Distance-based amino acid conservation score

Eoghan Chevé

L3 internship report: Distance-based amino acid conservation score

L3 Internship realised from 03/06/24 to 26/07/24 in the Dyliss team, IRISA/Inria de l'université de Rennes,

under the supervision of Pablo Espana Gutierrez and François Coste.

Eoghan Chevé
L3 SIF
Ecole Normale Supérieure
Rennes, France
eoghan.cheve@ens-rennes.fr

***Abstract*—It is a common task in biology to predict the function of a protein from its proteic sequence. A classical approach is to use predictive models which are estimated from a set of aligned proteic sequences sharing the same function. The models assigns a score to amino acids for each column of the alignment which tells how conserved they are in this column. Such score can be interpreted as expressing how important for the function each amino acids are at a given position.**

In this report we propose an approach that takes distances between pairs of sequences into account for the computation of the conservation score of amino acids at each position.

***Index Terms*—proteic sequences, mutation model, multiple sequence alignment, conservation score, evolutionary distances**

I. Introduction

Proteins are chains of amino acids that perform essential tasks for living beings such as DNA replication or carrying metabolic reactions.

It is a common task in biology to regroup proteins into families, for example, families of proteins which share the same function. But doing so experimentally does not scale to the large number of discovered sequences. Thus, it is a fruitful approach to use computational methods that, for a given family, assign a score to a protein expressing how likely it is to be part of the family.

Numerous methods already exist to help classifying sequences, such as Position Specific Scoring Matrices (PSSM) [1], or Hidden Markov Models (HMM) [6].

Most of these scoring methods only take evolutionary distance into account when removing input bias using sequence weighting methods.

During this internship, we introduced a scoring method describing how conserved amino acids can be and where the distance between sequences is the core of the model.

II. Bibliographic overview of some scoring methods

A. Notations

We will represent amino acids as characters of the alphabet $\mathcal{A} = \{A, R, \dots, V\}$ whose cardinal is $q = 20$. Unspecified amino acids will be denoted with lower case letters such as a or b.

Protein chains will be represented as sequences of characters from $\mathcal{A}$ (*e.g. ALIMFVAWRQM is the proteic sequence representation of a protein chain*).

B. Multiple Sequence Alignment

A Multiple Sequence Alignment (MSA) of n sequences $s^1, \dots, s^n$ is the input of most scoring methods. It is represented by adding gap characters $(-)$ to the sequences in order to obtain aligned sequences $x^1, \dots, x^n \in \mathcal{A}'^*$, with $\mathcal{A}' = \mathcal{A} \cup \{-\}$, which all have the same length L' (see Fig. 1.).

Different algorithms exist to perform an alignment, the most commonly used being ClustalO [11], Muscle [13] and TCoffee [12].

C. Scoring Matrices

Most scoring methods are based on a probabilistic emission model $\mathcal{M}$ assigning to any sequence x the probability $\mathbb{P}(x|\mathcal{M})$ it has to be part of of the family described by the MSA $x^1, \dots, x^n$.

Given a background model $\mathcal{R}$ assigning to any sequence x a background emission probability, we can

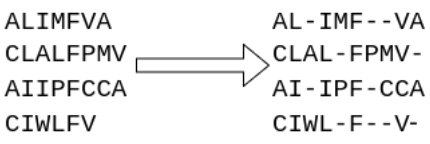

Fig. 1. An example of four sequences (left) being aligned into a MSA (right).

compute a score indicating how likely an aligned sequence x is to be emitted by the model compared to the probability it is to be emitted by $\mathcal{R}$:

$$Score(x) = log\frac{\mathbb{P}(x|\mathcal{M})}{\mathbb{P}(x|\mathcal{R})} \tag{1}$$

In the specific case of Position Based Scoring Matrices, the model will emit for $L < L'$ positions an amino acid likely to be present in $x^1, \ldots, x^n$ at each of those positions [1]. We will note x_i^k the character present at such a position $i \in [\![1, L]\!]$ in x^k.

Moreover, the model emits amino acids at each position independently from other positions. Thus, the score can be computed as follow:

$$Score(x) = log\prod_{i=1}^{L}\frac{p_i(x_i)}{p_0(x_i)} \tag{2}$$

$$Score(x) = \sum_{i=1}^{L} log\frac{p_i(x_i)}{p_0(x_i)} \tag{3}$$

where $p_i(a)$ is the probability that the model emits the amino acid a at position i and $p_0(a)$ the background observation probability of a.

