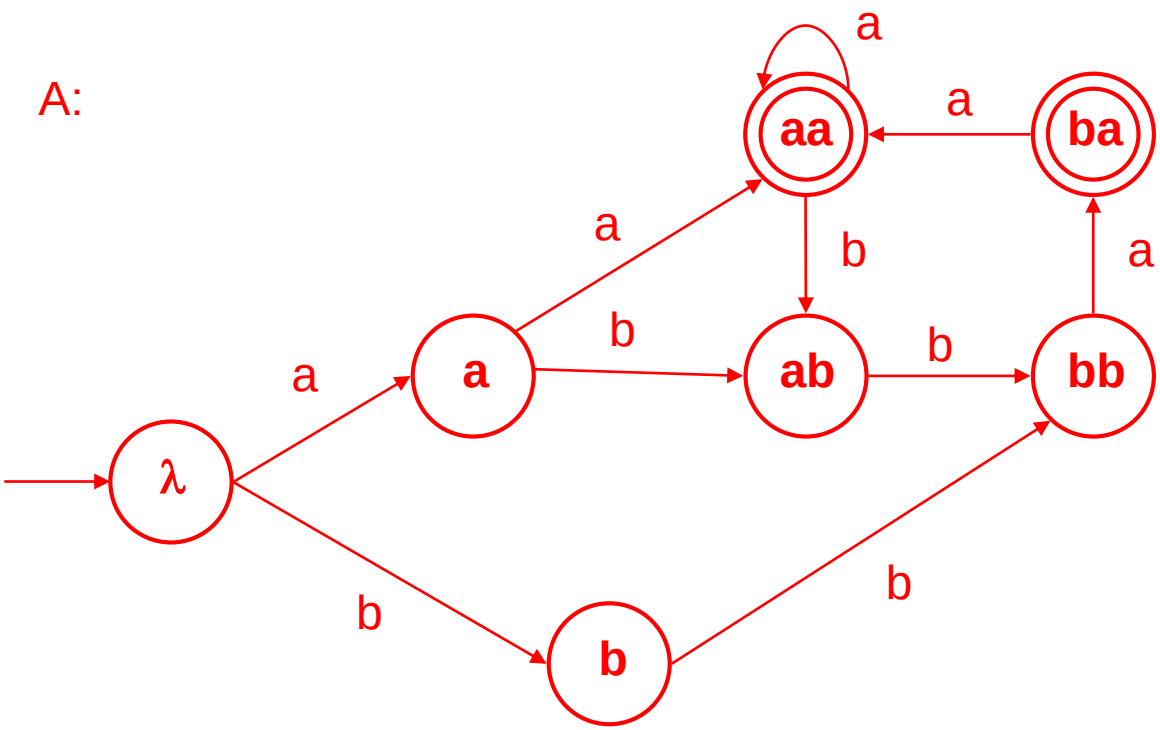
Learning k-TSS languages

Example 2

k = 3
S = { abba, aaabba, bbaaa, bba }

Σ= { a, b }
I = { ab, aa, bb }
T = {abb, bba, aaa, aab, abb, bba, baa }
F = { ba, aa }

A:

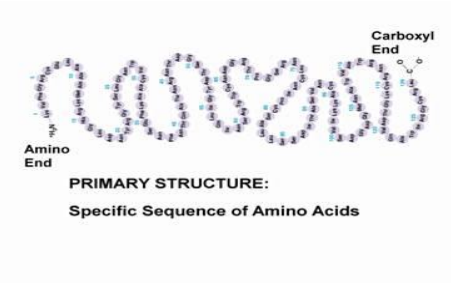


The Biosequence information

(from IUPAC Compendium of Chemical Terminology 2nd Edition (1997))

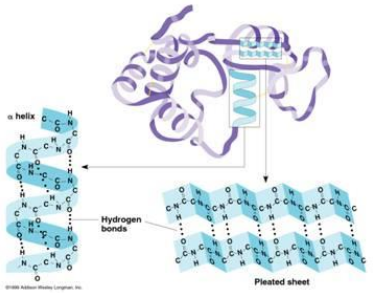
(a) Primary structure

In the context of *macromolecules* such as *proteins,* the constitutional formula, usually abbreviated to a statement of the sequence and if appropriate cross-linking of chains.



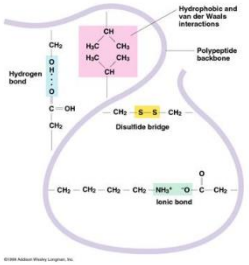
(b) Secondary structure

The conformational arrangement (α–helix, β-pleated sheet, etc.) of the backbone segments of a macromolecule such as a polypeptide chain of a protein without regard to the conformation of the side chains or the relationship to other segments.



