Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 335 (2005) 67–92

Theoretical Computer Science

www.elsevier.com/locate/tcs

Modeling and predicting all-α transmembrane proteins including helix–helix pairing

Jérôme Waldispühl*, Jean-Marc Steyaert

LIX, École Polytechnique, 91128 Palaiseau Cedex, France

Abstract

Modeling and predicting the structure of proteins is one of the most important challenges of computational biology. Exact physical models are too complex to provide feasible prediction tools and other ab initio methods only use local and probabilistic information to fold a given sequence. We show in this paper that all-α transmembrane protein secondary and super-secondary structures can be modeled with a *multi-tape S-attributed grammar*. An efficient structure prediction algorithm using both local and global constraints is designed and evaluated. Comparison with existing methods shows that the prediction rates as well as the definition level are sensibly increased. Furthermore this approach can be generalized to more complex proteins.

Keywords: All-α transmembrane proteins; Multi-tape S-attributed grammar; Folding modeling; Structure prediction; Helix–helix pairing

1. Introduction

Protein folding is a complex mechanism which has been studied since more than 30 years. A number of approaches have been proposed and studied, ranging from bottom-up folding based on elementary laws of physics and chemistry to direct structure prediction based on AI programs, that include sequence analysis and statistics. A comprehensive survey of the most important methods that have been used during this period can be found in [26],

* Corresponding author.
E-mail address: waldispu@lix.polytechnique.fr (J. Waldispühl).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.12.018

an article dedicated to Levinthal. From this overview, it becomes clear that attempting to folding proteins in silico by purely physical models is too complex to give systematically a good answer (since there are too many possible branching points), and that modeling with more abstract tools such as statistics on sequences, neural networks or other models based on specific properties of subsequences is likely to give more rapidly a more precise prediction with respect to the exact nature of the secondary structure: this point of view is at the origin of the PrISM package developed by Yang [59]. However, most underlying models take into consideration local properties only, a defect that leads inherently to a loss in predictive power.

In this paper we keep in the tracks of Honig's survey and propose a model (in fact a family of models) based on concepts quite familiar to computer scientists: formal grammars and dynamic programming. We propose to predict secondary structures along the following principles: structure depends (i) on local properties of subsequences, such as the ability to form an α-helix or a β-strand, and (ii) on long-range phenomena giving stability to the molecule according to thermodynamic principles and chemical interaction with the surrounding environment, which result in towers of helices and (parallel or anti-parallel) grouping of β-strands into β-sheets or β-barrels.

To describe the model we use grammars, in which most of the rules are context-free to express long-range interactions and the others are equivalent to regular expressions for local properties, with possibly stochastic rules. We also have to complexify the standard linguistic approach [49,54] and adopt the multi-tape framework of Lefebvre and Steyaert [37,38]. Since building a general grammar for all types of proteins is complex (if not impossible), we restrict ourselves to a subfamily of proteins whose structure is particularly adapted to this simple formalism: transmembrane proteins whose secondary structure is mainly formed with α-helices. Then the set of all their possible configurations can easily be described by extended context-free grammars: almost all helix pairings are parallel or anti-parallel, which is not the general case for soluble proteins (see the Globin-like family for example). In fact, parallel configurations cannot be modeled directly with context-free grammars but they can be expressed with some combinatorial techniques using attributes. Many well-known softwares deal with such proteins and predict their secondary structures: HMMTOP2 [23,7], MEMSAT [29], PHDpsihtm [47,46], PRED-TMR [43], SOSUI [25], TMHMM [51,34] and TOPPRED2 [52]. We prove in this article that our simple model gives results that compare favorably to these programs; we find with a low rate of false positive new sites for α-helices that they missed.

We do not pretend to perform folding at any moment, but we think that our method is useful for proposing structures that are very likely to exist. Also, our tools allow many variations in the choice of the various parameters that determine local substructures—such as residue interactions or local geometrical constraints imposed by some amino acids (Proline, Glycine...) [17,10]—as well as free energy or any similar criterion that could be used to specify long-range interactions, as an attempt to take into account all atomic forces, for which some programming partial answers have been developed [40,18,24,30,22]. The program that we have developed is in fact a compiler which, given a formal description of the family of structures that we want to recognize, identify and optimize, produces a real program; this program is then used to analyze protein sequences and ultimately to provide a tentative secondary structure for each of them.

In this paper, we illustrate our methodology by predicting the super-secondary structure of all-α transmembrane proteins using as main stability factor the hydrophobicity [44,27,57]. We prove that with a very simple mechanism we gain over all other methods, and we give hints as to possible additional gains. In Section 2 we introduce the basic notions to be used in the sequel. We describe in Section 3 the approximate physical model which is a simple thermodynamical model for all-α transmembrane proteins. We give an overview of our *MTSAG* model in Section 4. Finally, we evaluate the predictions given by this model and compare our results with existing prediction methods in Section 5.4. One of the main improvements of this article is that, to our knowledge, it is the first time that a method involving systematic long-range interaction (residue contacts) is used as a reliable prediction tool.

2. Multi-tape S-attributed context-free Grammars

In this section we recall the basic notions. The first part is well known to computer scientists, but could be of interest for biologists. The second part is a generalization which is necessary to describe complex long-range interactions.

2.1. S-attributed context-free grammars

Definition 1 (*Context-free grammar*). A context-free grammar $G = (V_T, V_N, P, S)$ consists of a finite set of terminals V_T, a finite set of nonterminals V_N such that $V_T \cap V_N = \emptyset$, a finite set of productions (rewriting rules) P and a start symbol $S \in V_N$. Let $V = V_T \cup V_N$ denote the vocabulary for the grammar. Each production in P has the form $A \to \alpha$, where $A \in V_N$ and $\alpha \in V^\star$. A is the left-hand side of the production and α its right-hand side.

The transitive closure of the derivation relation $\to$ is denoted by $\overset{\star}{\to}$ in general and $\overset{+}{\to}$ when it contains at least one non-trivial derivation. A derivation tree is the planar representation of a sequence of derivations $A \to \alpha$ such that $A \in V_N$ and $\alpha \in V^\star$. The set of strings in $V_T^\star$ derived from S is called the language generated by G and it is denoted by $L(G)$. The empty string is denoted by ε. In order to keep the number of derivation trees finite for a given word $\omega \in L(G)$, we assume that the grammar is non-circular, which means that no non-terminal A may verify $A \overset{+}{\to} A$. We also assume that the grammar is epsilon-free (i.e., it has no rules of the form $A \to \varepsilon$). An ambiguous grammar is a grammar for which there exists a string of symbols having at least two different derivation trees. For example, the grammar whose derivation trees describe t-RNA secondary structures has to be ambiguous because a given RNA has potentially several different secondary structures. Parsing is the process of finding a derivation tree for a string in $L(G)$, which is called the parse tree of the sequence.

S-attributed context-free grammars, which are a proper subset of attributed-grammars introduced by Knuth in his seminal paper [33], are an extension of context-free grammar allowing the assignment of a value (called attribute) to each vertex of a derivation tree.

Definition 2 (*S-attributed grammar*). An S-attributed grammar, denoted by $G = (V_T, V_N, P, S, \mathcal{A}, S_\mathcal{A}, F_P)$, is an extension of the context-free grammar $G = (V_T, V_N, P, S)$; an

attribute $x \in \mathcal{A}$ is attached to each symbol $X \in V$, and a string of attributes $\lambda \in \mathcal{A}^\star$ to each string $\alpha \in V^\star$. $S_\mathcal{A}$ is a function from V_T to $\mathcal{A}$ assigning attributes to terminals. F_P is a set of functions from $\mathcal{A}^\star$ to $\mathcal{A}$. A function $f_{A\to\alpha}$ is in F_P for every production $A \to \alpha$ in P.

The attribute of a string $\alpha \in V^\star$, denoted by λ_α, is the concatenation of the attributes of the symbols in α. When a function $f_{A\to\alpha}$ is applied to the attribute λ_α derived from A it returns the attribute $x = f_{A\to\alpha}(lambda_\alpha)$. Thus, the functions of F_P determine the bottom-up computation of the attribute of non-terminal A in derivations $fAg \overset{\star}{\to} u$, where u belongs to $L(G)$. Attributes are thus synthesized bottom-up (hence the denomination S-attributed grammars).

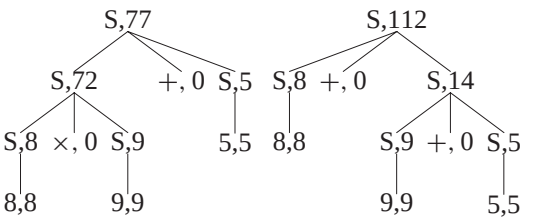
Example 1. We consider the following ambiguous context-free grammar for arithmetical expressions:

$$\begin{aligned} S &\to S + S, \\ S &\to S \times S, \\ S &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9. \end{aligned}$$

We extend it into an S-attributed grammar such that each attribute stores the value of the arithmetical expression corresponding to the derivation subtree rooted at this node.

$$\begin{aligned} &\mathcal{A} = \mathbb{N} \\ &S_\mathcal{A}(0) = 0 \\ &\vdots \\ &S_\mathcal{A}(9) = 9 \\ &S_\mathcal{A}(+, \times) = 0 \end{aligned} \qquad F_P = \left\{ \begin{array}{l} f_{S\to S+S}(xyz) = x + z \\ f_{S\to S\times S}(xyz) = x \times z \\ f_{S\to a\in V_T}(x) = x \end{array} \right\}$$

Thus, the sequence $8 \times 9 + 5$ can be parsed in different ways corresponding to different derivation trees, hence producing different values, as shown below:



As previously noted, grammars used for biological sequence analysis are ambiguous because they aim at describing the space of all possible configurations. Attributes are designed to give an evaluation of the quality for each of them, providing thus a choice criterion. For example, if the attribute of a vertex is an energy or a probability, the criterion may be the selection of the derivation tree with the lowest energy or the highest probability at the root. (But attributes are not restricted to simple real values in general even if this paper deals mainly with this type of value.) The selection of the best possible values for an attribute is done by means of a function called optimization constraint.

Definition 3 (*Optimization constraint*). Let G be an S-attributed grammar and let $a_1 \dots a_n$ be a string of $L(G)$. Let x be the attribute of a derivation $A \to \alpha_1 \to \cdots \to \alpha_k \to a_{i+1} \dots a_j$ and let x' be the attribute of another derivation $A \to \alpha'_1 \to \cdots \to \alpha'_k \to a_{i+1} \dots a_j$. The optimization constraint $\mathcal{C}_{\mathcal{A}}$ applied to these two derivations, takes as input the attributes x and x' and returns either x or $'$; noted $\mathcal{C}_{\mathcal{A}}(x, x')$.