We can now build a matrix of $\mathcal{M}_{q,L}(\mathbb{R})$ where the coefficient in row a and column i is $log\frac{p_i(a)}{p_0(a)}$. This is a Position Specific Scoring Matrix and it allows to calculate the score in equation (3).

D. Estimation of $p_i(a)$

Different estimations of $p_i(a)$ are possible. Here is an overview of the most common one:

1) Using maximum likelihood principle: We can estimate $p_i(a)$ as the observed frequency of the amino acid a at the column $c_i = \langle x_i^1, \ldots, x_i^n \rangle$ of the MSA:

$$p_i(a) := \frac{o_i(a)}{o_i}$$

where $o_i(a) = \sum_{k=1}^{n} \delta(x_i^k = a)$ is the number of occurences of a in c_i and $o_i = \sum_{a \in \mathcal{A}} o_i(a)$ is the number of non gap characters in c_i.

2) Using substitution matrix: an estimation of $p_i(a)$ introduced by Henikoff and Henikoff consists in using the probability $P(a|b)$ that a would be substituted from b, given from a substitution matrix estimated from a great number of MSA. [2]

$$p_i(a) := \sum_{b \in \mathcal{A}} P(a|b)\frac{o_i(a)}{o_i}$$

3) Using pseudo-counts: the method described in 1) has a flaw: when the number of sequences in the MSA is low, some amino acids may not be present at all at a given position, setting their emission probability to be 0.

It is common use to add pseudo-counts of amino acids in the calculation of $p_i(a)$ [3] :

$$p_i(a) = \frac{o_i(a) + p_0(a) \cdot M_i}{o_i + q \cdot M_i}$$

where M_i can be set to various values [4]:

- $M_i = 1$,
- $M_i = \sqrt{o_i}$,
- $M_i = \sqrt{n}$,
- $M_i = \gamma r^i$, where r^i is the number of distinct characters in column i of the MSA and γ is constant.

4) Using Dirichlet mixtures: for all $j \in [\![1, M]\!]$, we can define a vector $\overrightarrow{\beta}_j$ which represent a typical MSA column [5] (e.g. Fig. 2.). $\overrightarrow{\beta}_j$ is statistically estimated over a large number of MSA

A	0.2706
C	0.0398
D	0.0175
E	0.0164
F	0.0142
G	0.1319
H	0.0123
I	0.0225
K	0.0203
L	0.0307
M	0.0153
N	0.0482
P	0.0538
Q	0.0206
R	0.0236
S	0.2161
T	0.0654
V	0.0654
W	0.0037
Y	0.0096

Fig. 2. An example of a vector $\overrightarrow{\beta}_j$ that represents a typical MSA column.

We define a new estimation of the emission probability:

$$p_i(a) := \sum_{j=1}^{M} \frac{o_i(a) + \overrightarrow{\beta}_j(a)}{o_i + |\overrightarrow{\beta}_j|}\mathbb{P}(\overrightarrow{\beta}_j|i, \theta)$$

where $\overrightarrow{\beta}_j(a)$ is the coefficient of $\overrightarrow{\beta}_j$ corresponding to a and $\mathbb{P}(\overrightarrow{\beta}_j|o_i,\theta)$ is the probability that the column i of the MSA correspond to a typical column represented by $\overrightarrow{\beta}_j$ given a parameter θ.

E. Sequence weighting

In practice, the MSA used to infer the model doesn't represent a uniform sample of sequences.

A common issue is sample bias. For example: if we want to infer the model on sequences representing insulin proteins of different species, it is possible that the number of sequences coming from primates (which are very close sequences) is greater than the number of sequences coming from other mammals because the former where more studied. Yet, primates don't represent the majority of mammal species.

Thus, it is common to assign weights to sequences in order to mitigate over-representation of close sequences in the MSA.

To that extent, $o_i(a)$ can be redefined as follow :

$$o_i(a) = \sum_{k=1}^{n} \delta(x_i^k = a)\omega_k$$

where ω_k is a weight associated to the aligned sequence x^k.

This is what is called internal sequences weighting. Another approach consists of weighting all the sequences at once in order to change the value o_i to make it more or less important in the estimation of $p_i(a)$ when using pseudo-counts or Dirichlet mixtures. However, we will only focus on internal weighting strategies:

1) Clustering approach: the idea is to realise a clustering based on an identity threshold in the sequence space. The weight associated to a sequence will be the inverse of the cardinal of its cluster [2].