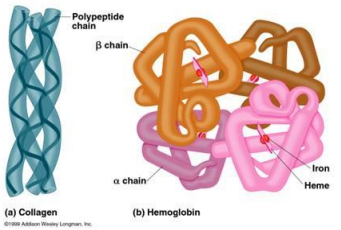
(c) Tertiary structure

The spatial organization (including conformation) of an entire protein molecule or other macromolecule consisting of a single chain.

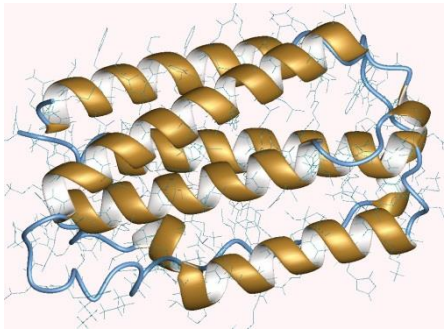


(d) Quaternary structure

The defined organization of two or more macromolecules with tertiary structure such as a protein that are held together by hydrogen bonds and van der Waals and coulombic forces.



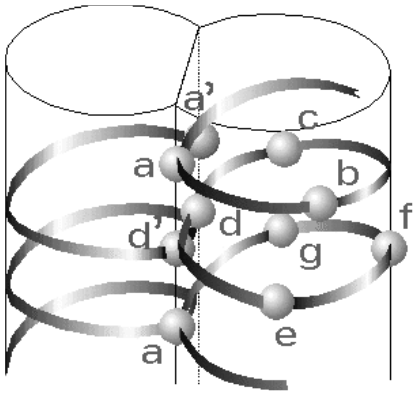
The coiled coil protein prediction problem (CCPP)



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 coiled coil protein prediction problem (CCPP)

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 viruses or parasites*** [SW98] [CK98].

Predictions based on analysis of primary sequences suggest that **approximately *2-3 % of all protein residues form coiled coils*** [WKB97].

The CCPP problem can be enunciated as follows:

"Given the primary structure sequence of any protein establish whether the protein contains the coiled coil motif or not"

Previous solutions to the CCPP problem (I)

Several programs for predicting coiled coil domains have been proposed.

The most relevant to large-scale annotations are : ***coils*** [LDS91],
paircoil [BWW+95], and
multicoil [WKB97].

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.

All of them are based on a ***Position Specific Scoring Matrix*** (PSSM) (also known as ***Position Weighted Matrix***) approach [MSS+02].

This 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.

Previous solutions to the CCPP problem (II)

The work by **Lupas et al**. [LDS91] 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.

Problem: the method leads to a significant number of false positives

Berger et al. [BWW+95] 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.

