Master research Internship

Bibliographic report

Embedding background knowledge in automata weights to discover new proteins of functional families

Domain: Bioinformatics

Supervisor:
François Coste

Dyliss

Author:
Manon Ruffini

Abstract: Proteins perform very important functions within organisms. Predicting these functions is a major problem in biology. To tackle this issue, predictive models of functional families have been developed, based on the sequences of amino acids that form the proteins. This bibliography presents the current state-of-the-art models, and how background knowledge can be used to enhance their estimation in case of small training sets.
The Dyliss team recently introduced a more expressive model, called Protomata, that better represents the families of proteins, and their possible sub-families. Although this model is efficient, no specific technique for the estimation of the parameters is available. The focus of the internship will be to elaborate a weighting strategy to fully benefit from the expressivity of this new model. Moreover, research has to be done on the significativity of scores, in order to predict well the family of a new and previously unseen protein sequence.
Keywords: proteins, pseudo-counts, Dirichlet mixture, automata

Contents

Introduction

Proteins perform a lot of actions within biological organisms, like transporting molecules, replicating DNA, catalysing metabolic reactions, or responding to stimuli. A protein is a chain of amino acids, that can be seen as a word over the alphabet of amino acids (twenty letters). The chain folds into a three dimensional structure, determining a function. Although identifying the exact function of a particular protein is a complicated task, it can be rather easily sequenced, and the number of available sequences has been growing exponentially. Therefore, a lot of research has been going on the inference of the function from the sequence.

Here, the task at hand is the following: we are given a database of proteins, where some sequences are annotated as belonging to a family or another. We want to predict the family of the proteins that are not annotated. To do so, we will train models of the families from the available sequences. State-of-the-art methods in bioinformatics proceed with identifying the conserved segments, shared by all the available sequences of the family, and modelling these segments. Then, the model is used to compute a score for any new sequence, that gives information about whether the corresponding protein belongs to the family or not. In some families though, some subfamilies might be observed, that do not share the same segments conservations. To tackle this issue, the team Dyliss recently introduced a more expressive model, called Protamata, that enables the representation of these disjunctions.

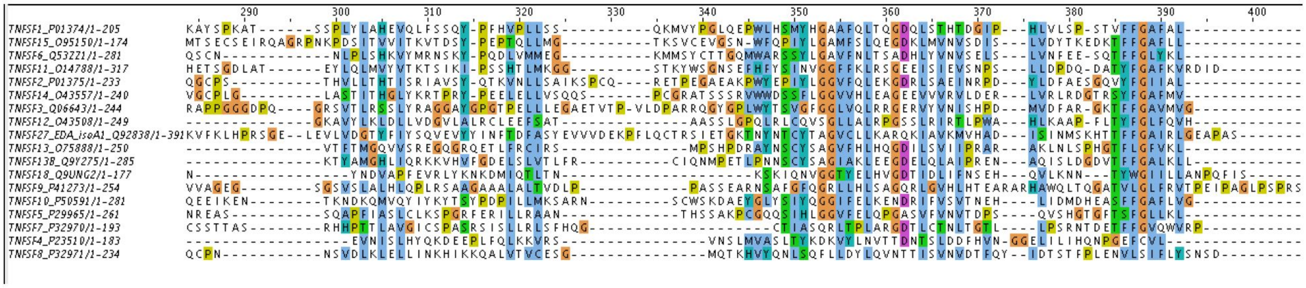
Furthermore, there are usually only a few annotated sequences, that can be used to design a model of the corresponding family. This lack of data can lead to a poor estimation of the parameters. On the other hand, a huge amount of data on protein sequences is available. Background knowledge can be extracted from this data. For example, which amino acids are more or less frequent, or which ones are more likely to be substituted, and why, are valuable information that can be used to compensate the lack of training data and enhance the estimation of the model. Indeed, if the training sequences present a lot of occurrences of an otherwise rare residue, it is likely to be a characterisation of the family, whereas if the residues are very common, they do not provide as much information.

Section 1 describes the state-of-the-art representations of families of proteins, and the current strategies to compensate the lack of annotated sequences. Then, in section 2, we will see a new approach, called Protomata, that was introduced by the Dyliss team, and discuss its possible improvements.

1 Position specific modelling

Given sequences from a family of related proteins, we aim at building a model of the family. We want this model to be explicit, so that experts might extract new knowledge from it, and to be able to find new members of the family, in order to label previously unseen proteins.