2) Tree-based approach: this approach uses a phylogenetic tree inferred from the MSA and assign a weight to each sequence according to their position in the tree [6].

3) Position based approach: the idea is to set the weight assigned to each sequence based on how different they are from others at each position [7]:

$$\omega_k = \sum_{i=1}^{L} \sum_{a \in \mathcal{A}} \delta(x_i^k = a) \frac{1}{r^i o_i(a)}$$

This is the approach used by the HMMER package [14].

4) Distance based approaches: several approaches take into account the distance $d(x^k, x^l)$ which is the number of mismatches between x^k and x^l.

An early example was proposed by Vigron and Argos in 1989 [8]:

$$\omega_k = \sum_{l=1}^{n} d(s^k, s^l)$$

Later, Argos will use Voronoï cells to compute the weights: for each sequence x^k, ω_k will be the area of the associated Voronoï cell in a rectangle of the sequence space [9].

5) Novelty approach: The weight associated to a sequence is relative to its novelty which represents how many substitutions occured from a common ancestor to another sequence [10].

III. Contributions

It is worth mentioning that the approaches mentioned in II-E4 and II-E5 are distance based approaches, but they are used to assign a weight to the whole sequences and are not enabling to score amino acids for each positions of the sequences with respect to sequences distances.

During this internship, we designed a model describing the probability to observe an amino acid at position i after it just mutated.

Then, we were able to create a scoring matrix from this model so it would be possible to compare it to a PSSM.

A. Mutation model

Let's assume, given n aligned sequences $x^1, \ldots, x^n$, that each pair of sequences $\{x^k, x^l\}$ possess a *most recent common ancestor* $y^{k,l}$ which is an unknown sequence that is the latest to have presumably mutated to both x^k and x^l in the evolutionary history.

A first hypothesis in the model is to consider that given a position i, if x_i^k and x_i^l share the same amino acid a, then it was inherited from $y^{k,l}$ at position i. Else, $y_i^{k,l}$ will be an unknown character of $\mathcal{A}'$ noted X (see Fig. 3.).

Considering the distance $d(x^k, x^l)$ between x^k and x^l as the number of mismatches between them, a second hypothesis made in the model is to consider that the mutations are distributed evenly and that the number of mutations that happened between $y^{k,l}$ and x^k and between $y^{k,l}$ and x^k are both $d_{k,l} = \lfloor \frac{d(x^k,x^l)}{2} \rfloor$.

Mutations are modelled as successive uniform selections of a position in the sequence and substitutions of the corresponding character.

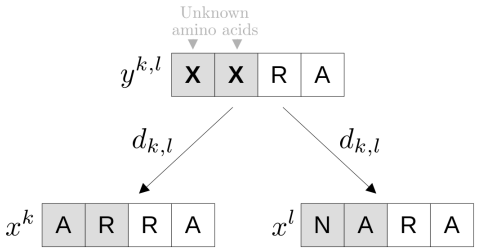

Fig. 3. Representation of the mutation model: $d_{k,l}$ is the number of occurred mutations from $y^{k,l}$ to its descendants.
The characters conserved in x^k and x^l which are the R in position 3 and the A in position 4 were already present in $y^{k,l}$, while others are descendants of an unknown character X.

When $y^{k,l}$ mutates to x^k, a position i is uniformly chosen at each mutation. The character in that position will then be substituted according to an observation distribution in position after a single mutation, noted α_i.

The vectors $\alpha_i = (\alpha_{i,a})_{a\in\mathcal{A}'}$ are the parameters of the model and need to be estimated. They will define the distance-based conservation score.

B. Set of independent pairs

The parameters α_i will be inferred from a set of observed pairs of sequences with respect to their most common ancestor.

Given an input MSA, a first task is to define a set of such sequence pairs.

We consider a set G of supposedly independent pairs of sequences, which means that for two different pairs $\{x^k, x^l\}$ and $\{x^t, x^u\}$ of G, $y^{k,l}$ is not a descendant of $y^{t,u}$ and is not an ancestor of either x^t or x^u.
See Fig. 4. for an example of two independent pairs represented in a phylogenetic tree. The pairs are represented as the same colour nodes in the trees.

