Protein structure prediction

Francesca Paoletti, Adrien Deverin

January 28, 2022

1 Introduction

The objective of this project consists in predicting the 3D protein structure and comparing it with its real structure to quantify the similarity between the two models. The goal is to model the structure of the protein that carries the mutation responsible for high resistance to tetracyclin of a Salmonella variant. It belongs to a specific protein family, the Tet Repressor proteins (known as TetR), which play an important role in conferring antibiotic resistance to large categories of bacterial species.

2 MSA

Using the Pfam Database we found the associated protein family of the sequence given in input and we fetched a multiple sequence alignment for it. By giving in input to the *Pfam* database a FASTA format file, it returned all the alignments to the given protein. Multiple Sequence Alignments (MSA) of homologues proteins can provide us with at least two types of information: the conserved amino acids at certain positions and the inter-relationship between two or more positions. The chosen size for the MSA is around 13000 sequences, in order to have an accurate estimation of residues covariations.

After loading the MSA, we have removed the columns in the MSA which present more than 50% of gaps ("-"): the length of each sequence has therefore become 46.

To predict the protein structure from its amminoacids sequence, it is necessary to understand the interactions between the amino acid of the protein sequence and the function in the organism. The 3D structure of the protein is stable mainly due to electrostatic interaction between amino-acid being in contact. Therefore, if one mutates, the amino-acid in contact has to compensate the mutation: if X and Y are in contact and X is mutating in a negatively charged aminoacid, Y must have been mutated in a positive charged amino-acid.

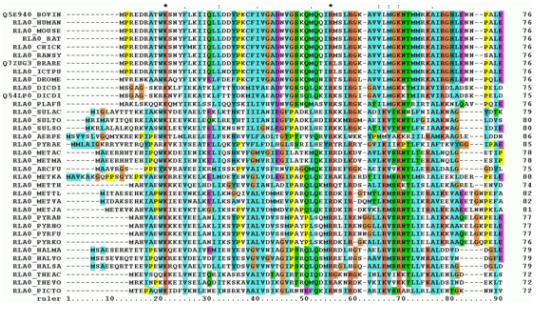


Figure 1: Multiple Sequence Alignment.

3 Mutual Information

Mutual Information from information theory can be used to estimate the extent of the mutual coevolutionary relationship between two positions in a protein family. Mutual information theory is often applied to predict positional correlations in a MSA to make possible the analysis of those positions structurally or functionally important in a given fold or protein family. Since evolutionary variations in the sequences are constrained by a number of requirements, such as maintenance of favorable interactions in direct residue-residue contacts, using the information contained in MSAs may be possible to predict residue pairs which are likely to be close to each other in the three-dimensional structure.

To infer interactions and possible contacts, by aligning multiple related protein sequences, we can compute the mutual information between the positions of the alignment.

For each position (i, j) (i.e. for each possible combination of columns in the MSA), we calculate the mutual information value, which codes how much the two columns are correlated.

So for each (i, j) the MI is defined as the sum over each couple of amino-acids of the joint probability times the logarithm of the ratio between the joint probability and the product of the two marginal probabilities.

The total number of amino-acids is 20 (alanine (A), arginine (R), asparagine (N), aspartic (D), cysteine (C), glutamine (Q), glutamic acid (E), glycine (G), histidine (H), isoleucine (I), leucine (L), lysine (K), methionine (M), phenylalanine (F), proline (P), serine (S), threonine (T), tryptophan (W), tyrosine (Y) and valine (V)) and the number of possible combinations is therefore 400.

$$M(i,j) = \sum_{a,b} p(x_i = a, x_j = b) * \log(\frac{p(x_i = a, x_j = b)}{p(x_i = a)p(x_j = b)})$$

where:

$$p(x_i = a) = \frac{\#\text{sequences that have the amino-acid a in position i}}{\text{TOT sequences}}$$

$$p(x_j = b) = \frac{\#\text{sequences that have the amino-acid b in position j}}{\text{TOT sequences}}$$

$$p(x_i = a, x_j = b) = \frac{\#\text{sequences that have the amino-acid a in position i and b in j}}{\text{TOT sequences}}$$

If a position i is independent from a position j, it means that $p(x_i = a, x_j = b) = p(x_i = a) * p(x_j = b)$, so that the argument of the logarithm becomes equal to 1 and the overall mutual information $MI(i, j) = 0$.

If a position i is fully conserved across sequences, it means that it has been maintained by natural selection and that it does not carry additional information about the mutations in position j. The corresponding value of mutual information will be close to 0.

The maximum value of mutual information is taken from sequences that co-evolve together, so that for each mutation in position i, the corresponding mutation happens in position j: if $p(x_i = a, x_j = b)$ assumes the highest possible value, the overall value of $MI(i, j)$ will be the maximum one. The maximum value of the joint probability occurs when both positions i and j co-evolve together: if there is a mutation in position i, there will always be the corresponding mutation in position j.