In subsection 1.1, we will introduce the sequence alignments and we will see how they highlight the conserved segments and the important positions within the set of sequences. Then, in 1.2, we will see different models that can be built from an alignment of the sequences. Finally, in subsection 1.3, we will present strategies used to improve the estimation of parameters.



```
TNFSF1_P01374/1-205            KAYSPKAT------SSPLYLAHEVQLFSSQY-PFHVPLLSS----------QKMVYPGLQEPWLHSMYHGAAFQLTQGDQLSTHTDGIP---HLVLSP-STVFFGAFAL-------------
TNFSF15_O95150/1-174           MTSECSEIRQAGRPNKPDSITVVITKVTDSY-PEPTQLLMG----------TKSVCEVGSN-WFQPIYLGAMFSLQEGDKLMVNVSDIS---LVDYTKEDKTFFGAFLL-------------
TNFSF6_Q53ZZ1/1-281            QSCN------------NLPLSHKVYMRNSKY-PQDLVMMEG----------KMMSYCTTGQMWARSSYLGAVFNLTSADHLYVNVSELS---LVNFEE-SQTFFGLYKL-------------
TNFSF11_O14788/1-317           HETSGDLAT------EYLQLMVYVTKTSIKI-PSSHTLMKGG---------STKYWSGNSEFHFYSINVGGFFKLRSGEEISIEVSNPS---LLDPDQ-DATYFGAFKVRDID---------
TNFSF2_P01375/1-233            QGCPS----------THVLLTHTISRIAVSY-QTKVNLLSAIKSPCQ----RETPEGAEAKPWYEPIYLGGVFQLEKGDRLSAEINRPD---YLDFAESGQVYFGIIAL-------------
TNFSF14_O43557/1-240           VGCPLG---------LASTITHGLYKRTPRY-PEELELLVSQQS-------PCGRATSSSRVWWDSSFLGGVVHLEAGEEVVVRVLDER---LVRLRDGTRSYFGAFMV-------------
TNFSF3_Q06643/1-244            RAPPGGGDPQ----GRSVTLRSSLYRAGGAYGPGTPELLLEGAETVTP-VLDPARRQGYGPLWYTSVGFGGLVQLRRGERVYVNISHPD---MVDFAR-GKTFFGAVMVG------------
TNFSF12_O43508/1-249           --------------GKAVYLKLDLLVDGVLALRCLEEFSAT--------------AASSLGPQLRLCQVSGLLALRPGSSLRIRTLPWA---HLKAAP-FLTYFGLFQVH------------
TNFSF27_EDA_isoA1_Q92838/1-391 KVFKLHPRSGE--LEVLVDGTYFIYSQVEVYYINFTDFASYEVVVDEKPFLQCTRSIETGKTNYNTCYTAGVCLLKARQKIAVKMVHAD---ISINMSKHTTFFGAIRLGEAPAS-------
TNFSF13_O75888/1-250           ---------------VTFTMGQVVSREGQGRQETLFRCIRS--------------MPSHPDRAYNSCYSAGVFHLHQGDILSVIIPRAR---AKLNLSPHGTFLGFVKL-------------
TNFSF13B_Q9Y275/1-285          ---------------KTYAMGHLIQRKKVHVFGDELSLVTLFR----------CIQNMPETLPNNSCYSAGIAKLEEGDELQLAIPREN---AQISLDGDVTFFGALKLL------------
TNFSF18_Q9UNG2/1-177           N-----------------YNDVAPFEVRLYKNKDMIQTLTN---------------------KSKIQNVGGTYELHVGDTIDLIFNSEH---QVLKNN---TYWGIILLANPQFIS------
TNFSF9_P41273/1-254            VVAGEG--------SGSVSLALHLQPLRSAAGAAALALTVDLP------------PASSEARNSAFGFQGRLLHLSAGQRLGVHLHTEARARHAWQLTQGATVLGLFRVTPEIPAGLPSPRS
TNFSF10_P50591/1-281           QEEIKEN--------TKNDKQMVQYIYKYTSYPDPILLMKSARN---------SCWSKDAEYGLYSIYQGGIFELKENDRIFVSVTNEH----LIDMDHEASFFGAFLVG------------
TNFSF5_P29965/1-261            NREAS----------SQAPFIASLCLKSPGRFERILLRAAN-------------THSSAKPCGQQSIHLGGVFELQPGASVFVNVTDPS----QVSHGTGFTSFGLLKL-------------
TNFSF7_P32970/1-193            CSSTTAS--------RHHPTTLAVGICSPASRSISLLRLSFHQG--------------------CTIASQRLTPLARGDTLCTNLTGTL----LPSRNTDETFFGVQWVRP-----------
TNFSF4_P23510/1-183            ------------------EVNISLHYQKDEEPLFQLKKVRS----------------VNSLMVASLTYKDKVYLNVTTDNTSLDDFHVN-GGELILIHQNPGEFCVL---------------
TNFSF8_P32971/1-234            QCPN-----------NSVDLKLELLINKHIKKQALVTVCESG---------------MQTKHVYQNLSQFLLDYLQVNTTISVNVDTFQY--IDTSTFPLENVLSIFLYSNSD---------
```

Figure 1: Sequence alignment of sequences of the family of TNF, reprinted from [Ker08]

1.1 Sequence alignment

Before going any further, we need to properly define the families of proteins that we will consider, as there are many possibilities of grouping proteins into families. Usually, we will consider functional (resp. structural) families, defined as families of proteins that share the same function (resp. structure). In bioinformatics, we also consider proteins deriving from a shared common ancestor, called homologous, that are likely to have kept the same function or structure due to natural selection. They are called equivalogous when they are homologous and have kept the same function through evolution. In this section, we will focus on families of homologous proteins.
A sequence alignment is based on a measure of similarity of the sequences, defined by a substitution matrix, defined below.

