Statistical analysis of protein families

Martin Weigt

The objective of this little project is to analyze a family of protein domains given by a multiple-sequence alignment (MSA)

- aim 1: finding conserved positions,
- aim 2: decision, if other sequences belong to the same family,
- aim 3: detection of amino-acid co-variance between different MSA columns, and relation to the 3D protein structure.

The aim is to get information about a protein just by analyzing the statistics of evolutionarily related sequences.

I. DATA

We have 4 files:

- `train.faa` is a MSA of 1000 protein domains coming from the same family. It is given in FASTA format:
  ```
  >IVBI5_BUNMU/30-82
  -CNLPPDPGPCHDNKFAFYHHPASNKCKEFVYGGCGGNDNRFKTRNKCQCTC-
  >C1IC53_WALAE/30-82
  -CHLPADPGPCSNYRPAYYYNPASRKCEEFMYGGCKGNKNNFKTRHECHRVCV
  >IVBI2_PSETT/30-82
  LCELPPDTGPCRVRFPSFYYNPDEQKCLEFIYGGCEGNANNFITKEECESTC-
  >IVBI1_OXYMI/30-82
  LCELPADTGPCRVGFPSFYYNPDEKKCLEFIYGGCEGNANNFITKEECESTC-
  ```
  Rows starting with ">" are comment lines, they contain e.g. the protein and species name, and the localization of the domain inside the protein. They have no importance for our analysis. The other lines contain the actual sequence alignment, i.e. they are important for us. The sequences are aligned, as you can guess from the existence of gaps and the constant sequence length ($L = 53$ in our case). Each position contains either an amino acid (A,C,...,Y, there are 20 different amino acids) or a gap ($-$), which we consider as the 21st letter. Together they form the alphabet

  $$\mathcal{A} = \{\mathtt{A}, \mathtt{C}, ..., \mathtt{Y}, -\}$$

  with $q = 21$ letters. Note that the alignments may contain non-standard / not completely determined amino acids (e.g. X), which, for the sake of simplicity, we consider as gaps.
- `test1.faa`: Same format as `train.faa`. It is the first file we will test with the inferred statistical model, which will be learned starting from `train.faa`.
- `test2.faa`: Same format as `train.faa`. It is the second file we will test with the inferred statistical model, which will be learned starting from `train.faa`. One of the two test files is in the same, one in another family.
- `distances.txt`: Contains the distances between amino acids in an exemplary 3D structure of a protein from the same family. The file has three columns: The ID of the first position, the ID of the second position, and the actual distance in Angstrom.

II. MODELING BY PSWM

A. Estimating a PSWM

We aim at modeling a protein family, which is given by the MSA

$$D_{train} = \begin{pmatrix} a_1^1 & a_2^1 & ... & a_L^1 \\ a_1^2 & a_2^2 & ... & a_L^2 \\ & & ... & \\ a_1^M & a_2^M & ... & a_L^M \end{pmatrix} .$$

The entries $a_i^m \in \mathcal{A}$ are amino acids (or a gap) in column $i \in \{1, ..., L\}$ and in the sequence number $m \in \{1, ..., M\}$, with M being the total number of sequences (rows) in the MSA ($M = 1000$ in our case). The quantities needed for our modeling of the family are

$$n_i(a) = \text{number of occurrences of amino acid } a \text{ in position } i \ . \tag{1}$$

These numbers are specific for each position (column of D_{train}) and each amino acid. They satisfy $\sum_{a\in\mathcal{A}} n_i(a) = M$ for each $i \in \{1, ..., L\}$.

The statistical model is a factorized model,

$$P(a_1, ..., a_L|\omega) = \prod_{i=1}^{L} \omega_i(a_i) \ , \tag{2}$$

i.e. we assume distinct positions to be statistically independent. The parameters $\omega_i(a)$, which form a $L\times q$-dimensional matrix, are called the "position specific weight matrix" (PSWM). The weights can be calculated using the occurrence numbers,

$$\omega_i(a) = \frac{n_i(a) + \mu}{M + q\mu} \ . \tag{3}$$

The parameter μ is a so-called pseudo-count, we will use $\mu = 1$ here.

B. Conservation

First we are looking for *conserved positions*, i.e. positions where we have one single high weight for one amino acid, and small weights for the others. Such positions are frequently conserved since they have some essential biological function, substituting them may interrupt the biological function of the protein.

To find these positions, we search the highest weight for each position $i = 1, ..., L$

$$\omega_i(a_i^\star) = \max_{a\in\mathcal{A}} \omega_i(a) \ . \tag{4}$$

We determine all positions i with $\omega_i(a_i^\star) > 0.5$.