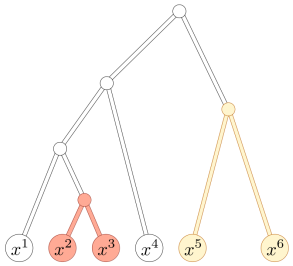

Fig. 4. Representation of two sequence pairs $\{x^2, x^3\}$ and $\{x^5, x^6\}$ of G in a phylogenetic tree. Note that there are no interference between the two pairs which means they are independent under our definition. Moreover, the pair $\{x^1, x^4\}$ is independent to the two others, meaning it can be part of G too

The choice of G is crucial for the model and we will discuss different definitions of G in section III-E.

Moreover, the first and second hypothesis of the model are viable for pair of sequences that are close to each other, i.e. if $d(x^k, x^l) << L'$. Thus, choosing pairs of G that are both independent and close seems to be the way to go.

C. Estimation of α_i

1) Likelihood function: we introduce the following likelihood function and we want to estimate the distribution α_i that maximise it:

$$\mathcal{L}_{(\theta=(\alpha_i)_{i\in[\![1,L]\!]})} = \mathbb{P}\left(\bigcap_{\{l,k\}\in G} x^k \swarrow^{y^{k,l}} \searrow x^l \mid \theta\right) \quad (4)$$

Where $x^k \swarrow^{y^{k,l}} \searrow x^l$ is the event: "$y^{k,l}$ mutated to x^k and x^l".
The L vectors $(\alpha_i)_{i\in[\![1,L]\!]}$ are the parameters of the function and model the probability of substitution if a mutation happens at position i between $y^{k,l}$ and x^k.

We can assume that the independence of the pairs of G allows the events $x^k \swarrow^{y^{k,l}} \searrow x^l$ to be mutually independents. Moreover, we can make a third and last hypothesis on the mutation model which states that mutations in position i happen independently to the mutations in other positions j. Thus, equation (4) can be expressed as:

$$\mathcal{L}_\theta = \prod_{\{l,k\}\in G}\prod_{i=1}^{L'} \mathbb{P}\left(x_i^k \swarrow^{y_i^{k,l}} \searrow x_i^l|\theta\right)$$

$$\mathcal{L}_\theta = \prod_{\{l,k\}\in G}\prod_{i=1}^{L'} p_i^{k,l}(\theta)$$

Where $x_i^k \swarrow^{y_i^{k,l}} \searrow x_i^l$ is the event "$y^{k,l}$ mutated to the characters x_i^k and x_i^l at position i" and $p_i^{k,l}(\theta) = \mathbb{P}\left(x_i^k \swarrow^{y_i^{k,l}} \searrow x_i^l|\theta\right)$.

Instead of maximising the likelihood function, we maximise its logarithm as it won't change the estimated value of α_i and it will simplify its computation. We define the L distributions $(\alpha_i)_{i\in[\![1,L]\!]}$ as the parameters that maximise the log-likelihood function $ln(\mathcal{L}_\theta)$ under the constraints $\sum_{a\in\mathcal{A}'} \alpha_{i,a} = 1$ and $\alpha_{i,a} \geq 0\ \forall a \in \mathcal{A}'$.

We end up to the following maximisation problem:

$$\begin{aligned} \max_{\theta=(\alpha_i)_{i\in[\![1,L]\!]}} \quad & ln(\mathcal{L}_\theta) = \sum_{\{l,k\}\in G}\sum_{i=1}^{L'} ln(p_i^{k,l}(\theta)) \\ \text{s.t} \quad & \alpha_{i,a} \geq 0\ \forall a \in \mathcal{A}',\ \forall i \in [\![1,L]\!] \\ & \sum_{a\in\mathcal{A}'} \alpha_{i,a} = 1\ \forall i \in [\![1,L]\!] \end{aligned} \quad (5)$$

2) System of equations: there are different ways to define $p_i^{k,l}(\theta)$ (cf. Appendix A and B) and we will focus on the one using random draws with returns (cf. Appendix B):

$$p_i^{k,l}(\theta) = \begin{cases} \left(\frac{L'-1+\alpha_{i,a}}{L'}\right)^{2d_{k,l}} & \text{if } x_i^k = a \\ & \text{and } x_i^l = a \\ \left(1 - (\frac{L'-1}{L'})^{d_{k,l}}\right)^2 \alpha_{i,a}\alpha_{i,b} & \text{if } x_i^k = a \\ & \text{and } x_i^l = b \neq a \end{cases} \tag{6}$$