Substitution matrices Due to selection pressure, amino acids are more likely to be substituted by residues with close physical and chemical properties, that will not affect the structure or the function of the protein. Sequences that share common history can share very conserved segments in their sequences, from which we can deduce the probability of each amino acid to be substituted by another. These probabilities are used to compute a 20×20 substitution matrix, such as the BLOSUM62 matrix ([Cos16]). Such matrices provide measures of similarity between sequences: if two sequences have very substitutable amino acids in the same positions, they are similar.

Sequence alignment The proteins that we consider have a common evolution history, and hence their sequences share similarities, that can be harnessed to determine characterisations. Given a set of homologous protein sequences, we align the conserved positions of the sequences in columns, where each column is assigned a position in each sequence, ideally corresponding to the evolution of the position in the offspring sequence. Gaps can be introduced, in order to maintain the alignment, even if sequences present extra amino acids, due to insertions and deletions, that might have happened in the evolution process. The evolutionarily correct alignment is very difficult to infer, so it is approximated by the alignment that maximises the similarity score, computed from a substitution matrix ([DEKM98]). Programs that compute sequence alignments are available, like ClustalW ([THG94]). An example of a multiple alignment is given in figure 1. A sequence is said to be involved in a column if its corresponding aligned position presents an amino acid, and not a gap.

The longer the sequences, the harder it is to find an appropriate alignment. Mostly, sequences present very well conserved regions, that are important to the family, and unspecific chains of amino acids in between. Hence, instead of computing the global alignment with the whole of the sequences, algorithms like MEME or DIALIGN ([BWML06],[Cos16]) were developed to compute a local alignment, that only involves a small segment of the sequences. If several regions can be locally aligned, we will talk about a multiple local alignment.

1.2 From sequence alignment to family modelling

The alignment of sequences of homologous proteins shows the conservations in the family. Moreover, it can be used to build the following predictive models, that are employed to score previously unseen sequences and determine whether they belong to the family or not.

Position specific scoring matrix (PSSM) Position specific scoring matrices (PSSM) describe the successive positions of the sequences of the family ([Cos16]). Each column of the matrix corresponds to a column in the alignment, and each line to an amino acid. The coefficient of the line a and column i corresponds to the probability of finding amino acid a at the i-th position, that can be estimated by maximum likelihood. Let us denote $e_i(a)$ this probability, and $E_i(a)$ the number of occurrences of a at position i, then:

$$e_i(a) = \frac{E_i(a)}{\sum\limits_{a'=1}^{20} E_i(a')}$$

For a given new sequence, its positions are aligned with the columns of the PSSM, and the probabilities of each position are added, leading to an alignment score of the sequence. This score is then compared to a threshold, in order to decide of the belonging to the family. A sketch of the typical topology of a PSSM is shown in figure 2.
This topology is suited to represent very conserved, and rather small, regions with few insertions or deletions.

Hidden Markov model profiles (pHMM) In order to represent longer sequences, the model must be more flexible. Specifically, insertions and deletions must be taken into account to be in harmony with the evolution process.
Hidden Markov model profiles (pHMM) can be seen as PSSMs with possibilities of insertions and deletions added to each column ([Cos16]). Their topology is a left-to-right hidden Markov model (HMM) with three hidden states for each position: a state of match, that corresponds to the column in the PSSM, a state of insertion, and a state of deletion (figure 2). This model allows small and rare insertions and deletions, but also the insertion of greater segments of amino acids.
Once this hidden Markov model profile (pHMM) is computed, the best score of a new sequence can be computed, using the Viterbi algorithm for example, and compared to a threshold.

The construction of a pHMM usually starts from a multiple local alignment of the sequences. The columns of the PSSM that involve enough sequences, say more than half of the set for instance, are converted in match states, and associated with the corresponding insertion and deletion states.

The coefficients in the match states, called emission probabilities, can be computed as in the columns in the PSSMs. As far as transition probabilities are concerned, they are assigned as follows: for each pair of distinct states (k, l), let us denote the transition probability a_{kl} and frequency A_{kl}, by the application of the maximum likelihood principle, we get the following formula ([DEKM98]):

$$a_{kl} = \frac{A_{kl}}{\sum\limits_{l'} A_{kl'}}$$

The amino acids that are generated by the insertion states are not important for the characterisation of the family. However, their number is important, since it determines the relative position of conserved segments in the sequence. Hence, the quantity of inserted amino acids between two conserved segments, i.e. the number of loops on the corresponding insertion state, has to be controlled.