For the analysis, it is interesting to compute the entropy for each column in the MSA, to measure the "amount of information" contained in each of them. To perform this computation, we adopted the Shannon Entropy analysis, which is possibly the most sensitive tool to estimate the diversity of

a system. For a multiple protein sequence alignment the Shannon entropy (H) for every position i is as follow:

$$H(i) = -\sum_{a}(p_i = a) * log_2(p_i = a)$$

where $p_i = a$ is the fraction of residues of amino acid type a, and a corresponds to each possible amino-acid (20). The plot in Figure 2 represents the values of the Shannon entropy computed for each column in the MSA.

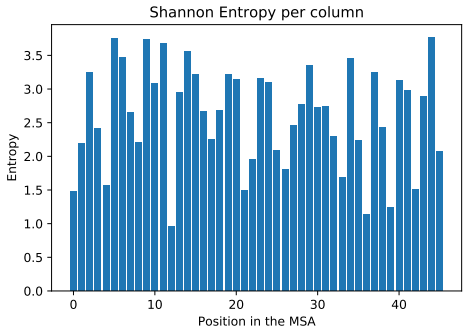
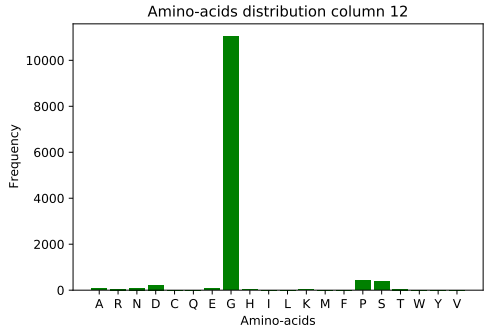
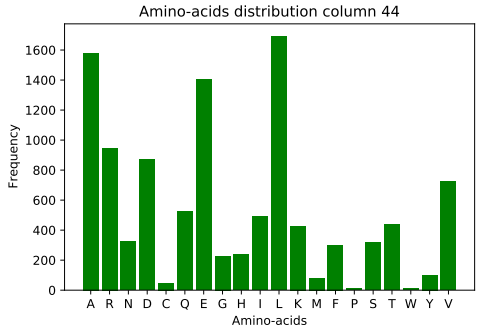


Figure 2: Shannon Entropy.

It is clear that column 12 is the one with the lowest entropy, meaning that this position is almost fully conserved in all the MSA. This can be confirmed by Figure 3a, which shows the frequency of each amino-acid in that sequence: the amino-acid **G** is conserved for 11035 sequences over 12702, which corresponds to the 87%. On the contrary, the 44^{th} column is the one with the highest entropy, meaning it assumes many different values, as shown in Figure 3b.



(a) Frequency Distribution of the 12^{th} column.



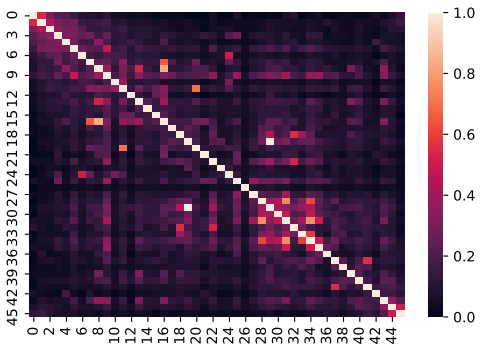
(b) Frequency Distribution of the 44^{th} column.

To compute residues it is necessary to have at least 1000 samples, in order to have quite acceptable results. Using the initial set of 12702 sequences, we computed the MI matrix, whose heatmap is reported in Figure 4a. Since, when applying the rule $M(i,j) > \tau$, some positions predict too many contacts, we introduced a correction to the mutual information matrix (Figure 4b).

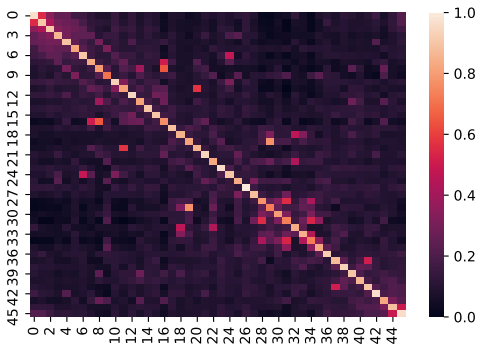
$$M'(i,j) = M(i,j) - \frac{1}{N}\sum_{k}(MI(k,j) + MI(i,k))$$

For each position (i,j) we subtracted to the original value of the matrix the mean of the corresponding row and column.

From the obtained mutual information corrected matrix, we can retrieve the contact map, which is a $\{0,1\}$ matrix, that contains the contact information between two positions in the MSA.