With this method, it is possible to reduce the maximisation problem (equation (5)) to the following system using the method of Lagrange multipliers for each position i (see appendix C):

$$\begin{cases} \frac{|M_{i,a}|}{\alpha_{i,a}} + \sum\limits_{\{k,l\}\in C_{i,a}} \frac{2d_{k,l}}{L'-1+\alpha_{i,a}} + \lambda_i + \lambda_{i,a} = 0 & \forall a \in \mathcal{A}' \\ \lambda_{i,a}\alpha_{i,a} = 0 & \forall a \in \mathcal{A}' \\ \alpha_{i,a} \geq 0 & \forall a \in \mathcal{A}' \\ \lambda_{i,a} \geq 0 & \forall a \in \mathcal{A}' \\ \sum\limits_{a\in\mathcal{A}'} \alpha_{i,a} = 1 \end{cases} \tag{7}$$

where $M_{i,a} = \{\{k,l\} \in G \mid x_i^k = a \oplus x_i^l = a\}$ is the set of pairs containing an amino acid a coming from a *mutation* at position i and $C_{i,a} = \{\{k,l\} \in G \mid x_i^k = a \wedge x_i^l = a\}$ is the set of pairs containing a *conserved amino* acid a at position i.

D. Building a scoring matrix

Given a background model $\mathcal{R}$, we can now define a score expressing how likely a sequence x is to be the descendant of an unknown sequence following our mutation model $\mathcal{M}'$.

In the same way as (1), we can define:

$$Score(x) = log\frac{\mathbb{P}(x|\mathcal{M})}{\mathbb{P}(x|\mathcal{R})}$$

$$Score(x) = \sum_{i=1}^{L} log\frac{\alpha_{i,x_i}}{p_0(x_i)}$$

which allows to build a scoring matrix which coefficients in line a and column i are the $log\frac{\alpha_{i,a}}{p_0(a)}$.

E. Choice of G

The choice of G should be an easier task if the phylogenetic tree of the sequences was known. In practice, we don't have access to the exact tree, so we have to find a compromise between representativeness (higher number of couples of sequences) and exactitude (couples more likely to be independent with each others).

Thus different approaches are possible:

1) Maximum representativeness: by choosing G as $\{\{x^k, x^l\} | (k,l) \in [\![1,n]\!]^2\}$, each possible pair of sequences is represented in G, so there is no loss of information. But it is certain that pairs of sequences aren't independent with each others, making the model inexact.

2) Closest neighbours: A good compromise is to realise a stable marriage between the sequences. This method set the cardinal of G to n or $n-1$ which is the maximum cardinal where it can not be certain that two pairs are dependants. Moreover, the fact that sequences are close to each other make the first and second hypothesis of the mutation model more viable.

F. Improvements

The preferred choice of G is the one portrayed in III-E2. But sequences in pairs being really close imply that at a given position i, an amino acid a conserved in a pair will contribute little to nothing to the computation of $\alpha_{i,a}$.

For example, with $G = \{\{x^1, x^2\}, \{x^3, x^4\}\}$, $x^3 = R$, $x^4 = N$ and $x^1 = x^2 = A$, the resolution of (7) gives $\alpha_{1,A} = 0$ and $\alpha_{1,R} = \alpha_{1,N} = 0.5$ even though A is observed at position 1.

An explanation to the result of this example is that after a mutation occurs at position i, we only observe R and N, as A is observed in sequences that never mutated from their most recent common ancestors.

To fix this issue, it is possible to add hypothetical ancestor sequences containing the unknown character X to the MSA the model is based on, and compute an estimation of the probability that those ancestor sequences are descendant from a more ancient ancestor sequence.

1) Building an expanded set G: we can join $G^{(1)}$ to the previously defined set G where $G^{(1)} \in (\mathcal{H}^{(1)})^2$ is chosen with the closest neighbour method similarly to G and $\mathcal{H}^{(1)} = \{y^{k,l} | \{k,l\} \in G\}$ is the set of the most common ancestors of the pairs of G.

Recursively, we can compute for each $t \in \mathbb{N}$ a set $G^{(t)} \in (\mathcal{H}^{(t)})^2$ where $\mathcal{H}^{(t)}$ is the set of most common ancestors of the pairs of $G^{(t-1)}$.

With this method, we estimate the evolution history of the input sequences, which means the score is estimated from a set containing more information.

Because of this, we recommend to use this approach when it comes to infer G.

IV. Implementation

We implemented an algorithm that takes an input MSA and returns each distribution α_i that models the probability of conservation of amino acids at position i after a single mutation by resolving the system described in equation (7).