C. Evaluating a new sequence

How can we decide if a new sequence $b = (b_1, ..., b_L)$ (which is not contained in the original MSA D_{train}) belongs to the same family, or to another one? According to Eq. (2) its probability is

$$P(b_1, ..., b_L|\omega) = \prod_{i=1}^{L} \omega_i(b_i) \ . \tag{5}$$

To decide, if this probability is "sufficiently large" for saying that b is in the family given by D_{train}, we have to compare it with a *null model*, which is not position specific:

$$P^{(0)}(b_1, ..., b_L) = \prod_{i=1}^{L} f^{(0)}(b_i) \tag{6}$$

with (for all $a \in \mathcal{A}$)

$$f^{(0)}(a) = \frac{1}{L} \sum_{i=1}^{L} \omega_i(a). \tag{7}$$

The non-specific weight is simply given as the average of the position-specific weight $\omega_i(a)$ over all columns $i = 1, ..., L$.

To compare the PSWM model with the null model, we calculate the *log-odds ratio*

$$S(b_1, ..., b_L) = \ln \frac{P(b_1, ..., b_L|\omega)}{P^{(0)}(b_1, ..., b_L)} = \sum_{i=1}^{L} \ln \frac{\omega_i(b_i)}{f^{(0)}(b_i)} \ . \tag{8}$$

For $S(b_1, ..., b_L) > 0$, the sequence b is more probable under the position-specific model, so we can assume it to be part of the same family. For $S(b_1, ..., b_L) < 0$, the sequence b is, however, more probable in the null model. This is an indication that b is not part of the protein family given by D_{train}.

For the two files `test1.faa` and `test2.faa`, you have to determine the log-odds ratio for *each* sequence. At this point, it becomes simple to decide, which file belongs to the same family of D_{train}, and which not.

D. Implementation

- First step: Read the data set (in the beginning `train.faa`, later on also the test files), and convert it into a more practical format. As an example, you may skip every odd line (the comment line), and convert the sequences from a strings of characters two a 2D array of numbers from 1 to 21.
- Second step (to be applied to `train.faa`): For each position (column) $i = 1, ..., L$ and each amino-acid $a \in \mathcal{A}$ (including the gap), determine the occurrence number $n_i(a)$ (Eq. (1)) and the weight $\omega_i(a)$ (Eq. (3)).
- Third step: For each position $i = 1, ..., L$, determine the maximum weight $\omega_i(a_i^\star)$ (Eq. (4)). If bigger than 0.5, the position is considered to be conserved.
- Fourth step: Determine the parameters $f^{(0)}(a)$ of the null model (Eq. (6)).
- Fourth step (to be applied to `test1.faa` and `test2.faa`): Determine the log-odds ratios $S(b_1, ..., b_L)$ (Eq. (8)) for each sequence in the two test sets. For the calculation, use the leftmost expression of the log-odds ratio as a sum over logarithms (the probabilities P and $P^{(0)}$ are products of many small numbers, their calculation may cause a numerical underflow). Plot the histograms of the log-odds ratios, and compare them.

III. CO-EVOLUTION OF CONTACT RESIDUES

The alignments contain information, which cannot be detected with a simple factorized model using PSWM. As we have discussed in the lecture, the co-evolution between two positions, which are in contact in the folded 3D structure, induces correlations in the amino acid occurrences in these positions (inside the protein family corresponding to the 3D structure). To detect such correlations, we have to calculate the number of co-occurrences of amino-acid pairs in the same sequence

$$n_{ij}(a, b) = \text{number of occurrences of } a \text{ in position } i \text{ and of } b \text{ in position } j \text{ in the same sequence.} \tag{9}$$

This number allows us to determine the respective frequencies

$$\omega_{ij}(a, b) = \frac{n_{ij}(a, b) + \mu/q}{M + q\mu} \ . \tag{10}$$

The pseudocount is chosen to guarantee $\sum_b \omega_{ij}(a, b) = \omega_i(a)$.

Correlations between two positions can now be quantified using the *mutual information*

$$M_{ij} = \sum_{ab} \omega_{ij}(a, b) \ln \frac{\omega_{ij}(a, b)}{\omega_i(a)\omega_j(b)} \tag{11}$$

for each position pair $1 \leq i < j \leq L$. The mutual information equals zero if and only if the two positions are statistically independent ($\omega_{ij}(a, b) = \omega_i(a)\omega_j(b)$). For each statistical dependence, M_{ij} assumes positive values.

It is now possible to compare the most correlated position pairs (highest mutual information) with the distances provided in the file `distances.txt`.

A. Implementation

- First step: The same as before, determine $\omega_i(a)$.
- Second step: For each pair of positions $1 \leq i < j \leq L$ and each amino-acid combination $a, b \in \mathcal{A}$ (including the gap), determine the number of co-occurrences $n_{ij}(a, b)$ (Eq. (9)) and the frequency $\omega_{ij}(a, b)$ (Eq. (10)).
- Third step: For each pair of positions $1 \leq i < j \leq L$, determine the mutual information M_{ij} (Eq. (11)).
- Forth step: Sort the M_{ij}, select the 1,2,4,8,16,32,64,128... position pairs of highest values M, and assign a distance to these pairs (using `distances.faa`). Determine the fraction of pairs which have a distance below 8 (i.e. the contact pairs: true-positive predictions),