In practice, $\mathcal{C}_{\mathcal{A}}$ will be a comparison function between x and x' which returns the optimal attribute (the lowest energy, if attributes are free energies). Finally, we denote by λ_ω the optimal attribute of a string $\omega \in L(G)$.

2.2. Multi-tape S-attributed grammars

We extend now this formalism in order to use an "m-tape" alphabet.

Definition 4 (*m-tape index*). An m-tape index $\vec{\imath}$ is a vector of $\mathbb{Z}^m$. The notation $\vec{\imath}^{(k)}$ will denote the kth tape. We shall say that an m-tape index $\vec{\imath}$ is inferior to an m-tape index $\vec{\jmath}$ if this order stands on every tape, and we shall denote this relation by $\vec{\imath} \leqslant \vec{\jmath}$.

The m-tape index $\vec{1}$ has value 1 on all tapes. The sum of two m-tape indices $\vec{\imath}_1$ and $\vec{\imath}_2$ is an m-tape index $\vec{\jmath}$ defined by $\vec{\jmath}^{(k)} = \vec{\imath}_1^{(k)} + \vec{\imath}_2^{(k)}$. Thus, $\vec{\imath} \leqslant \vec{\jmath}$ also means $\vec{\imath} - \vec{\jmath} \leqslant \vec{0}$.

Definition 5 (*m-tape string*). Given m alphabets $\Sigma^{(i)}$ $(1 \leqslant i \leqslant m)$, an m-tape string is a vector of m strings $\left(a_{\vec{1}^{(1)}}^{(1)} \dots a_{\vec{n}^{(1)}}^{(1)}, \dots, a_{\vec{m}^{(m)}}^{(m)} \cdots a_{\vec{n}^{(m)}}^{(m)}\right)$, where each string $a_{\vec{1}^{(i)}}^{(i)} \cdots a_{\vec{n}^{(i)}}^{(i)}$ belongs to $(\Sigma^{(i)})^\star$. We shall denote the set of string so defined by $\langle \Sigma^\star \rangle = \bigotimes_{i=1\cdots n} \Sigma^{(i)^\star}$. As a shorthand, any m-tape string $\left(a_{\vec{1}^{(1)}}^{(1)} \cdots a_{\vec{n}^{(1)}}^{(1)}, \dots, a_{\vec{m}^{(m)}}^{(m)} \cdots a_{\vec{n}^{(m)}}^{(m)}\right)$ may be denoted by $a_{\vec{1}} \cdots a_{\vec{n}}$. Substrings of $a_{\vec{1}} \cdots a_{\vec{n}}$ will be denoted by $a_{\vec{\imath}} \cdots a_{\vec{\jmath}} = \left(a_{\vec{\imath}^{(1)}}^{(1)} \cdots a_{\vec{\jmath}^{(1)}}^{(1)}, \dots, a_{\vec{\imath}^{(m)}}^{(m)} \cdots a_{\vec{\jmath}^{(m)}}^{(m)}\right)$ with the usual conventions that $\vec{1} \leqslant \vec{\imath}, \vec{\jmath} \leqslant \vec{n}$ and, if $\vec{\imath}^{(k)} > \vec{\jmath}^{(k)}$, $a_{\vec{\imath}^{(k)}}^{(k)} \cdots a_{\vec{\jmath}^{(k)}}^{(k)} = \varepsilon$.

Example 2 (*abba, dcd*). is a 2-tape string on $\Sigma^{(1)} = a, b$ and $\Sigma^{(2)} = c, d$. We shall also write this 2-tape string as $\begin{smallmatrix} a & b & b & a \\ d & c & d \end{smallmatrix}$, which is a somewhat more natural notation in the context of alignments. This 2-tape string $a_{\binom{1}{1}} \cdots a_{\binom{4}{3}} = \begin{smallmatrix} a & b & b & a \\ d & c & d \end{smallmatrix}$ has a 2-tape substring $a_{\binom{2}{1}} \cdots a_{\binom{3}{2}} = \begin{smallmatrix} b & b \\ d & c \end{smallmatrix}$.

Definition 6 (*m-tape alphabet*). An m-tape alphabet Σ is a product of m alphabets $\Sigma^{(i)}$ augmented with the empty string: $\Sigma = \bigotimes_{i=1\cdots m} (\Sigma^{(i)} \cup \{\varepsilon\})$.

Definition 7 (*m-tape alignment*). An element $a_1 \cdots a_l$ of the free monoid $\Sigma^\star$, generated by formal concatenation of m-tape elements of Σ, is called an m-tape alignment of length l. The empty alignment of $\Sigma^\star$ is denoted by ε.

Definition 8 (*ε-deletion*). Given any *m*-tape alignment $a_1 \dots a_l$, we get an *m*-tape string $a_{\vec{1}} \dots a_{\vec{n}}$ by concatenation of symbols of the projection of $a_1 \dots a_l$ on every tape.

$$\begin{aligned} \Sigma^\star &\to \langle \Sigma^\star \rangle, \\ a_1 \dots a_l &\to a_{\vec{1}} \dots a_{\vec{l}} = \langle a_1 \dots a_l \rangle. \end{aligned}$$

Example 3. The 2-tape string $\begin{smallmatrix} a & b & b & a \\ d & c & d & \end{smallmatrix}$ may be defined as an ε-deletion of the alignments $\left\langle \left[\begin{smallmatrix} \varepsilon \\ b \end{smallmatrix}\right] \left[\begin{smallmatrix} a \\ \varepsilon \end{smallmatrix}\right] \left[\begin{smallmatrix} b \\ \varepsilon \end{smallmatrix}\right] \left[\begin{smallmatrix} b \\ c \end{smallmatrix}\right] \left[\begin{smallmatrix} a \\ d \end{smallmatrix}\right] \right\rangle$ or $\left\langle \left[\begin{smallmatrix} a \\ \varepsilon \end{smallmatrix}\right] \left[\begin{smallmatrix} b \\ d \end{smallmatrix}\right] \left[\begin{smallmatrix} \varepsilon \\ c \end{smallmatrix}\right] \left[\begin{smallmatrix} b \\ \varepsilon \end{smallmatrix}\right] \left[\begin{smallmatrix} a \\ d \end{smallmatrix}\right] \right\rangle$.

Once *m*-tape alphabets and strings have been defined, we naturally extend a context-free grammar into an *m*-tape context-free grammar.

Definition 9 (*m-tape context-free grammar*). An *m*-tape context-free grammar $G = (V_T, V_N, P, S)$ is a classical context-free grammar such that V_T *is a subset of an m-tape alphabet*.

Notice that in the *m*-tape case, $L(G)$ is not the set of derivable strings but the set of ε-deleted such strings.

Furthermore, as in the mono-tape case, an *m*-tape context-free grammar can be extended into an *m*-tape S-attributed grammar by addition of attributes and functions to compute the non-terminal attributes.

Definition 10 (*m-tape S-attributed grammar*). An *m*-tape *S*-attributed grammar, denoted by $G = (V_T, V_N, P, S, \mathcal{A}, S_\mathcal{A}, F_P)$, is an extension of an *m*-tape context-free grammar $G = (V_T, V_N, P, S)$, where an attribute $x \in \mathcal{A}$ is attached to each symbol $X \in V$ and a string of attributes $\lambda \in \mathcal{A}^\star$ to each string $\alpha \in V^\star$. $S_\mathcal{A}$ is a function from V_T to $\mathcal{A}$ assigning attributes to terminals. F_P is a set of functions from $\mathcal{A}^\star$ to $\mathcal{A}$. A function $f_{A\to\alpha}$ is in F_P iff $A \to \alpha$ is in P.

3. The physical approximate model

Transmembrane proteins are mainly composed of α-helices that recombine at distance into more complex secondary structures. We limit our study to the set of predictable super-secondary structures known as all α-helix bundle formed by anti-parallel helices (see Fig. 1) [1] . In fact, only the closing pairing of the first and last helices of the bundle is allowed to be parallel (when the bundle has an odd number of helices). In this Section, we describe the different structural levels, starting from the local structure of an α-helix, and we show how to organize the global arrangement of such helices.

3.1. Local description of α helix

Roughly, an α-helix is a secondary structure characterized by hydrogen bonds between residues at positions *n* and $n+4$. It could also be seen as a coiling up of the primary structure

[1] Images obtained with PyMOL [13].

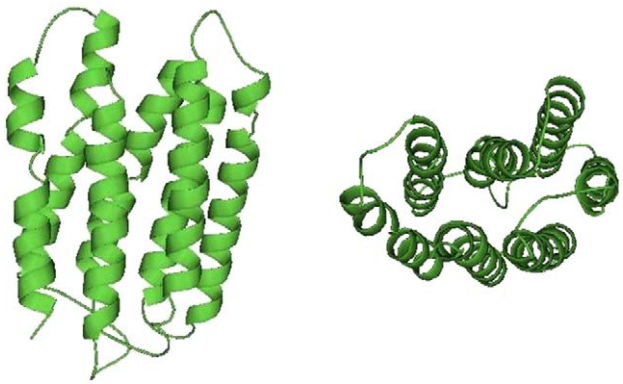


Fig. 1. Modeled protein class.

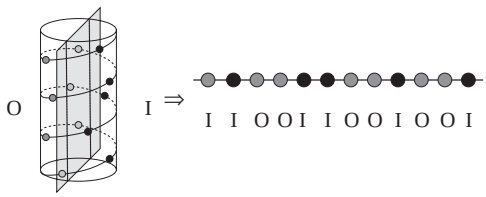


Fig. 2. Helix partition.

(the amino acid sequence) around a virtual axis. We describe a helix by the position of its residues on the helix surface: for this purpose, we define two faces, denoted *I* and *O*, in order to localize them. Each amino acid belongs to one of these faces and we can naturally associate them with their corresponding face. In this formalism a helix has a linear representation of its geometry given by the sequence of the positions of its amino acids, that is the sequence of their corresponding faces *I* and *O* (see Fig. 2). Let I_i and O_i be the two *helical faces* which compose the *i-th helical turn*. Since nature has no reason to stick to arithmetics, helices do not have an integer number of residues per turn: experimental measures give a periodicity of 3.6 amino acids per helical turn; since this property cannot be easily expressed by a deterministic automaton, we describe this structural regularity by a set of rules, which can be expressed by a variant of probabilistic regular expressions:

- a helix is an alternate sequence of faces I_i and O_i,
- a helical turn is composed of 3 or 4 amino acids,
- a helical face (I_i or O_i) compounds 1 or 2 amino acids,
- on the average the number of residues on a helical turn is 3.6.