For the resolution, we didn't find an analytic formula for the estimator $\alpha_{i,a}$. Instead, we expressed each $\alpha_{i,a}$ as a function of λ_i noted $\alpha_{i,a}(\lambda_i)$ (see appendix D) and estimated the roots of $-1 + \sum_{a \in \mathcal{A}'} \alpha_{i,a}(\lambda_i)$ with a solver from the 'scipy.optimize' Python library.

We first tested the algorithm on a MSA of the set of sequences PS50001 from Profile. This set contain 455 sequences which contain the SH2 domain which is required for intracellular signalling cascades.

Unfortunately, we lacked time to compare the resulting scoring matrix performances to the performances of other methods.

V. Conclusion and perspectives

During this internship we introduced a new approach to estimate the conservation of amino-acids based on the distance between pairs of sequences. It is, to our knowledge, the first method that integrate distances directly in its model.

This mutation model represents how amino-acids tend to mutate in the input family under evolutionary pressure thanks to the parameters α_i. Those parameters are the probability distributions which determine the conservation score for each amino acids at a given position and are obtained through the maximisation of a log-likelihood function.

This method needs now to be evaluated by experiments and real data and to be compared to existing HMM and PSSM methods.

Currently, we approximate the estimations of α_i using an equation solver and it would be an interesting task to find an analytical formula for each α_i.

Another future work would be to add pseudo-counts to the estimation of the distributions α_i. An idea we have is to add couples $\{xa, xx\}$ to G where xa is a sequence with only the character a, and xx is a totally unknown sequence which only have Xs.

Acknowledgment

I would like to thank François Coste for giving me the opportunity of an internship and for giving me vital advice on the rigour needed for such an internship. Thank you to Pablo Espana Gutierrez for supporting me in my work and particularly for helping me formalising the model from the idea it was at the start. Thank you to my fellow interns for sharing their point of view and for helping me discover the bio-informatics domain. Finally, I want to thank everyone on the Dyliss team for the interesting discussions we could share.

References

[1] Stormo et al., *Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli*, 1982
[2] Henikoff and Henikoff, *Amino acid substitution matrices from protein blocks.*, 1992
[3] Dodd Egan, *systematic method for the detection of potential lambda Cro-like DNA-binding regions in proteins.*, 1987
[4] Henikoff and Henikoff, *Using substitution probabilities to improve position-specific scoring matrices*, 1996
[5] Sjölander, *Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology*, 1996
[6] Durbin, *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*, 1998
[7] Henikoff and Henikoff, *Position-based Sequence Weights*, 1994
[8] Vingron, Argos, *A fast and sensitive multiple sequence alignment algorithm*, 1989
[9] Sibbald, Argos, *Weighting Aligned Protein or Nucleic Acid Sequences to Correct for Unequal Representation*, 1990
[10] De Maio et al., *A phylogenetic approach for weighting genetic sequences*, 2021
[11] Fabian Sievers and Desmond G. Higgins, *Clustal Omega for making accurate alignments of many protein sequences*, 2018
[12] Cédric Notredame, Desmond G. Higgins and Jaap Heringa,*T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment*, 2000
[13] Robert C. Edgar, *MUSCLE: multiple sequence alignment with high accuracy and high throughput*, 2004
[14] Robert D. Finn, Jody Clements, and Sean R. Eddy, *HMMER web server: interactive sequence similarity searching*, 2011

APPENDIX A : DETAILED ESTIMATION OF $p_i^{k,l}(\theta)$ USING RANDOM DRAWS WITHOUT RETURNS

In this appendix, we will detail the computation of $\mathbb{P}\left(x_i^k \swarrow^{y_i^{k,l}} \searrow x_i^l|\theta\right)$.

We define:

$x_i^k \swarrow^{y_i^{k,l}} \searrow x_i^l = (y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k) \bigcap (y_i^{k,l} \xrightarrow{d_{k,l}} x_i^l)$

where the events $y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k$ can be interpreted as: "After $d_{k,l}$ mutations from $y^{k,l}$ to x^k, position i mutated to x_i^k", and are mutually independent.

In this approach, we consider the mutation model to be a random draw without return which means each position can mutate at most once:

- If $x_i^k = x_i^l = a$ we can assume that $y^{k,l} = a$. Thus, $y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k$ can be seen as the event : "After $d_{k,l}$ mutations from $y^{k,l}$ to x^k, position i did not change". Thus:

  $\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \mathbb{P}(\bigcap_{j=0}^{d_{k,l}-1} \overline{A}_i^j|\alpha_i)$

  Where A_i^j is the event: "character at position i changed at the j^{th} mutation".