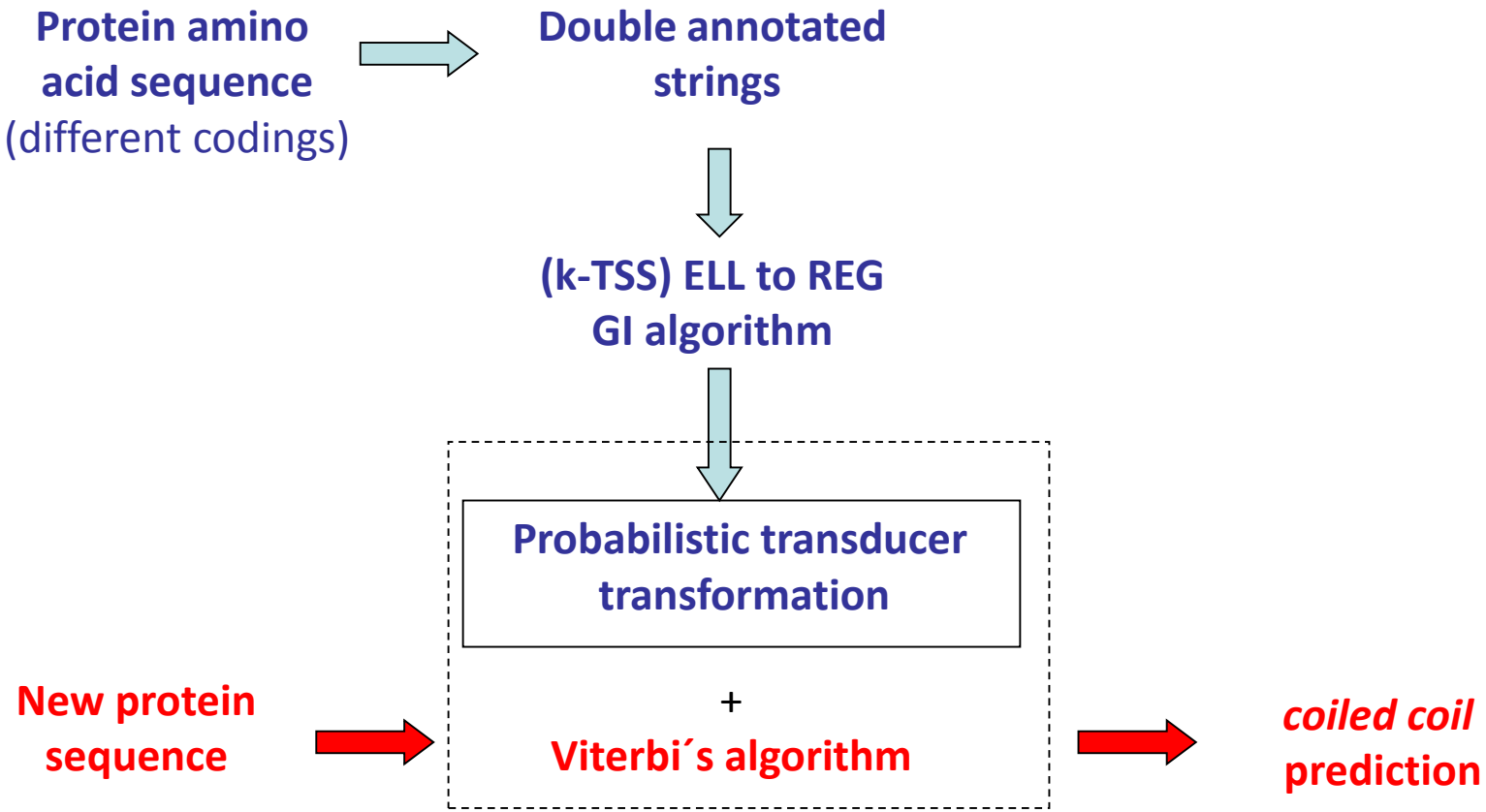
Previous solutions to the CCPP problem (and III)

Grammatical approaches to locate protein structures

- **α-helix** structures in protein sequences [YIK94]
- the **coiled coil** motif [LCV+04,LCV+05]

The problem of locating general coiled coil motifs still remains open !!!

The approach based on GI



The approach based on GI

Amino acid (AA)	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

Protein amino acid sequence (different encodings)

$x = x_1\, x_2 \ldots x_n$

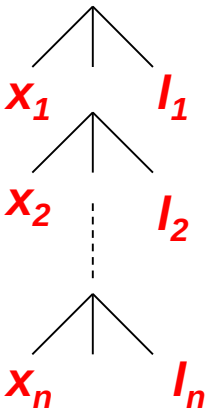
⇩

Double annotated strings

$l = l_1\, l_2 \ldots l_n$

$$l_i = \begin{cases} \text{c} & \text{if AA } x_i \text{ belongs to a coiled coil region} \\ \text{n} & \text{otherwise} \end{cases}$$

Even linear structure



$x = x_1\, x_2 \ldots x_n\, l_n \ldots l_2\, l_1$

The approach based on GI

(k-TSS) ELL to REG GI algorithm

$A \rightarrow a\ B\ b \mid b \mid \lambda$

Even linear grammar

$A \rightarrow [ab]\ B \mid b \mid \lambda$

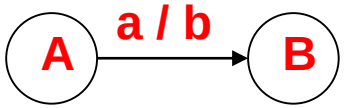
Regular grammar

The σ transformation [SG94]

If the regular language is k-TSS then the ELL is k-TSS [SG02]

Transducer transformation

$A \rightarrow a\ B\ b$



$A \rightarrow \lambda$

Experiments and results

- **Data sets:** (1) **SwisProt Database (rel. 40, April 2003)**
  (2) **Delorenzi and Speed database obtained from Protein Data Bank [DS02]**

- **Dayhoff and P/H codings**

- **Measures to evaluate the results**

TP: True Positives
TN: True Negatives
FP: False Positives
FN: False Negatives

$$S_n = \frac{TP}{TP + FN}$$

$$S_p = \frac{TP}{TP + FP}$$

Correlation Coefficient **(CC)**

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

Approximate Correlation **(AC)**

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

Experiments and results

Experiment 1: SwisProt Database (350 protein sequences)
Test and learning protocol : *iterated leaving-one out*