3.2. Helix pairing

We define a helix pair as the (side by side) association of two helices, corresponding to, two *I*-faces facing each other. Let *I* be the common face and I^k be the *I*-face of the *k*th helix,

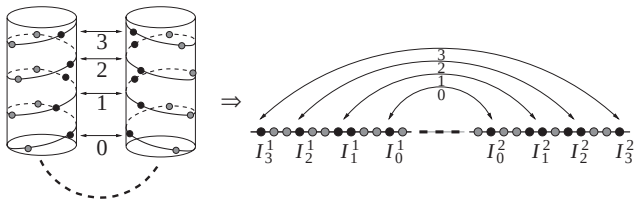


Fig. 3. Helix association.

then I_i^k will denote the helical face I of ith helical turn of the kth helix. As one can see in Fig. 3, a helix pairing implies the association of common helical faces that one can write as pairs (I_i^1, I_j^2) called *helical face pairs*. Due to topological constraints, if (I_i^1, I_j^2) and $(I_{i'}^1, I_{j'}^2)$ are two helical face pairs then $i' = i$ implies $j' = j$, and $i' = i + 1$ implies $j' = j + 1$. Moreover, let us assume that the helix pairs are anti-parallel and have the same number of helical turns (which means that all helical faces of the common face are paired): then, a helix pair could be described as a sequence of helical face pairs $\{(I_0^1, I_0^2), \ldots, (I_n^1, I_n^2)\}$ where n is the number of helical turns of the helix. The linear representation of the unfolded primary structure shows a kind of distant relationship (see Fig. 3) that could indicate well-bracketed expressions. We will see later that this kind of constraints is easily described as a context-free language.

3.3. Helix bundle

We describe a helix bundle with more than two α-helices as a composition of several helix pairings. Each helix of these super-secondary structures is paired with its two neighbors (in the primary structure): therefore we can write a helix bundle as a set of sequences of helix pairings. Because of the restriction to classical context-free language, we cannot pair a helix with two others. However, using of a *multi-tape context-free grammar*, we can duplicate the helix sequence and perform two different and compatible pairings (see Fig. 4). Each helix is now represented twice; instead of making a "double pairing", we make two "simple pairings" of each of them. For example, the association of the ith helix with the $(i-1)$st and $(i+1)$st helices is described as the pairing of the second representation (duplicated helix) of the $(i-1)$st helix with the original ith helix *and* the pairing of the second representation of the ith helix with the original $(i+1)$st helix. Unfortunately, this model cannot represent the pair of extremal helices (the first and last helices closing the helix bundle. i.e., helix 1 and 4 in Fig. 4) because crossed pairings are not allowed, but this information will be computed in a last round by using grammar attributes.

3.4. Folding energy

Structural elements have been previously described; we now need to associate to each folding an energy that will allow to evaluate the structure quality and likelihood. This energy is computed from properties of each residue.

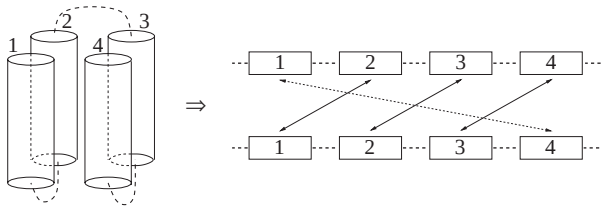


Fig. 4. Helix bundle. The additional pairing of helices 1 and 4 is represented in dashed line.

In this paper, the only property taken into account is hydrophobicity: thus, a hydrophobicity value is assigned to each amino acid. The folding energy is computed from these values as follows (see Fig. 5):

(1) E_{contact} is the energy induced by the helix association,
(2) E_{memb} is the energy induced by the interaction with the membrane,
(3) E_{turn} is the energy induced by the coil between two paired helices.

Let $\omega = \omega_1 \ldots \omega_n$ be the primary structure of a protein of size n and λ_i be the hydrophobicity value of the ith amino acid ω_i. Helix pairing imposes spatial proximity for the residues of the internal face I (see Fig. 3) which implies an interaction between them. The resulting energy is computed as the product of the hydrophobicity values of amino acids brought into contact by the folding. Formally, assume that the kth and $k+1$-st helices are paired; the resulting energy is

$$E_{\text{contact}} = \sum_{i=0}^{n} f(I_i^k, I_i^{k+1}),$$

$$\text{where } f(I_i^k, I_i^{k+1}) = \sum_{\omega_j \in I_i^k} \sum_{\omega_{j'} \in I_i^{k+1}} \frac{\lambda_j \cdot \lambda_{j'}}{\sqrt{\#I_i^k \cdot \#I_i^{k+1}}}. \tag{1}$$

Each amino acid which belongs to the external face of a transmembrane helix (called O) has an interaction with the membrane. Let the energy induced by the interaction of a residue ω_i be $\mathcal{K}\lambda_i$, where $\mathcal{K}$ is an experimental constant representing the hydrophobicity of the membrane environment. Obviously, as the membrane is a hydrophobic environment, this constant favors hydrophobic residues. Then

$$E_{\text{memb}} = \sum_{\omega_i \in O_i^k} \mathcal{K}\lambda_i. \tag{2}$$

The coil between two paired helices is exposed to the external environment of the membrane. Thus, we define a constant $\mathcal{C}$ modeling the hydrophilic constraint and similarly to the energy E_{memb}, the energy induced by the interaction of a residue ω_i with this environment is $\mathcal{C}\lambda_i$. In addition, the energy induced by the torsion applied to the chain must be considered. Then, we define a function $\mathcal{T}(l)$ where l is the number of residues of the inter-helix chain, which determines this energy: the coefficients of this function have been determined empirically from various experiments. Assume $\omega_m \ldots \omega_n$ is an inter-helix chain, then

$$E_{\text{turn}} = \mathcal{T}(n-m) + \sum_{i=m}^{n} \mathcal{C}\lambda_i. \tag{3}$$

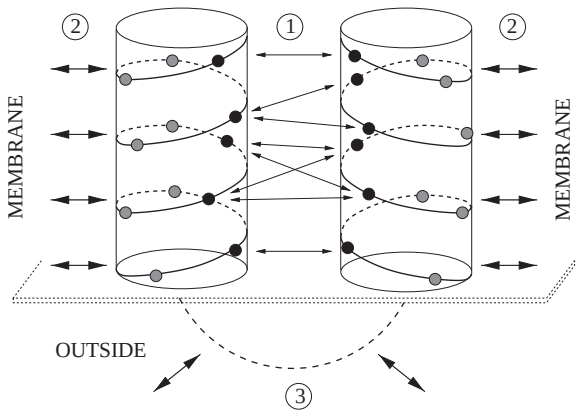


Fig. 5. The energy partition.

3.5. Additional restrictions

External constraints must be considered if we want to have realistic predictions. First, the membrane thickness imposes a minimal size to transmembrane helices. On the average, a helical turn has a length of 5.4 Å and a membrane a thickness of 4 nm; this implies that 7–8 turns are necessary for a helix to cross the membrane and thus a minimal number of 25 amino acids is required.

Moreover it has been shown by White and Wimley [27,57] that a minimal transfer energy value is required by a transmembrane α-helix for spanning a membrane. The authors did compute a hydrophobicity scale which can be used to determine if a polypeptide segment can form a transmembrane α-helix. This scale is integrated as a threshold function in our model, to select valid transmembrane segments which are likely to form an TM α-helix.

Finally, another constraint concerns the chain located between two consecutive paired helices. We want to limit the set of this configuration by allowing a pairing if and only if the inter-helix chain is flexible enough. We define a threshold $\mathcal{R}(n, E_{\text{turn}})$ which determines the minimal energy to fold a chain of size n.

4. Grammatical modeling

4.1. Rewriting rules

We now show how we can express the physical properties and constraints in a grammatical framework that was first introduced by Lefebvre and Steyaert [36,37]; a protein is now seen as a word on an alphabet Σ; this alphabet is composed of the 20 amino acids and so the primary structure of a protein is a *string* of letters over Σ. Grammatical rules express the geometrical constraints of the folding described in the previous section. The grammar is very ambiguous (i.e., there are several *derivation trees* for the same word or protein) since it

is designed to represent all possible foldings whose number is exponential in the sequence length. Each of them is associated with a unique *derivation tree* (i.e., a planar representation of the sequence of derivation rules) which is used to compute the folding energy. Thus, the predicted structure is the folding with the optimal energy, that is to say the derivation tree with the optimal attribute.

The type of grammar that we use is known as *multi-tape S-attributed grammars*. We will call them MTSAG in the following. Instead of giving a complete and technical description of them, we briefly describe the different steps of the construction.

In a first step, we design two grammars G_{single} and G_{couple} (see Fig. 6) which formalize the structural constraints respectively given in Sections 3.1 and 3.2. Then, we compose G_{couple} by G_{single} and obtain the grammar G_{pairing} (see Fig. 7) describing helix pairings. Let $Contact_x$ (resp. $Membrane_x$) be any non-terminal of the subset $\{Contact_P, Contact_L, Contact_R, Contact_B\}$ (resp. $\{Membrane_P, Membrane_L, Membrane_R, Membrane_B\}$). Then, the rewriting rules with $Contact_x$ as left-hand side produce amino acids of α-helices which are brought into contact by the pairing; those with $Membrane_x$ produce amino acids of α-helices exposed to the core of the membrane; the ones with *Turn* produce amino acids of a coil between two paired helices.

Up to this point, only classical context-free grammars have been used, but modeling helix bundle forces us to extend our framework to *multi-tape context-free grammars*. In order to achieve our goal, we need a *2-tape context-free grammar* to describe the two pairing sequences of 3.3. Let G_{bundle} be this grammar. Its description is given in Fig. 8. Terminals are now *2-tape* terminals and have two fields (one for each tape): as mentioned previously a 2-tape terminal is represented as a vector $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ where x and y can be any amino acid or the empty string ε.