  Moreover, the events A_i^j are mutually independent, so

  $\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \prod_{j=0}^{d_{k,l}-1} \frac{L'-j-1+\alpha_{i,a}}{L'-j}$

- Else, if $x_i^k = a$ and $x_i^l = b$, then, we have no information on $y_i^{k,l}$. We can assume the character at position i is a character that would change if its position were to be drawn (the character X would get from unknown to a character of $\mathcal{A}'$). So, $\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i)$ would be the probability that position i is chosen to mutate by one of the $d_{k,l}$ draw and it mutate to the character a. Thus :

  $\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \frac{d_{k,l}}{L'}\alpha_{i,a}$.

Thus, with this approach,

$$p_i^{k,l}(\theta) = \mathbb{P}\left((y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k) \bigcap (y_i^{k,l} \xrightarrow{d_{k,l}} x_i^l)|\alpha_i\right)$$

$$p_i^{k,l}(\theta) = \begin{cases} \prod_{j=0}^{d_{k,l}} \left(\frac{L'-j-1+\alpha_{i,a}}{L'-j}\right)^2 & \text{if } x_i^k = x_i^l = a \\ \left(\frac{d_{k,l}}{L'}\right)^2 \alpha_{i,a}\alpha_{i,b} & \text{if } x_i^k = a \text{ and } x_i^l = b \end{cases}$$

APPENDIX B : DETAILED ESTIMATION OF $p_i^{k,l}(\theta)$ USING RANDOM DRAWS WITH RETURNS

This approach is exactly the same as the one presented in appendix A, but the mutation model is a random draw with return, which means positions can mutate several times.

Thus, if $x_i^k = a$ and $x_i^l = a$, we now have

$\mathbb{P}(\bigcap_{j=0}^{d_{k,l}-1} \overline{A}_i^j|\alpha_i) = \prod_{j=0}^{d_{k,l}-1} \frac{L'-1+\alpha_{i,a}}{L'}$

$\mathbb{P}(\bigcap_{j=0}^{d_{k,l}-1} \overline{A}_i^j|\alpha_i) = \left(\frac{L'-1+\alpha_{i,a}}{L'}\right)^{d_{k,l}}$

So

$\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \left(\frac{L'-1+\alpha_{i,a}}{L'}\right)^{d_{k,l}}$

If $x_i^k = a$ and $x_i^l = b$, the probability that i is chosen to mutate by one of the $d_{k,l}$ draw would be $\mathbb{P}(B_i) = 1 - \left(\frac{L'-1}{L'}\right)^{d_{k,l}}$, leading to :

$\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \mathbb{P}(B_i)\alpha_{i,a}$

$\mathbb{P}(y_i^{k,l} \xrightarrow{d_{k,l}} x_i^k|\alpha_i) = \left(1 - \left(\frac{L'-1}{L'}\right)^{d_{k,l}}\right)\alpha_{i,a}$

Following this, we have :

$$p_i^{k,l}(\theta) = \begin{cases} \left(\frac{L'-1+\alpha_{i,a}}{L'}\right)^{2d_{k,l}} & \text{if } x_i^k = a \text{ and } x_i^l = a \\ \left(1 - (\frac{L'-1}{L'})^{d_{k,l}}\right)^2 \alpha_{i,a}\alpha_{i,b} & \text{if } x_i^k = a \text{ and } x_i^l = b \neq a \end{cases}$$

APPENDIX C : REDUCTION OF THE MAXIMISATION PROBLEM TO A SYSTEM OF EQUATIONS

Given i a position and a an amino acid, we define $g_i : (x_{j,a})_{a\in[\![1,q+1]\!],j\in[\![1,L]\!]} \mapsto \sum_{a=0}^{q} x_{i,a} - 1$ and $g_{i,a} : (x_{j,a})_{a\in[\![1,q+1]\!],j\in[\![1,L]\!]} \mapsto x_{i,a}$