Let us consider an insertion state i, let g_i be the probability of a loop on i and μ_i be the average number of inserted amino acids (computed from the data). Let X be the random variable that represents the number of loops on i. Then, for all $n \in \mathbb{N}$:

$$\mathbb{P}(X = n) = g_i^n(1 - g_i) \text{ and } \mathbb{E}[X] = \frac{g_i}{1 - g_i}$$

The parameter g_i is then set, so that $\mathbb{E}[X] = \mu_i$:

$$g_i = \frac{\mu_i}{1 + \mu_i}$$

Meta-MEME Though pHMMs enable the representation of longer sequences, they induce the estimation of a lot of parameters. Moreover, in sequences, the succession of some positions in very conserved regions might be important, and no insertion or deletion should be allowed. For these reasons, pHMMs can be simplified to Meta-MEME models ([GBEB97]): Insertion and deletion states are removed, some of the match states are kept and grouped in blocs, that correspond to the most characteristic segments of the family. These blocs are separated with gaps, that enable the insertion of amino acids (figure 2). Blocs can be assimilated to PSSMs, and gaps to insertion states. Parameters are estimated as before.

1.3 Enhancement of the estimation of parameters

We have described several possibilities of representations of families of proteins. Doing so, we limited ourselves to the maximum likelihood principle in the estimation of the emission and transition probabilities. However, this approximation can be greatly improved. Firstly, we must consider the fact that the training set of sequences might be biased and not representative of all the family. Secondly, as only few sequences are available for the construction of the model, background knowledge needs to be integrated to avoid overspecialisation.

Weights on the sequences The training set of sequences does not necessarily represent well the family of proteins. If a sub-family is over-represented, it might lead to a model closer to this particular subset, rather than a model of the whole family. To avoid this overspecialisation, higher

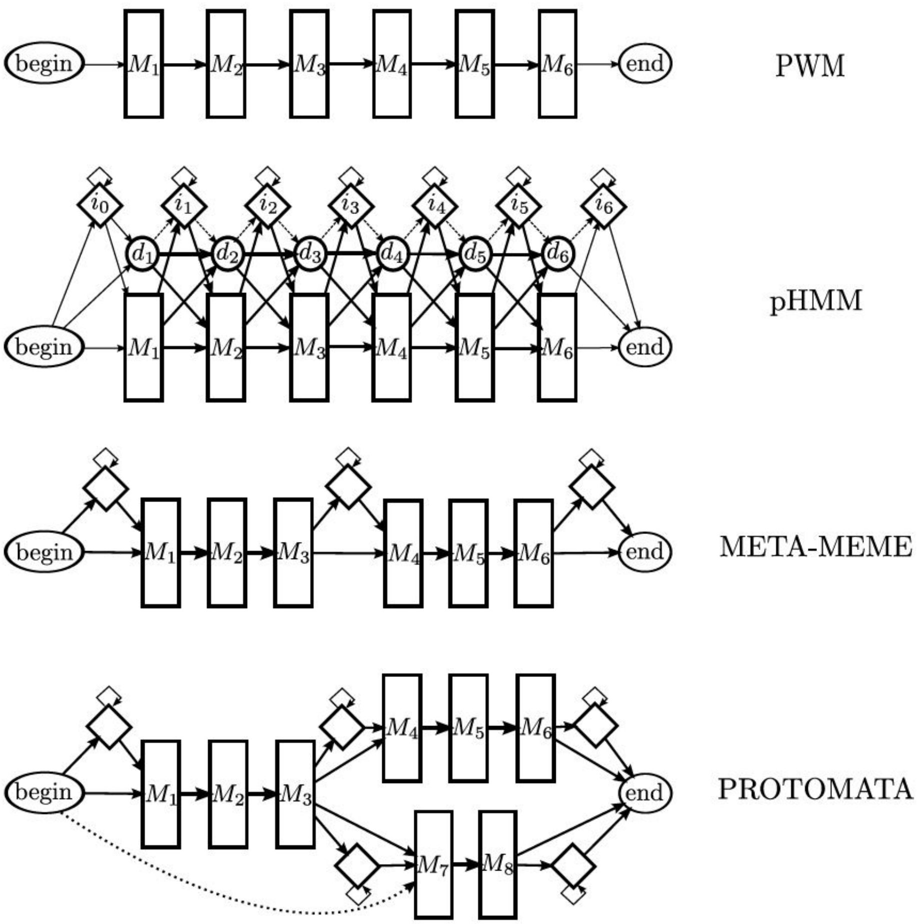


Figure 2: PSSM (PWM), pHMM, Meta-MEME and Protomata types of architecture, reprinted from [Cos16]

weights are to be attributed to rare sequences, and smaller weights to similar sequences. The weights are chosen so that the highest coefficient is equal to one ([THG94]).