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

Experiments and results

Experiment 2: Delorenzi's Database (397 coiled coil protein sequences)
Test and learning protocol : ***iterated leaving-one out***

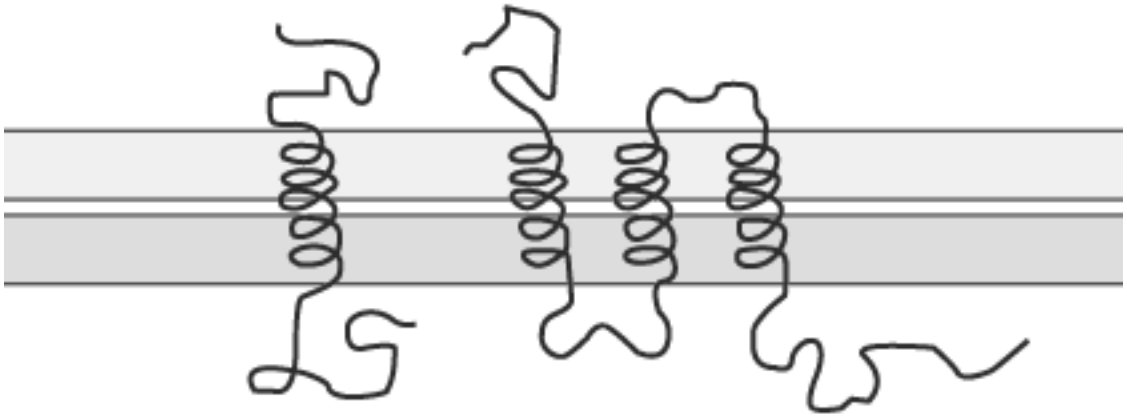
Method		*Sn*	*Sp*	*CC*	*AC*
coils		0.6688	0.8552	0.6372	0.6222
paircoil		0.6511	0.8489	0.6693	0.5972
IGcoils (Dayhoff coding)	k=2	0.6269	0.7420	0.6501	0.5058
	k=3	0.6353	0.7616	0.6730	0.5342
	k=4	0.6275	0.7778	0.6793	0.5654
	k=5	0.5709	0.8249	0.6842	0.5729
	k=6	0.5262	0.8730	0.7395	0.5692
	k=7	0.5465	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

Experiments and results

Experiment 3: Delorenzi's Dataset (397 coiled coil protein sequences)
SwissProt Database (350 coiled coil protein sequences)
Test : Delorenzi's Dataset (1525 non coiled coil protein sequences)
Dayhoff coding

		IGcoils			coils	paircoil
		k=6	k=7	k=8		
% error rate	SwissProt	0.0118	0.0175	0.0254	0.0058	0.0023
	Delorenzi's	0.0036	0.0030	0.0016		
# of erroneous sequences	SwissProt	104	123	140	57	12
	Delorenzi's	26	18	12		

The transmembrane protein prediction problem (TPP)



Schematic representation of two transmembrane proteins (one single spanning and one multi-spanning)

The TPP problem can be enunciated as follows:

"Given the primary structure sequence of any protein segment establish which amino acids are out/through/in the membrane"

The transmembrane protein prediction problem (TPP)

1. **Sequences**: labelled proteins (sequences of labelled amino acids)

$$M = \begin{cases} A_i T_i L_i Q_i S_i E_i H_i K_i I_M L_M A_M F_M S_M A_M T_o P_o W_o L_o H_o \\ A_o A_o V_o I_M L_M G_M T_M V_M S_i D_i G_i W_i L_M V_M S_M W_M W_o N_o L_o N_o \\ M_o N_o S_o L_M A_M C_M F_M Y_M A_M P_i F_i K_i K_i K_i W_i S_i W_i E_i T_i M_i \end{cases}$$

2. **Simplification function**:

$$M_I = \begin{cases} iMo, \\ oMiMo, \\ oMi \end{cases}$$

3. **Extraction function**:

$$M_i = \begin{cases} ATLQSEHK, \\ SDGW, \\ PFKKKWSWETM \end{cases} \qquad M_M = \begin{cases} ILAFSA, \\ ILGTV, \\ \\ LVSW, \\ LACFYA \end{cases} \qquad M_o = \begin{cases} TPWLH, \\ AAV, \\ \\ WNLN, \\ MNS \end{cases}$$

The transmembrane protein prediction problem (TPP)

4. Training:

- Automata A_I, A_M, A_o, Al are built from the sequences of each subset, with k-tss algorithm

- A probabilistic transducer is built by substituting each transition l = {i,M,o} of the automaton A_l with the corresponding automaton (A_i, A_M or A_o). Labels i,M,o are the output of each edge of the transducer.