Formally, the rewriting rules are quite similar to the ones of G_{pairing}. The main difference with these grammars is that G_{bundle} pairs amino acids from different tapes: $Contact_x$ and $Membrane_x$ associate residues of the first tape with residues of the second one. We use the following constraint on the *derivation tree* of a *2-tape alignment*: the *2-tape string* ω obtained after ε-deletion satisfies $\omega^{(1)} = \omega^{(2)}$ (i.e., identical strings on each tape, $\omega^{(i)}$ being the string on tape i of the m-tape string ω); this ensures that each amino acid is associated with the same α-helix on each tape (i.e., $\omega_i^{(1)}$ and $\omega_i^{(2)}$ belong to the same secondary structure: $\omega_i^{(k)}$ is the ith letter component of $\omega^{(k)}$).

The grammar G_{bundle} designed in Fig. 8 requires that the first and last helices of each bundle have exactly the same number of amino acids $\left(\text{see the rule } S_{\text{bundle}} \to \left[\begin{smallmatrix} \varepsilon \\ \bullet \end{smallmatrix}\right] S_{\text{bundle}} \left[\begin{smallmatrix} \bullet \\ \varepsilon \end{smallmatrix}\right]\right)$, but in practice the grammars implemented are more flexible: this constraint is not strictly respected. Moreover, the grammar rules cannot be directly used to model the "closing" helix pairing because this pairing could be parallel or anti-parallel: this feature depends of the odd (or even) number of helices in the bundle. Fortunately, this pairing can be modeled with a slight extension of the attribute system designed in the next section.

We show in Figs. 9 and 10 how [2] a 3 α-helix bundle can be described by a *derivation tree* of the grammar G_{bundle}. A schematic representation, with long-range interactions, of the C_α chain is given in Fig. 9, while its representation as a *derivation tree* is given in Fig. 10.

[2] Image obtained with PyMOL [13].

$$G_{\text{single}} = \begin{cases} S_{\text{single}} \to I_0 \mid I_1 \mid O_0 \mid O_1 \\ I_0 \to \bullet\, I_1 \mid \bullet\, O_0 \\ I_1 \to \bullet\, O_0 \mid \bullet\, O_1 \\ O_0 \to \bullet\, O_1 \\ O_1 \to \bullet\, I_0 \end{cases} \qquad G_{\text{couple}} = \begin{cases} S_{\text{couple}} \to \mathit{Face}_O \mid \mathit{Face}_I \\ \mathit{Face}_I \to I\ \mathit{Face}_O\ I \mid \mathit{Turn} \\ \mathit{Face}_O \to O\ \mathit{Face}_I\ O \mid \mathit{Turn} \\ \mathit{Turn} \to \bullet\, \mathit{Turn} \mid \bullet \end{cases}$$

Fig. 6. Grammatical modeling of the α-helix structure (G_{single}) and the pairing of two helices (G_{couple}). Nonterminal $\bullet$ stands for any amino acid while I and O mean any amino acid which belongs respectively to helical faces I and O.

$$G_{\mathit{pairing}} = \begin{cases} S_{\mathit{pairing}} \to \mathit{Contact}_P \mid \mathit{Membrane}_P \\ \mathit{Contact}_P \to \bullet\,\bullet\ \mathit{Membrane}_P\ \bullet\,\bullet \mid \bullet\ \mathit{Membrane}_L\ \bullet\,\bullet \mid \\ \qquad\qquad \bullet\,\bullet\ \mathit{Membrane}_R\ \bullet \mid \bullet\ \mathit{Membrane}_B\ \bullet \mid \mathit{Turn} \\ \mathit{Contact}_L \to \bullet\,\bullet\ \mathit{Membrane}_P\ \bullet\,\bullet \mid \bullet\,\bullet\ \mathit{Membrane}_R\ \bullet \mid \mathit{Turn} \\ \mathit{Contact}_R \to \bullet\,\bullet\ \mathit{Membrane}_P\ \bullet\,\bullet \mid \bullet\ \mathit{Membrane}_L\ \bullet\,\bullet \mid \mathit{Turn} \\ \mathit{Contact}_B \to \bullet\,\bullet\ \mathit{Membrane}_P\ \bullet\,\bullet \mid \mathit{Turn} \\ \mathit{Membrane}_P \to \bullet\,\bullet\ \mathit{Contact}_P\ \bullet\,\bullet \mid \bullet\ \mathit{Contact}_L\ \bullet\,\bullet \mid \\ \qquad\qquad \bullet\,\bullet\ \mathit{Contact}_R\ \bullet \mid \bullet\ \mathit{Contact}_B\ \bullet \mid \mathit{Turn} \\ \mathit{Membrane}_L \to \bullet\,\bullet\ \mathit{Contact}_P\ \bullet\,\bullet \mid \bullet\,\bullet\ \mathit{Contact}_R\ \bullet \mid \mathit{Turn} \\ \mathit{Membrane}_R \to \bullet\,\bullet\ \mathit{Contact}_P\ \bullet\,\bullet \mid \bullet\ \mathit{Contact}_L\ \bullet\,\bullet \mid \mathit{Turn} \\ \mathit{Membrane}_B \to \bullet\,\bullet\ \mathit{Contact}_P\ \bullet\,\bullet \mid \mathit{Turn} \\ \mathit{Turn} \to \bullet\, \mathit{Turn} \mid \bullet \end{cases}$$

Fig. 7. Grammatical modeling the pairing of two helices. Non-terminal $\bullet$ stands for any amino acid.

As mentioned previously, the "closing" pairing of the first and last helices is not described because it uses the attribute system.

Finally, the grammar G_α (see Fig. 11) describing the set of protein conformations of Section 3 is simply a concatenation of the grammars G_{single} for unpaired helix and G_{bundle} for helix bundles (including helix couples which are bundles with only two helices).

4.2. Attribute systems

Once this grammar has been written, we extend it to an S-attributed grammar by addition of attributes λ for storing the folding energy. Each rule of G_α is associated with a function allowing the energy computation of the induced structure. Then, the optimal structure for the thermodynamical model can be deduced from the derivation tree with the lowest attribute value, that is the folding energy (or any approximation of it).

The attribute functions compute the folding energy (see Eqs. (1)–(3) in Section 3.4) from equations given above. We give in Fig. 12 the set of the attribute functions allowing the recursive computation of the folding energy resulting from a helix pairing. The functions associated with the nonterminals $\mathit{Contact}_x$ as left-hand side compute the contact energy

$$
G_{bundle} = \begin{cases}
S_{bundle} \to \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] S_{bundle} \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] \mid Bundle \\
Bundle \to Contact_P\ Bundle \mid Membrane_P\ Bundle \mid Contact_p \mid Membrane_p \\
Contact_P \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_L \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \\
\qquad\qquad \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_R \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_B \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Contact_L \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_R \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Contact_R \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_L \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Contact_B \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Membrane_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Membrane_P \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_L \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \\
\qquad\qquad \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_R \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_B \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Membrane_L \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_R \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Membrane_R \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_L \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Membrane_B \to \left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\ \bullet\end{smallmatrix}\right] Contact_P \left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\ \varepsilon\end{smallmatrix}\right] \mid Turn \\
Turn \to \left[\begin{smallmatrix}\bullet\\ \bullet{=}1\end{smallmatrix}\right] Turn \mid \left[\begin{smallmatrix}\bullet\\ \bullet{=}1\end{smallmatrix}\right]
\end{cases}
$$

Fig. 8. Grammatical modeling of bundles: Non-terminal • stands for any amino acid and ε stands for the empty string. Notation $\bullet = 1$ means the same amino acid as on the first tape.

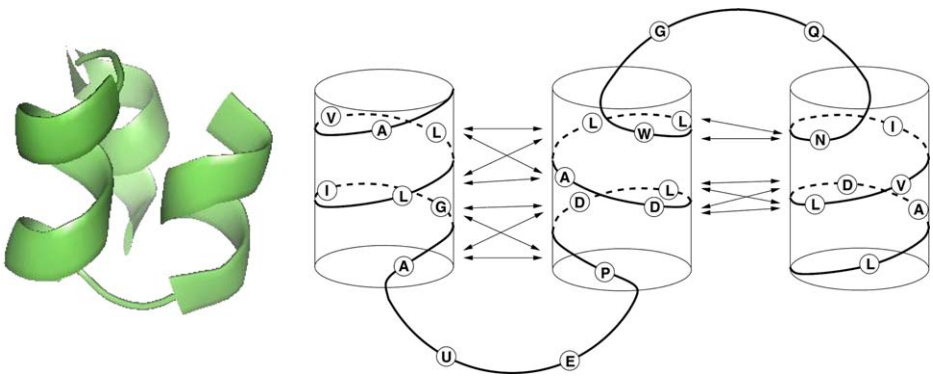


Fig. 9. A 3 α-helix bundle. (left) 3-D view of the C_α chain, (right) a schematic representation with long-range interactions.

E_{contact} resulting from the amino acids brought into contact by the pairing (see Eq. (1)). Those associated with $Membrane_x$ compute the energy E_{memb} resulting from the interaction of residues with the membrane (see Eq. (2)). And those associated with the non-terminal *Turn* compute the energy E_{turn} resulting from the interaction of the residues with the exterior environment (see Eq. (3)).

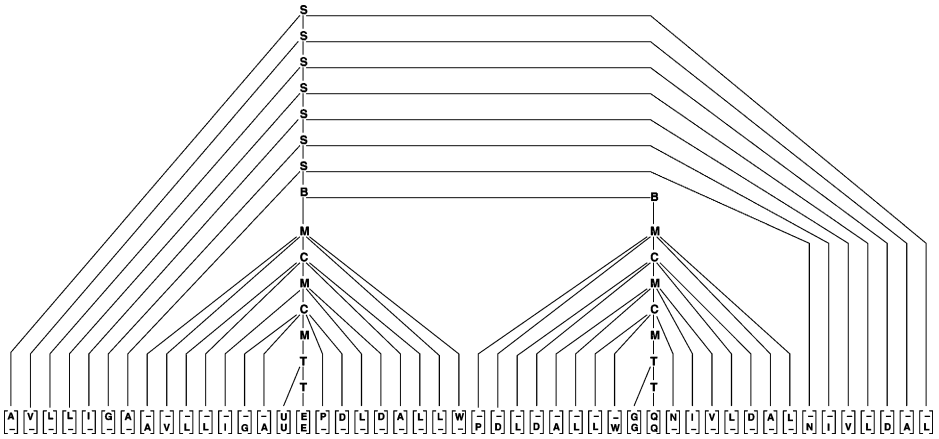


Fig. 10. Tree representation of a 3 α-helix bundle protein. **S** stands for non-terminal S_{bundle}, **B** for *Bundle*, **C** for $Contact_x$, **M** for $Membrane_x$ and **T** for *Turn*.

$$G_\alpha = \begin{cases} S_\alpha \rightarrow S_\alpha \; Motif \; Coil \mid Coil \\ Motif \rightarrow S_{\text{bundle}} \mid S_{\text{single}} \\ Coil \rightarrow \left[\begin{smallmatrix} \bullet \\ \bullet \!=\! 1 \end{smallmatrix} \right] \; Coil \mid \left[\begin{smallmatrix} \bullet \\ \bullet \!=\! 1 \end{smallmatrix} \right] \end{cases}$$

Fig. 11. The general grammar for all-α transmembrane proteins. Nonterminals S_{bundle} and S_{single} are the axioms of the grammars G_{bundle} (see Fig. 8) and G_{single} (see Fig. 6).