This weighting is important to ensure that our model represents the whole family of homologous proteins. To that end, let us denote $S_1, \ldots, S_N$ the sequences, and $\omega_1, \ldots, \omega_n$ the associated weights. Then, the number of occurrences of amino acid a in position i, $E_i(a)$, and the transition frequency from state k to state l, a_{kl}, are adjusted as follows:

$$E_i(a) = \sum_{j=1}^{N} \omega_j \delta_e(i, a, j) \text{ and } a_{kl} = \sum_{i=1}^{N} \omega_j \delta_a(k, l, j) \tag{1}$$

where $\delta_e(i, a, j) = 1$ if the j-th sequence presents an a in the i-th position and 0 otherwise and $\delta_a(k, l, j) = 1$ if the alignment of the alignment of the j-th sequence goes through the transition from state k to state l and 0 otherwise.

Pseudo-counts on emission probabilities The number of parameters to be estimated in a pHMM, for example, can be high compared to the number of sequences usually available. Hence, some of the previous emission probabilities might be estimated as zero, if the training sequence is too small for a given amino acid to be observed, while it might actually appear in other sequences of the family. Typically, if we have less than twenty learning sequences, it is impossible for each amino acid to appear, since there are twenty of them. However, if a new sequence happen to present a previously unseen letter in a position, and a lot of similarity otherwise, we don't want its score to be zero, since we need to recognise new members of the family. To tackle this issue, we can add pseudo-counts to the observed frequencies. The intuition is that we pretend we observed every amino acid even if we didn't, especially in the case of a small training set. If there are a lot of sequences for the training phase, and an amino acid never appears in a position, we are more willing to believe that it is due to a specificity of the family, and not a lack of data, and pseudo-counts should become insignificant compared to the data contributions.

A first possibility is to add a constant to all the counts, for example:

$$e_i(a) = \frac{E_i(a) + 1}{\sum\limits_{a'}(E_i(a') + 1)} \tag{2}$$

A more elaborated constant could take into account the background probability of the amino acid a, denoted q_a. From the numerous sequences, the share of each amino acid can be assessed and employed: an amino acid that usually appears more often is more likely to be observed at any position:

$$e_i(a) = \frac{E_i(a) + Aq_a}{\sum\limits_{a'}E_i(a') + A} \tag{3}$$

The constant A is to be chosen as the weight to put on pseudocounts. It appears that a value around twenty, corresponding to the number of amino acids, seems to provide satisfying results [DEKM98].

If there are only a few training sequences, we assume that the prior knowledge remains valid, and the emission probability is close to the background probability. However, if a lot of proteins are available, it seems reasonable to believe the data, and the pseudo-counts get dominated by the observed counts.

The next step in the elaboration of pseudo-counts consists in considering the observed amino acids at the given position and deduce from them the other possible residues, depending on the fact that some amino acids are more likely to be substituted than others, due to their physical and chemical properties.

Substitution matrices As explained above, substitution matrices are computed from sets of conserved sequences and provide information about the substitutability of amino acid residues. This expertise can be injected in the calculation of the emission probabilities: if the residues in the new sequence are very likely to have been substituted from the observed amino acids in the family,

then the sequence should get a high score ([HH96]).
Let us denote $S = (s_{ab})_{1 \leqslant a,b \leqslant 20}$ the substitution probabilities computed from a substitution matrix. Then for all amino acids a and b, $s_{ab} = s_{ba}$ correspond to the probability of a and b being substituted. The emission probabilities can be adjusted as follows, for each position i and each amino acid a:

$$e_i(a) = \sum_{b=1}^{20} \frac{E_i(b)}{\sum\limits_{b'} E_i(b')} s_{ab} \tag{4}$$

This method embeds more precise knowledge in the estimations, and has been proven to be efficient ([HH96]). However, it presents two drawbacks:

First, the number of training sequences is not fully taken into account. Prior knowledge is integrated, regardless of the fact that it is very needed in the case of a small training set, but should become less important when more data is available.

Secondly, the substitutability of amino acids is computed as a mean over all possible cases. Whether a substitution happens because of a property or because of another makes no difference.

Dirichlet mixtures Pseudo-counts based on Dirichlet mixtures enable the addition of more precise prior knowledge in the estimation of parameters. The idea consists in describing the prototypical distributions of the columns in the alignments ([SKB$^+$96]). Unlike substitution matrices, the important properties of amino acids at each position are considered in order to assess the more substitutable residues. For example, if a column of the alignment presents only small amino acids, Dirichlet mixtures will increase the emission probabilities of other small amino acids, but not the probability of other residues, even if they are exchangeable with the observations in other cases.
The method is described below and consists in creating typical column distributions ; assigning a distribution to each column of the alignment, depending on the observations ; and correcting the emission probabilities by adding the possibility of unobserved amino acids, that are consistent with the probability distribution.