The constraints given by equation (5) can be expressed as constraints functions $g_i(\theta) = 0$ and $g_{i,a}(\theta) \geq 0$. Thus, according to the Lagrange multiplier method and Kuhn–Tucker theorem, if θ is solution to the problem (5) then there exist $(\lambda_i)_{i\in[\![1,L]\!]} \in \mathbb{R}^L$ and $(\lambda_{i,a})_{i\in[\![1,L]\!],a\in\mathcal{A}'} \in \mathbb{R}^{L\times(q+1)}$ such as θ is also a solution to the following system:

$$\begin{cases} \frac{\partial ln(\mathcal{L}_\theta)}{\partial\alpha_{i,a}} + \sum_{j=1}^{L} \lambda_j \frac{\partial g_j(\theta)}{\partial\alpha_{i,a}} + \lambda_{i,a}\frac{\partial g_{i,a}(\theta)}{\partial\alpha_{i,a}} = 0 & \forall i \in [\![1,L]\!], a \in [\![0,q]\!] \\ g_{i,a}(\theta)\lambda_{i,a} = 0 & \forall i \in [\![1,L]\!], a \in [\![0,q]\!] \\ g_{i,a}(\theta) \geq 0 & \forall i \in [\![1,L]\!], a \in [\![0,q]\!] \\ \lambda_{i,a} \geq 0 & \forall i \in [\![1,L]\!], a \in [\![0,q]\!] \\ g_i(\theta) = 0 & \forall i \in [\![1,L]\!] \end{cases}$$

Where, for a fixed position i,

$$\begin{aligned} \frac{\partial ln(\mathcal{L}_\theta)}{\partial\alpha_{i,a}} &= \sum_{\{l,k\}\in G} \sum_{j=1}^{L'} \frac{\partial ln(p_j^{k,l})}{\partial\alpha_{i,a}}(\theta) \\ &= \sum_{\{l,k\}\in G} \frac{\partial ln(p_i^{k,l})}{\partial\alpha_{i,a}}(\theta) \end{aligned}$$

Where

$$\frac{\partial ln(p_i^{k,l})}{\partial\alpha_{i,a}} = \begin{cases} \frac{2d_{k,l}}{L'-1+\alpha_{i,a}} & \text{if } x_i^k = x_i^l = a \\ \frac{1}{\alpha_{i,a}} & \text{if } x_i^k = a \text{ and } x_i^l = b \neq a \\ 0 & \text{else} \end{cases}$$

And

$$\sum_{j=1}^{L} \lambda_j \frac{\partial g_j(\theta)}{\partial\alpha_{i,a}} = \lambda_j \frac{\partial g_j(\theta)}{\partial\alpha_{i,a}} = \lambda_i$$

$$\lambda_{i,a}\frac{\partial g_{i,a}(\theta)}{\partial\alpha_{i,a}} = \lambda_{i,a}$$

From this, we obtain, for each position i, the system described in equation (7).

APPENDIX D : EXPRESSING $\alpha_{i,a}$ AS A FUNCTION OF λ_i

Given a character a and a position i, assuming at least one known character is present in the column i of the alignment, two cases are possible :

- a is not present in the column i of the alignment:
  Then
  $\lambda_{i,a} = -\lambda_i$
  $\alpha_{i,a} = 0$
- a is present in the column i of the alignment:
  Then
  $\lambda_{i,a} = 0$
  $\alpha_{i,a} = \alpha_{i,a}(\lambda_i)$

Where the function $x \mapsto \alpha_{i,a}(x)$ is calculated this way:
Developing the first line of the system described in (7), we obtain the following equation :

$$\frac{\lambda_i \alpha_{i,a}^2 + \left(|G_{i,a}| + \sum_{\{k,l\} \in G'_{i,a}} 2d_{k,l} + (L'-1)\lambda_i \right) \alpha_{i,a} + |G_{i,a}|(L'-1)}{\alpha_{i,a}(L' - 1 + \alpha_{i,a})} = 0$$

Noting
$b_{i,a}(\lambda_i) = \left(|G_{i,a}| + \sum_{\{k,l\} \in G'_{i,a}} 2d_{k,l} + (L'-1)\lambda_i \right)$
$c_{i,a} = |G_{i,a}|(L'-1)$
$\Delta_{i,a}(\lambda_i) = b_{i,a}(\lambda_i)^2 - 4\lambda_i c_{i,a}$
We can express $\alpha_{i,a}(\lambda_i)$ as

$$\alpha_{i,a}(\lambda_i) = \frac{-b_{i,a}(\lambda_i) - \sqrt{\Delta_{i,a}(\lambda_i)}}{2\lambda_i}$$

(We consider that λ_i is negative as it can't be a solution of the system if positive, which means $\sqrt{\Delta_{i,a}(\lambda_i)} > b_{i,a}(\lambda_i)$ which leads to only one possible root, $\alpha_{i,a}(\lambda_i)$, of the equation given there is at least one root).