The set of attribute functions for G_{bundle} is very similar to F_{pairing} and the other ones needed to complete the extension of G_α into a S-attributed grammar are quite natural (Fig. 12).

5. Results and discussion

5.1. Some technical features of TMMTSAG

Given the S-attributed grammar described in the previous section, we use a parsing algorithm (called GCP) to find the optimal derivation tree (the optimal structure) of a given input sequence (the protein primary structure) for this grammar. A basic version of this algorithm has been described and implemented by Lefebvre [36,37] in a software called *mtsag2c*. We modified this algorithm and ran it on a sample of all-α membrane protein sequences in order to find their optimal derivation trees for G_α, that is to say the best folding predicted by our model.

$$F_{\text{pairing}} = \begin{cases} f_{S_{pairing} \to Contact_P | Membrane_P}(\lambda_1) = \lambda_1 \\ f_{Contact_x \to \bullet\bullet Membrane_P \bullet\bullet}(\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5) = \frac{(\lambda_1+\lambda_2)\cdot(\lambda_4+\lambda_5)}{2} + \lambda_3 \\ f_{Contact_x \to \bullet Membrane_L \bullet\bullet}(\lambda_1\lambda_2\lambda_3\lambda_4) = \frac{\lambda_1\cdot(\lambda_3+\lambda_4)}{\sqrt{2}} + \lambda_2 \\ f_{Contact_x \to \bullet\bullet Membrane_R \bullet}(\lambda_1\lambda_2\lambda_3\lambda_4) = \frac{(\lambda_1+\lambda_2)\cdot\lambda_4}{\sqrt{2}} + \lambda_3 \\ f_{Contact_x \to \bullet Membrane_B \bullet}(\lambda_1\lambda_2\lambda_3) = \lambda_1 \cdot \lambda_3 + \lambda_2 \\ f_{Contact_x \to Turn}(\lambda_1) = \mathcal{T}(Turn.length) + \lambda_1 \\ f_{Membrane_x \to \bullet\bullet Contact_P \bullet\bullet}(\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5) = \mathcal{K} \cdot (\lambda_1 + \lambda_2 + \lambda_4 + \lambda_5) + \lambda_3 \\ f_{Membrane_x \to \bullet Contact_L \bullet\bullet}(\lambda_1\lambda_2\lambda_3\lambda_4) = \mathcal{K} \cdot (\lambda_1 + \lambda_3 + \lambda_4) + \lambda_2 \\ f_{Membrane_x \to \bullet\bullet Contact_R \bullet}(\lambda_1\lambda_2\lambda_3\lambda_4) = \mathcal{K} \cdot (\lambda_1 + \lambda_2 + \lambda_4) + \lambda_3 \\ f_{Membrane_x \to \bullet Contact_B \bullet}(\lambda_1\lambda_2\lambda_3) = \mathcal{K} \cdot (\lambda_1 + \lambda_3) + \lambda_2 \\ f_{Membrane_x \to Turn}(\lambda_1) = \mathcal{T}(Turn.length) + \lambda_1 \\ f_{Turn \to \bullet Turn}(\lambda_1\lambda_2) = \mathcal{C} \cdot \lambda_1 + \lambda_2 \\ f_{Turn \to \bullet}(\lambda_1) = \mathcal{C} \cdot \lambda_1 \end{cases}$$

Fig. 12. The attribute system associated with the grammar G_{pairing}. Attributes λ_i compute the energy of the folding and the notation Turn.length gives the size of the substring generated by the nonterminal Turn.

A theoretical analysis of this algorithm [37,38] gives in the general case a complexity which is the product of the complexity of the parser for each projected grammar (one for each tape): that is, $\mathcal{O}(n^3) \cdot \mathcal{O}(n^3) = \mathcal{O}(n^6)$ in time and $\mathcal{O}(n^2) \cdot \mathcal{O}(n^2) = \mathcal{O}(n^4)$ in space, where n is the length of the input string. Fortunately, the projected grammars of G_α are strongly correlated (i.e., the parsing on the first tape is strongly correlated to that of the second tape) and in our case the experimental complexities are $\mathcal{O}(n^3)$ in time and $\mathcal{O}(n^2)$ in space.

We did experimentations on a Dec-Alpha with a 666 MHz bi-processor and 4 Gb Ram. Approximately 2 min of CPU-time and 1 Gb are required to parse protein sequences of 250 residue average length when turns are limited to 100 residues.

Moreover it is well-known that in vivo folding is not always the folding with the optimal folding energy. However, the folding energy of the "real" protein structure is most of the time close to the optimal one and thus called a *sub-optimal* folding energy. Hence, the prediction accuracy of our method can be sensibly increased if instead of giving only the optimal structure (structure with the optimal folding energy) we give as an output a set of sub-optimal structures, which can be done easily by moduling the constraints.

An optimized method has been designed: instead of using a single set of parameters to determine the optimal folding, we compute the optimal folding for eight different set of parameters (with some slight modifications in turn and/or helix constraints). Experimentally this results in two to four different topologies.

We denote by TMMTSAG-basic the structure prediction software which computes the optimal protein structure and by TMMTSAG-opt the structure prediction software which compute a subset of 8 sub-optimal structures. In Section 5.4, we report the results of TMMTSAG-opt obtained as the best prediction in the set of sub-optimal structures. In practice, an even realistic prediction can be deduced from the alignment of the eight predicted structures but this step is not yet fully automated.

Table 1
Training set for the determination of parameters

PDB	Protein	Chain
1H2S	Transducer of sensory rhodopsin II	B
1B9U	Subunit B of ATP synthase	A
1BL8	Potassium channel	A
1DXZ	α-Subunit of nicotinic acetylcholine receptor	A
1EQ7	Core structure of the outer membrane lipoprotein	A
1EZV	Yeast cytochrome Bc1 complex	H
1FJK	Phospholamban	A
1JB0	Photosynthetic reaction center and core antenna system	I,J,M,X
1KZU	Integral membrane peripheral light harvesting complex	A,B
1LGH	Light-harvesting complex II	A,B
1OCC	Bovine heart cytochrome C oxidase	E,J,K,L,M

5.2. Parameters

We have tested different hydrophobicity scales found in [9]; the predictions shown in the following use only those recommended by Cornette et al. (called PRIFT in the paper), which clearly give the best results.

Parameters $\mathcal{K}$ and $\mathcal{C}$ of the energy functions (cf. 3.4 and 3.5) have been determined experimentally with the help of a data set of 20 small membrane protein sequences (with at most 100 amino acids) with known structure (see Table 1). All parameters have been set to 0 initially, and then adjusted independently in order to satisfy the maximum likelihood of our predictions. By "maximum likelihood", we mean the best prediction rates with a correct topology prediction (i.e., each real helix is predicted and all helix pairs are found). Once these values are fixed, we repeat the process with these new parameters as default settings until numerical values became stable. During this learning process, the minimal length of a helix has been fixed to 10 residues in order to allow a large number of potential topologies.

Finally, a minimal length of 3 amino acids is required to form a coil between two paired helices, and each helix has a minimal length of 20 amino acids.

5.3. Methods

We evaluated our model on three different sets of protein sequences. The first one is composed with TM protein sequences whose 3-D structure is known at a high-resolution level. A complete list of them can be found at Stephen White Laboratory's home page: `http://blanco.biomol.uci.edu`. The second one is composed with TM protein sequences whose structure is known at a low-resolution level and finally, the third one is a set of globular protein sequences used to estimate the accuracy of distinguishing TM proteins from non-membrane proteins.

Chen et al. [5] propose a standardized benchmark for evaluating new methods predicting TM helices available at `http://cubic.bioc.columbia.ed u/services/tmh_benchmark`. Our predictions do not fit exactly with those examined in the paper: the reason is that we focus on the prediction of all-α TM channels (TM helices with their interactions) instead of TM helices only. It follows that the direct application of this protocol is not representative of the real performances of our algorithm. However, we can use it by eliminating sequences containing only one TM helix.

We have then selected all TM protein sequences proposed in [5] with at least 2 TM helices and with a length from 80 to 600 amino acids, corresponding to 11 TM protein sequences at high-resolution level and 82 TM protein sequences at low-resolution level. The upper bound of 600 residues is needed to prevent the explosion of the computational time of our algorithm. Nevertheless, most of the time we do not reach this bound: 90% of the TM protein sequences which are not suited to our criteria, are rejected because of their small length (< 80 residues). The high-resolution dataset has also been updated according to the up-to-date list of the White Laboratory's home page. So, 16 additional TM protein sequences have been found with respect to our constraints (2 TM helices at least and with a length from 80 up to 600 amino acids). The high-resolution dataset is thus composed with 28 TM protein sequences. Finally, the non-membrane protein sequence dataset is that used by Krogh et al. [34] for training TMHMM. This set has been extracted from the Protein Data Bank and homologous sequences removed as described by Lung et al. [39]. Here also, only sequences with at most 600 residues have been selected, which results into a dataset of 567 sequences.