We recall here that a Dirichlet density ρ of parameters $\vec{\alpha} = (\alpha_1, \ldots, \alpha_{20})$ is a density over the set of probability vectors $\vec{p} = (p_1, \ldots, p_{20})$, such that $\forall a, p_a \geqslant 0$ and $\sum\limits_a p_a = 1$, defined as follows:

$$\rho(\vec{p}) = \frac{\prod\limits_{a=1}^{20} p_a^{\alpha_a - 1}}{Z} \tag{5}$$

where Z is a constant such that $\int_{\vec{p}} \rho(\vec{p}) d\vec{p} = 1$.

Given k Dirichlet densities $\rho_1, \ldots, \rho_k$, and k mixture coefficients $q_1, \ldots, q_k$ such that $\sum_j q_j = 1$, a Dirichlet mixture ρ of the components $\rho_1, \ldots, \rho_k$, with coefficients $q_1, \ldots, q_k$ is defined as the weighted combination of the densities:

$$\rho = q_1 \rho_1 + \cdots + q_k \rho_k \tag{6}$$

The set of parameters of the density will be denoted $\Theta = (\vec{\alpha}_1, \ldots, \vec{\alpha}_k, q_1, \ldots, q_k)$.

A lot of sequences, and thus sequences alignments are available. They will be used to infer prior knowledge about proteins and amino acids as follows.
A data set is created with the columns of these alignments. Then, the Dirichlet mixture is chosen as the mixture that maximises the probability of the columns ([SKB$^+$96], [NBGA13]). The number of components has to be chosen. Too many components might lead to over-fitting, but not enough might not provide a good description of the columns. An example of one of the first computed mixtures, that had nine components is given in [SKB$^+$96]. For a fixed number of components, a lot of mixtures are possible, that give comparable results. This is due to the fact that there is no algorithm to compute the exact solution, and approximations, found for instance thanks to expectation maximisation techniques, are used ([SKB$^+$96]).
Once the components are available, and given an observed column in the model of a family of proteins, the coefficients q_i are computed, in order to find a Dirichlet mixture ρ that best meets the amino acid distribution of said column. So, the emission probability $e_i(a)$ of observing amino acid a in position i is assessed through the probability of observing amino acid a given the parameters $\Theta = (\vec{\alpha}_1, \ldots, \vec{\alpha}_k, q_1, \ldots, q_k)$ of the previously defined mixture:

$$e_i(a) = \mathbb{P}\left(a|\Theta, E_i\right) \tag{7}$$

where $E_i = (E_i(a))_a$ denotes the emission frequencies.
The previous equation leads to:

$$e_i(a) = \sum_{j=1}^{k} \mathbb{P}(\vec{\alpha_j}|E_i, \Theta) \frac{E_i(a) + \vec{\alpha}_j(a)}{\sum\limits_{a'}(E_i(a') + \vec{\alpha}_j(a'))} \tag{8}$$

It can be observed that with few sequences, $e_i(a)$ is approximately given by

$$e_i(a) \approx \sum_j q_j \frac{\vec{\alpha}_j(a)}{\sum\limits_{a'} \vec{\alpha_j}(a')} \tag{9}$$

So it is only estimated through prior knowledge. However, when the number of sequences becomes significant, the frequencies $E_i(a)$ get dominant over the $\vec{\alpha}_j(a)$ and the emission probability tends towards the maximum likelihood solution $\frac{E_i(a)}{\sum_{a'} E_i(a')}$.

In the case of a single component in the mixture, the previous formula becomes:

$$e_i(a) = \frac{E_i(a) + \alpha(a)}{\sum\limits_{a'}(E_i(a') + \alpha(a'))} \tag{10}$$

It is exactly the equation (3), that we described earlier, with $\alpha(a)$ replacing Aq_a.

Pseudo-counts based on Dirichlet mixtures enable the integration of precise prior knowledge on protein sequences. This use of background expertise provide the possibility of the efficient modelling of families based on fewer and fewer sequences.
In practice, PSSMs are employed in popular databases, like TRANSFAC and JASPAR ([Cos16]). The current state-of-the-art techniques mostly rely on pHMMs, for which several programs are available, like HMMER or SAM ([Cos16]).

2 Protomata

State-of-the-art models for the representation of families of homologous proteins were introduced. These models, like pHMMs, are left-to-right, and generate the positions independently. Moreover, they rely on the assumption that the sequences share enough similarities to be all aligned, at least locally. However, in the evolution process, there might be some disjunctions. Indeed, some sub-families might have evolved independently one from another, and there might be some regions that are common to some sequences, but not all of them. Moreover, when it comes to functional families of non-necessarily homologous proteins, some sub-families might share very few similarities.
In order to get a more accurate representation of a family of proteins, the Dyliss team recently introduced a new model, called Protomata. This model is based on conserved regions, but allows conserved blocs that don't involve all the proteins, and thus the disjunctions, and describes the possible successions of conserved blocs. Dyliss created a program, named Protomata Learner, that computes a new kind of multiple sequence alignment, said partial and local, from the set of sequences, and deduces the Protomaton.