5. Test:

We use Viterbi's algorithm and a test sequence to find the most likely transduction (the transduction is a sequence of labels, that is, the prediction of the method)

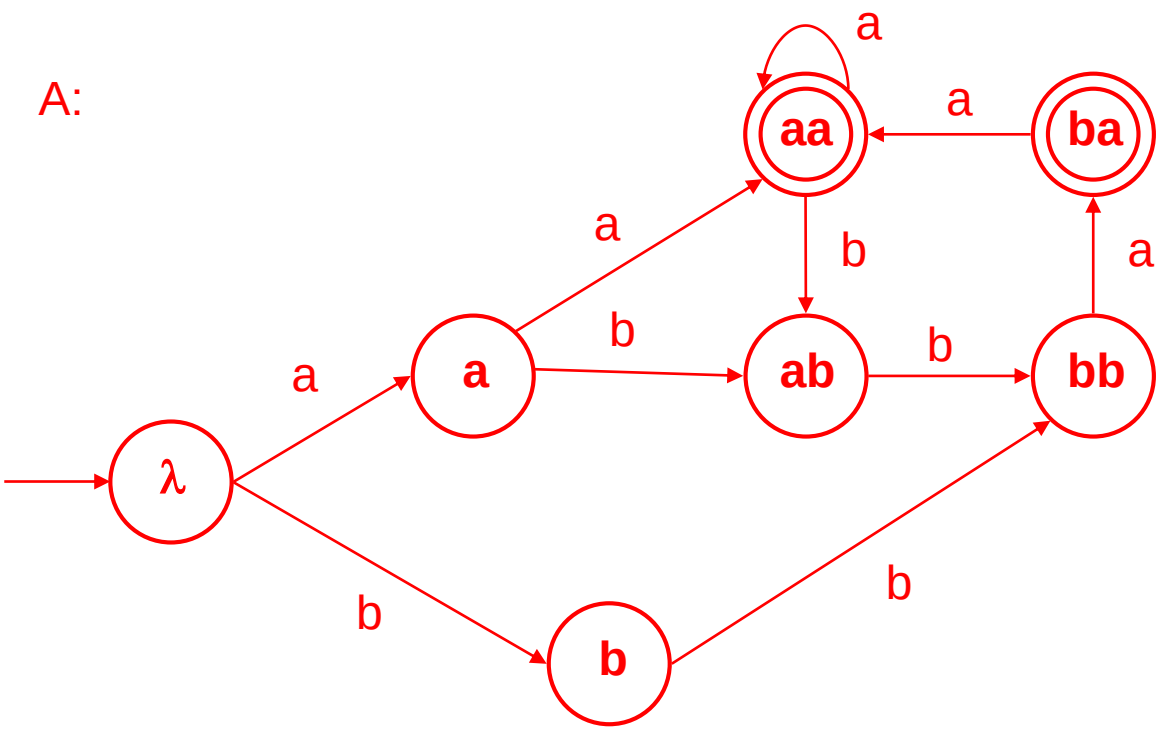
Learning k-TSS languages

Example 2

k = 3
S = { abba, aaabba, bbaaa, bba }

Σ= { a, b }
I = { ab, aa, bb }
T = {abb, bba, aaa, aab, abb, bba, baa }
F = { ba, aa }

A:



The transmembrane protein prediction problem (TPP)

Database

We used a dataset to train and test our method:
TMHMM database: set of 160 transmembrane proteins, available at: http://www.cbs.dtu.dk/krogh/TMHMM.

In order to test our method, we followed a leaving one out scheme.

The transmembrane protein prediction problem (TPP)

Results

TMHMM database

	Sn	Sp	CC	AC
igS	0.877	0.784	0.733	0.728
igTM config. 1	0.808	0.810	0.707	0.702
config. 2	0.819	0.796	0.715	0.707
TMHMM 2.0	0.900	0.879	0.830	0.827

Results of the experiments carried out with igS over the 160 proteins dataset, compared to the results of TMHMM 2.0 and and the two best configurations of igTM over the same dataset.

The transmembrane protein prediction problem (TPP)

Results

TMHMM database

	Sn	Sp	CC	AC
igS	0.877	0.784	0.733	0.728
igTM config. 1	0.808	0.810	0.707	0.702
config. 2	0.819	0.796	0.715	0.707
TMHMM 2.0	0.900	0.879	0.830	0.827

Results of the experiments carried out with igS over the 160 proteins dataset, compared to the results of TMHMM 2.0 and and the two best configurations of igTM over the same dataset.