We computed the scores defined by Chen et al. [5] for all these datasets, and compared the results with those obtained with seven other softwares (HMM-TOP2, MEMSAT, PHDpsi-htm, PRED-TMR, SOSUI, TMHMM and TOPPRED2) on the same datasets. However, we used a slightly different, but more restrictive, definition of a *correctly predicted* TM helix: an observed TM helix *intersects* a predicted TM helix if and only if the overlap contains at least seven residues and a TM helix is said *correctly predicted* if and only if the observed (resp. predicted) helix intersects with one and only one predicted (resp. observed) helix. We say also that a protein structure is *correctly predicted* if and only if each observed (resp. predicted) TM helix intersects with one and only one predicted (resp. observed) helix; it is said *almost predicted* if and only if all observed (resp. predicted) TM helices intersect with at most one predicted (resp. observed) helix. Then, we naturally define the following scores:

$$Q_{\text{htm}}^{\%\text{obs}} = 100\frac{\text{Number of correctly predicted TM helices}}{\text{Number of TM helices observed}},$$

$$Q_{\text{htm}}^{\%\text{pred}} = 100\frac{\text{Number of correctly predicted TM helices}}{\text{Number of TM helices predicted}},$$

$$Q_{\text{ok}} = 100\frac{\text{Number of correctly predicted protein structures}}{\text{Number of proteins}}.$$

The scores measuring the per-residue accuracy are similar to those defined by Chen et al. [5]. Each amino acid in the sequence has a secondary structure assignment (in a TM helix or in a non-membrane region) given experimentally (observed) or by predicting methods

(predicted). A residue is said correctly predicted when its observed and predicted secondary structure labels are the same.

$$Q_2 = 100\frac{\text{Number of residues correctly predicted in the protein}}{\text{Number of residues in the protein}},$$

$$Q_{2T}^{\%\text{obs}} = 100\frac{\text{Number of residues correctly predicted in TM helices}}{\text{Number of residues observed in TM helices}},$$

$$Q_{2T}^{\%\text{pred}} = 100\frac{\text{Number of residues correctly predicted in TM helices}}{\text{Number of residues predicted in TM helices}},$$

$Q_{2N}^{\%\text{obs}}$ and $Q_{2N}^{\%\text{pred}}$ are the corresponding percentage for non-membrane residues.

Finally the accuracy of distinguishing transmembrane from globular proteins is simply evaluated by giving the percentage of false positives (the percentage of globular proteins predicted as TM proteins) and the percentage of false negatives (the percentage of TM proteins predicted as non-membrane proteins). Obviously, the values of the percentage of false negatives have been computed independently for the high- and low-resolution datasets.

An example of the output of our software is given in Fig. 13. Line 1 displays the amino acid sequence, line 2 recalls the experimental structure given in the **PDB** and lines 3 and 4 display the prediction made by our modeling and our algorithm. While line 2 uses the standard secondary structure notation of PDB (**H** for a residue which belongs to a helix, **S** to a bend, **T** to a turn and **E** to a strand), our prediction uses another syntax. As shown in Fig. 4, since each helix of a bundle is paired with its two neighboring sequences, we need to represent it twice. Thus, the *i*th helix of tape 2 (line 4) is coupled with the $(i+1)$-st helix of tape 1 (line 3) and the last helix of tape 2 is coupled with the first helix of tape 1. Obviously, when the bundle contains only two helices, which is the case for the subunit C of the f1fO atpase (1A91), the super-secondary structure is a simple helix pairing and we do not need two tapes to represent it. Finally, in a transmembrane helix, there are two kinds of residues: those paired with one or two residues of a neighboring helix (noted **P**), and those exposed to the core of the membrane (noted **M**). Thus, it is possible to predict which residues are in contact. Assume that the *i*th helix on tape 2 is paired with the $i+1$-th helix on tape 1. The pairing mechanism is similar to a well-bracketed expression: the last **P**-face (the one or two consecutive residues labeled with **P**) of the *i*th helix on tape 2 is paired with the first **P**-face of the $(i+1)$th helix on tape 1, the **P**-face before the last one of the *i*th helix with the second **P**-face of the $(i+1)$th helix...

5.4. Results

5.4.1. Results on the high resolution dataset

The evaluation of the secondary structure made by TMMTSAG is shown in Table 2. The predictions of the two methods (TMMTSAG-basic and TMMTSAG-opt. cf. 5.1) are compared to those obtained on the same sequences with seven other advanced methods: HMMTOP2, MEMSAT, PHD, PREDTMR, SOSUI, TMHMM, TOPPRED2.

As we mentioned in the introduction, usual methods, due to their natures, can hardly cope with long-range interactions since they are based on the local information contained in the

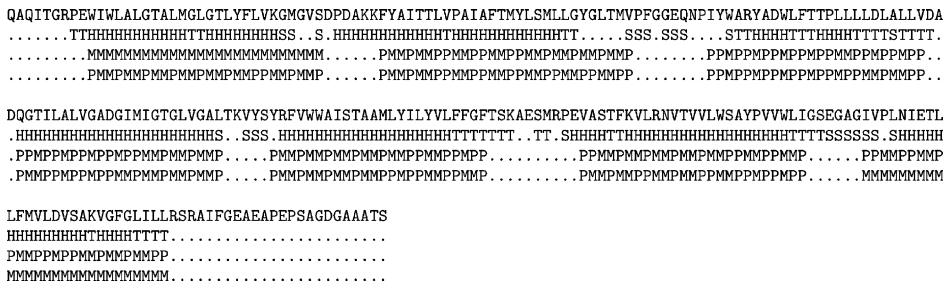


Fig. 13. An example of prediction: bacteriorhodopsin (1AP9).

Table 2
Evaluation of secondary structure prediction on the high-resolution dataset; scores inside parentheses concern almost predicted structures

Method	Per-segment accuracy			Per-residue accuracy				
	Q_{ok}	$Q_{\text{htm}}^{\%\text{obs}}$	$Q_{\text{htm}}^{\%\text{pred}}$	Q_2	$Q_{2T}^{\%\text{obs}}$	$Q_{2T}^{\%\text{pred}}$	$Q_{2N}^{\%\text{obs}}$	$Q_{2N}^{\%\text{pred}}$
TMMTSAG-opt	93.10(100.00)	99.35	98.72	80.13	89.22	80.41	65.85	79.55
TMMTSAG-basic	75.00(92.86)	96.48	93.20	78.27	87.59	78.79	64.08	77.21
HMMTOP2	71.43(100.00)	97.18	96.50	79.47	71.81	92.51	91.15	67.97
MEMSAT	64.29(96.43)	94.37	96.40	76.50	67.43	91.53	90.42	64.41
PHD	75.00(96.43)	96.48	97.16	77.78	66.74	94.96	94.60	65.12
PREDTMR	53.57(100.00)	88.73	100.00	74.43	59.12	97.52	97.71	61.11
SOSUI	81.48(100.00)	97.06	99.25	80.31	72.25	93.88	92.74	68.43
TMHMM	75.00(100.00)	97.18	97.87	79.83	72.78	92.16	90.56	68.59
TOPPRED2	71.43(96.43)	96.48	95.80	75.77	66.36	91.05	90.07	63.77

protein sequence. Furthermore we have a clearer description of the overall topology since our model describes also the helix associations.

Residue contact predictions have been evaluated by computing the average distance between C_α's in the structure given by the PDB. Since the definition of a contact between two residues has not yet been fixed in the literature (a brief overview of them can be found in [20]), two measures have been computed: the first one is the mean value μ_c and the standard deviation δ_c of *face distances* (the distance between two faces is defined as the minimum distance of two C_α belonging to each face). The second one is, given a threshold value $d_{\max}$, the percentage of residues which are distant from at most $d_{\max}$. In this paper we have used four different values for $d_{\max}$ (9, 11, 13 and 16 Å).

In addition, we note that a shift of approximately 1 turn (3 or 4 amino acids) sensibly increases the reliability of residue contact predictions of some proteins (1BL8-A, 1EHK, 1F88, 1FQY, 1J4N, 1KQF-C, 1L0V-C, 1L0V-D, 1NEK-D, 1OCC-A, 1OED-A, 1OED-C, 1Q16-C and 1QLA-C). It is also noticeable that the standard deviation of the mean value of face distances is strongly correlated to the quality of the secondary structure prediction (Table 3).

Table 3
Evaluation of residue contact predictions on the high-resolution dataset

PDB	Face distance		Residue contact (%)			
	μ_c	σ_c	< 9 Å	< 11 Å	< 13 Å	< 16 Å
1A91	8.82	1.89	40	80	100	100
1AP9	11.28	3.52	22	54	70	91
1BL8-A	14.02	2.90	9	18	36	63
1E12	10.66	2.42	16	58	86	96
1EHK	17.97	9.82	7	17	35	58
1F88	16.28	5.83	13	24	33	43
1FQY	16.14	5.00	12	20	33	47
1FX8	19.36	10.76	27	31	37	42
1H2S-A	10.21	3.00	36	59	82	97
1J4N	16.06	5.80	10	14	38	60
1JB0-L	17.76	6.39	4	18	27	45
1KQF-C	22.94	6.86	6	6	9	15
1L0V-C	16.17	5.56	5	22	33	50
1L0V-D	12.47	4.10	25	40	50	80
1MSL	12.36	3.34	0	50	70	90
1MXM-A	14.40	5.52	11	44	50	61
1NEK-C	9.99	3.22	35	68	73	100
1NEK-D	13.15	3.14	10	20	45	75
1OCC-A	12.96	3.65	16	25	42	82
1OCC-C	14.79	6.55	14	28	48	67
1OED-A	11.74	3.46	15	53	65	96
1OED-B	10.46	1.59	25	62	95	100
1OED-C	12.56	3.20	24	41	68	79
1PRC-L	18.80	10.22	13	22	28	40
1Q16-C	21.08	7.35	5	10	10	35
1Q90-B	16.43	5.75	6	13	33	56
1Q90-D	22.47	13.59	13	17	26	43
1QLA-C	16.28	7.54	14	24	41	68

Table 4
Evaluation of secondary structure prediction on the low-resolution dataset; scores inside parentheses concern almost predicted structures

Method	Per-segment accuracy			Per-residue accuracy				
	Q_{ok}	$Q_{\text{htm}}^{\%\text{obs}}$	$Q_{\text{htm}}^{\%\text{pred}}$	Q_2	$Q_{2T}^{\%\text{obs}}$	$Q_{2T}^{\%\text{pred}}$	$Q_{2N}^{\%\text{obs}}$	$Q_{2N}^{\%\text{pred}}$
TMMTSAG-opt	60.98(95.12)	95.70	91.42	75.25	90.42	65.79	63.47	89.51
TMMTSAG-basic	37.80(73.17)	87.11	81.68	69.30	87.06	60.31	55.50	84.67
HMMTOP2	60.98(86.59)	87.11	93.31	84.39	81.69	82.44	86.48	85.88
MEMSAT	54.88(92.68)	91.02	93.20	85.26	81.40	84.80	88.35	85.60
PHD	24.36(43.59)	60.04	72.02	83.99	91.08	76.60	78.51	91.93
PREDTMR	50.00(96.34)	88.09	96.57	85.19	75.86	88.63	92.44	83.14
SOSUI	48.78(90.24)	86.33	94.85	82.36	79.17	80.21	84.83	83.98
TMHMM	69.51(89.02)	90.23	95.45	85.32	83.02	83.34	87.11	86.85
TOPPRED2	53.66(86.59)	83.59	95.75	83.46	74.48	85.81	90.43	82.02

5.4.2. Results on the low-resolution dataset

Since the 3-D atom coordinates are not available for proteins whose structures have been determined at a low-resolution level, we are not able to evaluate residue contact predictions. Thus, only the evaluation of the secondary structure predictions is performed in Table 4.