2.1 Partial Local Multiple Alignment (PLMA)

The Meta-MEME model described above was based on local alignments: the alignments only involved small segments of the sequences to enhance the conservation. In order to keep improving the detection of pertinent conservations, partial local alignments (PLA) can be considered: they are local alignments that do not necessarily involve all the sequences ([CK06],[Ker08]). The set of compatible PLAs constitutes a partial local multiple alignment (PLMA) (figure 3).

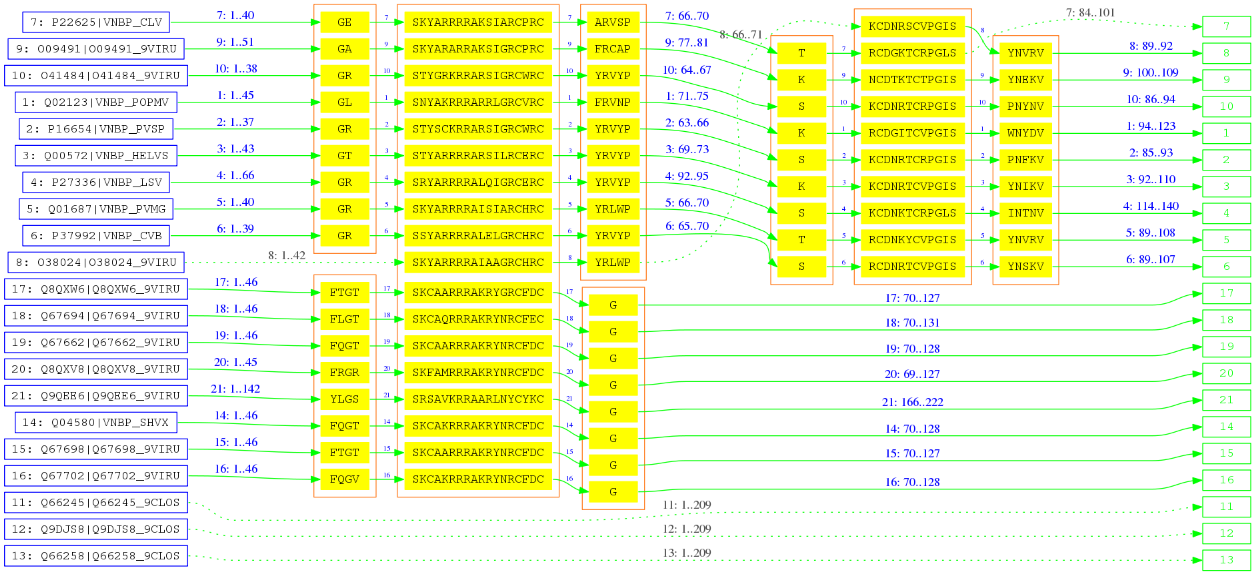


Figure 3: Example of a partial local multiple alignment computed with Protomata Learner
Shared segments of each segments are highlighted, the red square borders define PLAs. The whole forms a PLMA.

An algorithm to compute a PLMA is described in [CK06]. The authors proceed by first detecting similar fragment pairs (SFPs), and ordering them according to their support in the set of sequences. These sorted SFPs are then considered in turn to search for the best PLA including them, with an algorithm based on cliques or connected components, record it and discard the SFPs that are incompatible with it. Incompatibility appears when a position in a sequence corresponds through different SFPs to two different positions in another sequence. At the end of the iterations, a PLMA made of all the recorded PLAs is returned.

Given a partial local multiple alignment of the set of sequences, the Dyliss team proposed to build an automaton modelling the possible successions of PLAs ([CK06]), represented each with a position specific scoring matrix (PSSM), linked by gap states, to enable the insertion of less important segments between the conserved regions.
More specifically, to infer the automaton from the training data, the sequences are firstly used to produce the maximal canonical automaton (MCA). The current states of the MCA correspond to the positions in each sequence, so a PLA can be merged, by merging the states of its local alignments. PLAs of the ordered list are merged, if they are compatible with the previous mergings (see [CK06])
Eventually, states that were not involved in the previous steps can be considered as not representative of the family, and treated as ”gaps”.
An example of a Protomata is given in figure 4. As desired, the model is explicit, and the choice of the parameters should enable generalisation and the discovery of new members of the family.