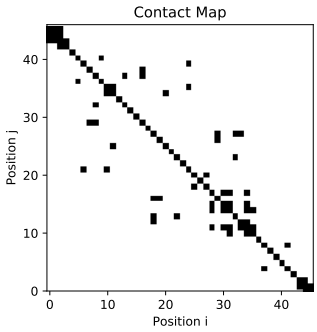


(a) Heatmap MI.

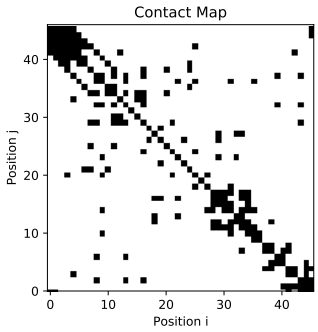


(b) Heatmap MI with correction term.

It is therefore necessary to implement a threshold such that:
`if MI(i,j) > threshold:  CM(i,j) = 1, else:  CM(i,j) = 0`
The first approach consists in adopting a naive strategy, setting randomly the threshold. With a value of 0.3, we get around 100 contacts. The corresponding greyscale heatmap is shown in Figure 5a. Decreasing the value of the threshold to 0.2, the number of contacts increase and the contact map changes as shown in Figure 5b.



(a) Contact map threshold = 0.3.



(b) Contact map threshold = 0.2.

To optimize the choice of the threshold, in order to get a value that allows to produce a 3D structure similar to the target one, there are different strategies. The adopted one consists in creating a full pipeline that from an MSA and a target structure, computes the RMSD between the predicted structure and the target one. To implement this optimization method, we retrieved the 3D structure of the TetR_N and we compared the result with the predicted structure, using the **PyMol** tool.

For different values of the threshold (between 0.1 and 0.4) of the predicted structure, we generated many different files (*lst* format) containing the information of the corresponding contact map. Using the FT−COMAR tool, and running the command `path/to/FT_COMAR my_contacts.lst 9 0 test.pdb` for each file, we obtained several *pdb* files.

To optimize the execution, we have generated a pyhton script executable directly on PyMol. For each *pdb* file, this tool generates the 3D structure and aligns it with the *TetR_N* one, getting as a result the RMSD between the two.

The RMSD corresponds to the root mean squared-deviation, which measures how far is the prediction from the truth value and whose formulation is:

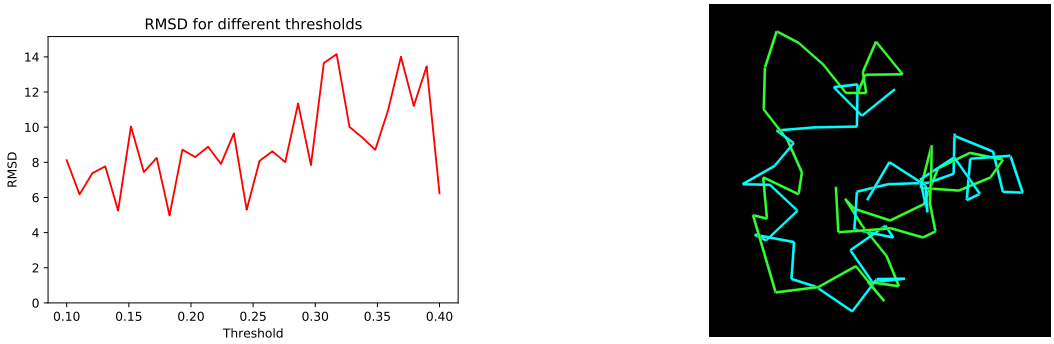
$$RMSD = \sqrt{\frac{(x_e - x_o)^2}{N}}$$

Figure 6: Alignment of the target and predicted structure.

where x_e is the expected value, x_o is the observed value and N is the total number of observations.

So that, for each contact map, we obtained the relative RMSD value, which has been plot in order to better visualize which is the threshold that corresponds to the lowest RMSD value.

What we can conclude from the plot in Figure 7a is that the optimal value of the RMSD found is 4.97 and it corresponds to a threshold value of 0.18. (Figure 7b).
Sometimes, the mirror reflection of the molecular structure could better approximate the target one. For this reason, since the FTCOMAR tool does not return both the structures, we have drawn it and generated the 3D structure using the PyMol tool to get a comparison with the other one. The RSMD obtained is higher, meaning that in this case, the first structure is the best approximation found.



(a)

(b)

Figure 7: RMSD and final result.

4 Conclusion

We have seen that the contact map is a quite good tool to make the approximation of a molecular structure. Nevertheless, our result is not 100% accurate, cause it does not take into account the internal composition of the molecule, which contains a huge quantity of information. An interesting follow-up could be to implement an algorithm to adapt the obtained structure to the internal energy. Starting from the gathered structure and using the Pfam database, it could be possible to make randoms brewing concerning the size and the rigidity of the amino acids and keep the ones that minimize the energy.