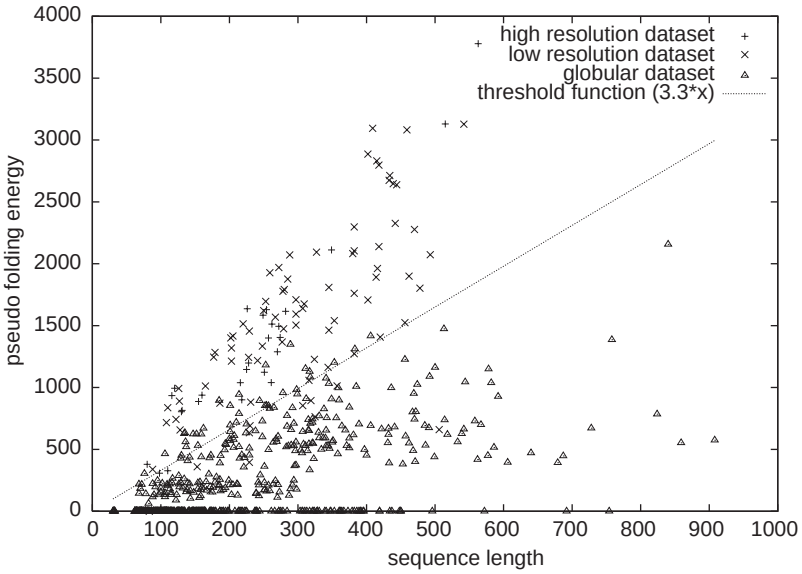


Fig. 14. Folding energy value vs. sequence length. Protein sequences of the high-resolution dataset are plotted with $+$. Protein sequences of the low-resolution dataset are plotted with $\times$. And protein sequences of the non-membrane dataset are plotted with $\triangle$. The plane is partitioned with the hyper-plane $y = 3.3 \times x$, exhibiting the difference of TM vs. globular folding energy values.

5.4.3. Distinguishing membrane protein from other proteins : a study of folding energy

We have plotted the folding energies of proteins of the globular protein sequences dataset with those of high- and low-resolution TM protein sequences in Fig. 14. A clear partition is observed between the two types of proteins. By cutting the plane with the hyper-plane $y = 3.3 \times x$ (where x represents the length of the sequence and y the folding energy), we have a criterion for distinguishing membrane proteins from other proteins: protein sequences of length n with an optimal folding energy lower than the threshold $\mathcal{M}(n) = 3.3 \times n$ are classified as non-membrane and those above are declared to be likely TM.

Our globular protein dataset contains 567 sequences proposed by Krogh et al. [34] for training TMHMM1 (those with a length up to 600). In comparison, we give the results obtained with the other softwares on a different dataset of 616 globular protein proposed by Chen et al. [5]. Since the method used for selecting the sequences (after removing homologous ones) in these two datasets as well as their sizes (approximately 600 sequences) are similar, we conclude that the values are comparable (Table 5).

5.5. Discussion

At this stage, the performance of our model is at least as good as those of the others. By combining the local and the global descriptions of the structure in the same formalism, which cannot be done neither by Hidden Markov Model nor by Neural Networks, we can consider simultaneously the local ability of an amino acid to fit into a structure (hydrophobicity) as well as long-range interactions appearing during secondary structure associations (residue

Table 5
Comparison of accuracy to distinguishing membrane protein from other proteins. The dataset used to evaluate TMMTSAG is not exactly the same than those used for the other softwares

Method	False positives (%)	False negatives (%)	
		High-resolution	Low-resolution
TMMTSAG	5.6	7.1	14.6
HMMTOP2	6	0	1
PHD	2	3	8
PREDTMR	4	8	1
SOSUI	1	8	4
TMHMM	1	8	4
TOPPRED2	10	8	11

contacts). Moreover, more information on the folding can be more easily added to the model than in other prediction methods. Residue contacts can be predicted with a relatively good accuracy even when the average distance between predicted neighbor amino acids is significantly higher than the expected one: in fact our method has not been designed to take this parameter as the major factor of optimization. In the following, we must keep in memory that most sequences in the datasets have been used for training other methods (see [5]). Thus results obtained with these software must be considered as upper bounds of their accuracies. We insist again on the fact that our model does not use classical learning methodology, and that the various parameters have been determined independently of the sequence sets used for validation.

5.5.1. High resolution dataset

The accuracy of our predictions is closely related to the protein topology: the more the 3-D structure fits into the grammar, the better the prediction rates are. This phenomenon can be observed on the Rhodopsin-like family (BacterioRhodopsin, Sensory Rhodopsin and HaloRhodopsin). Anyhow, the extended MTSAG describing a larger class of topologies should improve the prediction reliability.

Some inexact results come from a shift between pairing helices. Our model only considers "perfect" helix associations where the contact axis (the one containing the C_α's of coupled amino acids) is orthogonal to the helix axis. However, in real foldings, this property does not hold in general because helix axes are curved and residues are not so perfectly placed opposite to each other. In fact our model has not been designed to deal with features resulting from constraints defined at an atomistic level. Assuming the folding is determined primarily by hydrophobic forces [44,27,57] and long-range phenomena such as residue and/or helix interactions [20,12], we focus on the determination of the overall structure (the TM α-helices and their arrangement). Features enumerated below are due to a local optimization of the structure. Our algorithm is designed to propose a template (or a family of templates) of TM α-channels which has to be refined by purely physical models.

One could imagine that after this rough pairing, two helices could be slightly shifted because of their side-chain volumes. The addition of new parameters (side-chain volume for example) as attributes should solve these problems. Such refinements naturally fit into our formalism.

An important result is that the overall structure is more reliably predicted than with other methods, assuming that the structure is described by the grammar. As this phenomenon is mainly due to long-range interaction, it gives a strong argument to validate our approach. One should note that when some structural motif which appears is not described by the grammar, the overall topology cannot be reliably predicted and the per-residue accuracy falls drastically, although an important part of its features can be retrieved. We have observed that the presence of a "long" isolated helix (seven residues at least) between two paired ones strongly disturbs the prediction. Even when the correct topology is found, the helix prediction is damaged: helix lengths are stretched in order to "fill" the subchain left empty by the lack of the isolated helix. This feature could in some cases explain the drop in the per-residue specificity rating $Q_{2T}^{\%\text{pred}}$. However, this drawback should be corrected by integrating a Hidden Markov Model in the model or some other parameters allowing the evaluation of the helix stability [6,14–16,32,45,58], which would improve predictions.

In a general way, per-residue prediction rates are very satisfying and make a good trade-off between sensitivity ($Q_{2T}^{\%\text{obs}}$) and specificity ($Q_{2T}^{\%\text{pred}}$) of TM regions. If specificity is slightly lower than for other methods, on the other hand sensitivity is clearly increased. Fortunately, specificity rate should be improved by the extension of the grammar to the description of other topologies and the natural integration of a Hidden Markov Model and other physico-chemical parameters in the model.

Finally one can consider that the contact predictions are good for such a basic model.

5.5.2. Low resolution dataset

As was noted by Chen et al. [5], "prediction methods not significantly less accurate than low-resolution experiments". This fact can be a first argument to explain the drop in accuracy of our predictions. Since other methods have been trained on most of these sequences and cross-validation data are not available, we estimate that with similar results our algorithm compares favorably with them.

Finally, the following fact could explain the drop of per-residue accuracy $Q_{2T}^{\%\text{pred}}$: the average TM helix length is greater in the high-resolution dataset (27 amino acids in average) than in the low-resolution one (21 residues in average). Since the minimal length allowed by the model to TM helices is still the same (20 residues) it is natural to observe a drop of this score. Empirically, the specificity rate $Q_{2T}^{\%\text{pred}}$ in the high-resolution dataset is approximately of 80% and the helices in low-resolution dataset are approximately 80% shorter. Then, we must expect a score of $100 \times 0.8 \times 0.8 = 64\%$ for $Q_{2T}^{\%\text{pred}}$ in the low-resolution dataset... which is the observed one.

5.5.3. Distinguishing TM from other proteins

Finally, we have established a reliable distinction between TM and non-membrane proteins which is done by using a threshold on the folding energy value (proteins with a folding energy below the threshold are classified as non-membrane proteins). This phenomenon strongly argues in favor of our definition of the folding energy. At this moment, our accuracy to distinguish TM from other proteins is similar to those obtained with other softwares, but the future refinements of the energy function will surely improve this criterion.

6. Conclusion and future work

We started this work by the study of all-α transmembrane proteins since their topology is simple. This model is the starting point of a family of more generalized models for protein structure prediction and will be refined in a close future.

Firstly, the set of predictable structures will be extended. Future models will include a grammatical description of β-sheets, amphipatic α-helix and their interactions. The models of helices pairing will be refined (see 5.5) in the local description of α-helices and amino acids belonging to the middle or helix caps will be differentiated. This improvement is closely related to the addition of new attributes, as charge or side-chain volume, which can describe the capacity of a residue to fit in such a specific position.

Existing methods can be integrated in our model in order to improve prediction rate: for example, scores given by Hidden Markov Model or by some other scale [6,14–16,32,45,58], that are useful to specify the capacity of a residue to belong to a given secondary structure, or Dayhoff matrices together with the AMSAG algorithm [55]. These improvements are in progress.

Acknowledgements

We thank Francois André, from the CEA-Saclay (France), for his helpful remarks and instructions during the biological evaluation of the model.