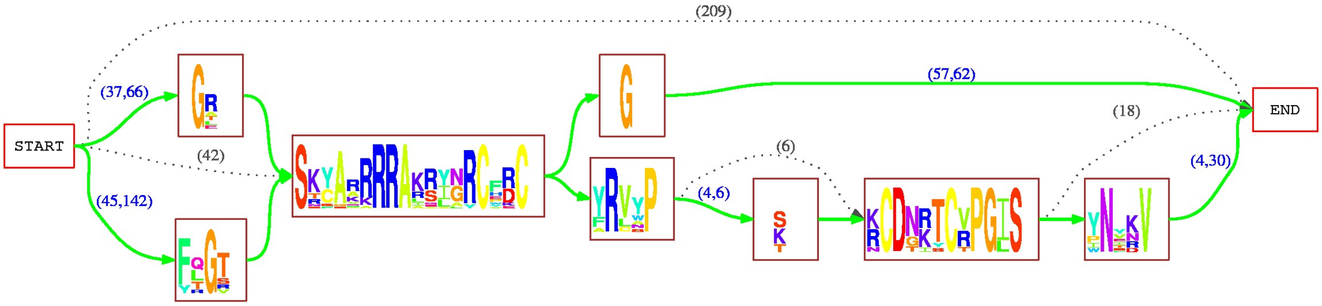


Figure 4: Example of a Protomaton computed with Protomata-Learner

2.2 Estimation of the parameters

From a training set of sequences, we are able to infer the topology of a Protomaton that represents the family. However, we need to elaborate scores for the emission of amino acids within the PSSM states. The currently employed technique is the one that was originally designed for pHMMs, based on the weighting of the training sequences, the maximum likelihood estimation and the elaboration of pseudo-counts through Dirichlet mixtures.
Given an unseen sequence, the score of its best alignment to the Protomaton is computed, by adding the scores of each emission state. This score is then compared to a threshold to decide whether the protein belongs to the family or not.

The presented weighting method provides the best estimation of parameters for pHMMs. However, as far as Protomata are concerned, depending on the path of an alignment, the score computed for a new sequence is highly variable: if the alignment encounters a lot of columns of PSSMs, the score might be a lot lower than if the alignment goes through only a small number of columns and transitions. This variability can make it difficult to compare the scores of two different sequences. Hence, emission probabilities could be adjusted, in order to reduce the variability of the score.
Furthermore, the transitions in Protomata are not weighted. Indeed, even though the training sequences are weighted, the training set might still be biased, and weighting the transitions in a disjunction might favour a subfamily over another. However, transition probabilities might be an answer to the reduction of the variability of the score.

Conclusion

Sequencing provides an exponentially growing database of protein sequences, gathered in families. The task at hand is to design predictive models for these families, that can be used to recognise new members.
The current state-of-the-art model on proteins is the pHMM. It describes conserved positions within the family and allows insertions and deletions. Moreover, the estimation of the emission and transition probabilities is very accurate, as it embeds background knowledge on amino acid distributions, using Dirichlet mixtures. Hence, the construction of the pHMM can be done with very few sequences.
More recently, the Dyliss team introduced a new approach, called Protomata. They propose to design an automaton modelling conserved regions and gaps. The transitions indicate the allowed successions of conserved blocs. This model is more expressive than pHMM, because it is able to handle dissimilarities between sub families. Protomata are currently weighted with the same techniques as pHMM and provide promising results.
However, the scores calculated for previously unseen sequences with Protomata are very variable, due to the variability of the length of the different paths.
Hence, to benefit from the full expressivity offered by Protomata, I will try to adjust the scoring strategy, in order to improve the accuracy of scores computed for new unseen sequences, and reduce the variability induced by the possibility of different paths. Another focus of research will involve the significativity of the scores. Indeed, given the alignment score of a new sequence, we need to predict whether it belongs to the family or not, by comparing the score to a threshold. We expect the models to be able to reject sequences that are very similar, but that do not belong to the family. Therefore, the choice of the threshold is very important and has to be very accurate.

References

[BWML06] T. Bailey, N. Williams, C. Misleh, and W. Li. MEME: discovering and analysing DNA and protein sequence motifs. *Nucleic Acids Research*, 34:369–373, 2006.

[CK06] F. Coste and G. Kerbellec. Learning automata on protein sequences. *JOBIM*, pages 199–210, 2006.

[Cos16] F. Coste. *Topics in Grammatical Inference*, chapter Learning the Language of Biological Sequences, pages 215–247. Springer-Verlag, 2016.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis, Probabilistic models of proteins and nucleic acids*. 1998.

[GBEB97] W. N. Grundy, T.L. Bailey, C.P. Elkan, and M.E. Baker. Meta-MEME: Motif-based hidden markov models of protein families. *Computer Applications in the Biosciences*, 13(4):397–406, 1997.

[HH96] J.G. Henikoff and S. Henikoff. Using substitution probabilities to improve position specific scoring matrices. *Computer applications in the Biosciences*, 12(2):135–143, 1996.

[Ker08] G. Kerbellec. *Apprentissage d'automates modlisant des familles de squences protiques*. PhD thesis, Université de Rennes 1, 2008.

[NBGA13] V. NGuyen, J. Boyd-Graber, and S. Altschul. Dirichlet mixture, the Dirichlet process and the structure of protein space. *Journal of computational biology*, 20(1):1–18, 2013.

[SKB$^+$96] K. Sjolander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian, and D. Haussler. Dirichlet mixtures: A methode for improved detection of weak but significant protein sequence homology. *Bioinformatics*, 12(4):327–345., 1996.

[THG94] J. Thompson, D. Higgins, and T. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.