References

[1] L. Adamian, J. Liang, Helix–helix and interfacial pairwise interactions of residues in membrane proteins, J. Molecular Biol. 311 (2001) 891–907.
[2] A. Aho, R. Sethi, J. Ullman, Compilers, Principles, Techniques, and Tools, Addison-Wesley, Reading, MA, 1988.
[3] R. Backofen, D. Gilbert, Bioinformatics and constraints, Constraints 6 (2–3) (2001) 141–156.
[4] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, P. Bourne, The protein data bank, Nucleic Acids Res. 28 (2000) 235–242, http://www.rcsb.org/pdb/.
[5] C. Chen, A. Kernytsky, B. Rost, Transmembrane helix predictions revisited, Protein Sci. 11 (2002) 2774–2791.
[6] H. Cid, M. Bunster, M. Canales, F. Gazitua, Hydrophobicity and structural classes in proteins, Protein Eng. 5 (1992) 373–375.
[7] M. Claros, G. von Heijne, Toppred 2: an improved software for membrane protein structure predictions, CABIOS 10 (1994) 685–686.
[8] C. Combet, C. Blanchet, C. Geourjon, G. Deléage, Nps@: Network protein sequence analysis, TIBS 25(3) (2000) 147–150, http://npsa-pbil.ibcp.fr.
[9] J.L. Cornette, C.K.B., M.H., S.J.L., B.J.A., D.C., Hydrophobicity scales and computational techniques for detecting amphipathic structures in proteins, J. Molecular Biol. (1987) 659–685.
[10] T.E. Creighton, Proteins (Structures and Molecular Properties), Freeman, New York, 1993.
[11] J. Dawson, R. Melnyk, C. Deber, D. Engelman, Sequence context strongly modulates association of polar residues in transmembrane helices, J. Molecular Biol. 331 (2003) 255–262.
[12] W. Degrado, H. Gratkowski, J. Lear, How do helix–helix interactions help determine the folds of membrane proteins? perspectives from the study of homo-oligomeric helical bundles, Protein Sci. 12 (2003) 647–665.
[13] W.L. Delano, The pymol molecular graphics system, http://www.pymol.org, deLano, Scientific LLC, San Carlos, CA, USA.

[14] D. Eisenberg, A. McLachlan, Solvation energy in protein folding and binding, Nature 319 (1986) 199–203.
[15] D. Eisenberg, R. Weiss, T. Terwilliger, The hydrophobic moment detects periodicity in protein hydrophobicity, Proc. Natl. Acad. Sci. 81 (1984) 140–144.
[16] D. Engelman, T. Steitz, A. Goldman, Identifying nonpolar transbilayer helices in amino acids sequences of membrane proteins, Annu. Rev. Biophys. Chem. 15 (1986) 321–353.
[17] G.D. Fasman (Ed.), Prediction of Protein Structure and the Principles of Protein Conformation, Plenum Press, New York, 1989.
[18] A. Fiser, R. Sanchez, F. Melo, A. Sali, Comparative protein structure modelling, in: O. Becker, A. Mackerell Jr., B. Roux, M. Watanabe (Eds.), Computational Biochemistry & Biophysics, Marcel Dekker, NY, 2001 (Chapter 14).
[19] D. Gilbert, D. Westhead, J. Viksna, J. Thornton, Topology-based protein structure comparison using a pattern discovery technique, J. Comput. Chem. 26 (1) (2001) 23–30.
[20] M.M. Gromiha, S. Selvaraj, Inter-residue interactions in protein folding and stability, Biophys. and Molecular Biol., in press.
[21] M. Gross, Linguistic analysis of protein folding, FEBS Lett. 390 (1996) 249–252.
[22] W. Gu, S. Rahi, V. Helms, Solvation free energies and transfer free energies for amino acids from hydrophobic solution to water solution from a very simple residue model, J. Phys. Chem. 108 (2004) 5806–5814.
[23] G. von Heijne, Membrane protein structure prediction: hydrophobicity analysis and the 'positive inside' rule, J. Mol. Biol. 225 (1992) 487–494.
[24] V. Helms, J. McCammon, Conformational transitions of proteins from atomistic simulations in: P. Deuflhard, J. Hermans, B. Leimkuehler, A. Mark, S. Reich, R. Skeel (Eds.), Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1998, pp. 66–77.
[25] T. Hirokawa, S. Boon-Chieng, S. Mitaku, Sosui: classification and secondary structure prediction system for membrane proteins, Bioinformatics 14 (1998) 378–379.
[26] B. Honig, Protein folding: from the levinthal paradox to structure prediction, JMB 293 (1999) 283–293.
[27] S. Jayasinghe, K. Hristova, S. White, Energetics, stability, and prediction of transmembrane helices, J. Molecular Biol. 312 (2001) 927–934.
[28] X. Jinbo, L. Ming, K. Dongsup, X. Ying, Raptor: optimal protein threading by linear programming, J. Bioinform. Comput. Biol. 1 (1) (2003) 95–117.
[29] D. Jones, W. Taylor, J.M. Thornton, A model recognition approach to the prediction of all-helical membrane protein structure and topology, Biochemistry 33 (1994) 3038–3049.
[30] H. Kamberaj, V. Helms, Monte-carlo simulation of biomolecular systems with biomcsim, Comput. Phys. Commun. 141 (3) (2001) 375–402.
[31] A. Kernytsky, B. Rost, Static benchmarking of membrane helix prediction, Nucleic Acids Res. 31 (13) (2003) 3642–3644.
[32] A. Kessel, N. Ben-Tal, Free energy determinants of peptide association with lipid bilayers, in: S. Simon, T. McIntosh (Eds.), Peptide-lipid Interactions, Academic press, New York, 2002, pp. 205–253.
[33] D. Knuth, Semantic of context-free languages, Mathematical Systems Theory 2 (1968) 127–145, correction: Mathematical Systems Theory 5 (1971) 95–96.
[34] A. Krogh, L.B., G. Von Heijne, E.L.L. Sonnhammer, Predicting transmembrane protein topology with a hidden markov model: application to complete genomes, J. Molecular Biol. 305 (2001) 567–580.
[35] A. Lee, Lipid-protein interactions in biological membranes: a structural perspective, Bioch. Biophys. Acta 1612 (2003) 1–40.
[36] F. Lefebvre, An optimized parsing algorithm well-suited to rna folding, in: A. Press (Ed.), Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology, 1995, pp. 222–230.
[37] F. Lefebvre, A grammar-based unification of several alignment and folding algorithms, in: A. Press (Ed.), Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, 1996, pp. 143–154.
[38] F. Lefebvre, Grammaires s-attribuées multi-bandes et applications à l'analyse automatique de séquences biologiques, Ph.D. Thesis, École Polytechnique, 1997.
[39] O. Lung, K. Frimand, J. Gorodkin, H. Bohr, J. Bohr, J. Hansen, S. Bruknar, Protein distant constraints predicted by neural networks and probality density functions, Protein Eng. 10 (1997) 1241–1248.
[40] A. MacKerell, Atomic models and force fields, in: O. Becker, A. Mackerell Jr., B. Roux, M. Watanabe (Eds.), Computational Biochemistry & Biophysics, Marcel Dekker, NY, 2001(Chapter 2).

[41] G. Mauri, G. Pavesi, A. Piccolboni, Approximation algorithms for protein folding prediction, in: Proc. Tenth Ann. ACM-SIAM Symposium on Discrete Algorithms, ACM Press, 1999, pp. 945–946.
[42] Y. Okomato, Protein folding simulations and structure predictions, Comp. Phys. Commun. 142 (2001) 55–63.
[43] C. Pasquier, S. Hamodrakas, An hierarchical artificial neural network system for the classification of transmembrane proteins, Protein Eng. 12 (8) (1999) 631–634.
[44] J.-L. Popot, D.M. Engelman, Helical membrane protein folding, stability, and evolution, Annu. Rev. Biochem. 69 (2000) 881–922.
[45] M. Prabhakaran, The distribution of physical, chemical and conformational properties signal and nascent peptides, Biochem. J. 269 (1990) 691–696.
[46] B. Rost, R. Casadio, P. Fariselli, Refining neural network predictions for helical transmembrane proteins by dynamic programming, in: A. Press (Ed.), Proc. Fourth Internat. Conf. on Intelligent Systems for Molecular Biology, 1996, pp. 192–200.
[47] B. Rost, R. Casadio, P. Fariselli, C. Sander, Transmembrane helices predicted at 95% accuracy, Protein Sci. 3 (4) (1995) 521–533.
[48] D. Sankoff, Silmutaneous solution of the rna folding, alignment and protosequence problems, SIAM J. Appl. Math. 45 (5) (1985) 810–825.
[49] D.B. Searls, The linguistics of dna, Amer. Sci. 80 (1992) 579–591.
[50] I. Simon, A. Fiser, A. Tusnady, Predicting protein conformation by statistical methods, Bioch. Biophys. Acta 1549 (2001) 123–136.
[51] E.L.L. Sonnhammer, G. Von Heijne, A. Krogh, A hidden markov model for predicting transmembrane helices in protein sequences, in: J.G. et al. (Eds.), Proc. Sixth Internat. Conf. on Intelligent Systems for Molecular Biology, AAAI Press, 1998, pp. 175–182.
[52] G. Tusnady, I. Simon, Principles governing amino acid composition of integral membrane proteins: applications to topology prediction, J. Molecular Biol. 283 (1998) 489–506.
[53] M. Ulmschneider, M. Sansom, Amino acid distributions in integral membrane protein structures, Bioch. Biophys. Acta 1512 (2001) 1–14.
[54] M. Vauchaussade de Chaumont, Nombre de strahler des arbres, langages algébriques et dénombrement de structures secondaires en biologie moléculaire, Master's Thesis, Université de Bordeaux I, 1985.
[55] J. Waldispühl, B. Behzadi, J.-M. Steyaert, An approximate matching algorithm for finding (sub-)optimal sequences in s-attributed grammars in: Proc. First European Conf. on Computational Biology, ECCB 2002, Vol. 18, Bioinformatics, Oxford University Press, 2002, pp. 250–259.
[56] S. White, W. Wimley, Hydrophobic interactions of peptides with membrane interfaces, Bioch. Biophys. Acta 1376 (1998) 339–352.
[57] S. White, W. Wimley, Membrane protein folding and stability: physical principles, Annu. Rev. Biophys. Biomol. Struct. 28 (1999) 319–365.
[58] R. Wolfenden, L. Andersson, P. Cullis, C. Southgate, Affinities of amino acid side chains for solvent water, Biochemistry 20 (1981) 849–855.
[59] A.-S. Yang, B. Honig, Sequence to structure alignment in comparative modeling using prism, Proteins: Struct. Funct. Genet.