Alphafold tutorial

In this tutorial we are going to learn how to use alphafold2 using the collab fold application. Collab fold is a google collaborate notebook that enables interactive execution of alphafold2 allowing the user to tune some of the parameters of the program. You can access collab fold at the following link: https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

Besides collab fold, alphafold2 has modeled most of the proteins in uniprot and shared the models in the following database: https://alphafold.ebi.ac.uk/. During todays tutorial we will focus on the use of the collab fold notebook, but the use of this database will be handy at some moment.

Since collab fold can take some time to run (specially for big proteins) we are sharing with you the results of each execution. You will find them in the directory for today's session organized by the step of the tutorial at which we are.

Before starting with the session, let's see how is collab fold structured and what are the parameters we can tune:

- **Input protein sequence(s):** In this section of the notebook we input the protein sequence we want to model. In this part we can introduce either one single amino acid sequence, or two amino acid sequences separated by the ":" symol. In the second case we will model a protein complex. At this part of the notebook we can tune the following parameters:
    - **Jobname:** This is the name or ID that will be associated with this job.
    - **Num_relax:** By default, collab fold is going to generate 5 different models. Num_relax is, from these 5 models, how many of them we want to relax using an Amber force field. This relaxation is an small molecular simulation to optimize the conformation of the model.
    - **Template_mode:** This option enables the program to perform homology modeling using a template besides using alphafold machine learning mechanisms. You can choose between using no template, using a template choosen by alphafold in a non redundant PDB database (pdb70) or to use a custom template provided by the user.
- **MSA options:** The input for alphafold2 it is not just the sequence of the target, but a MSA of the target and homologous sequences. This is the section of notebook devoted to build this MSA. At this part of the notebook we can tune the following parameters:
    - **Msa_mode:** At this point we choose the databases at which alphafold2 will look for homologous sequences to build the alignment. These databases can be uniref (a subsection of uniprot) and the environmental database. Alignments are made using the mmseq2 program.
    - **Pair_mode:** It sets the way in which the sequences in the MSA are handled. Sequences can be paired if they belong to the same organism or labeled as independent.
- **Advanced settings:** Advanced parameters that we will not cover in this session.

It is good to understand some of the parameters involved in the program. However, all the executions that we will perform in this session will be done with the default parameters.

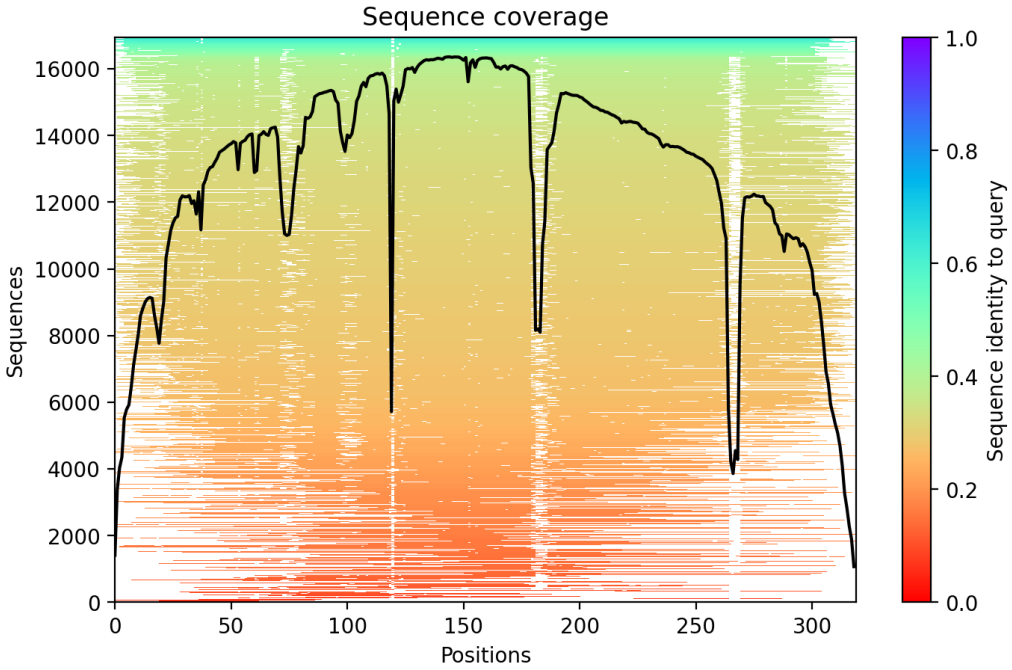
Step 1: Using alphafold with a globular protein

We will model a serine protease from Bacilus subtilis. This is exactly the same protein we modeled in practicals 4 and 5. You can find the sequence of this protein in the uniprot database with the following ID: P11018.

Collab fold is returning us 5 pdb files that are the models of our protein. Having this diversity of plots is useful because it can help us to understand the conformational variability of our protein (for example, the regions that are more variable from model to model is likely that they are more flexible).

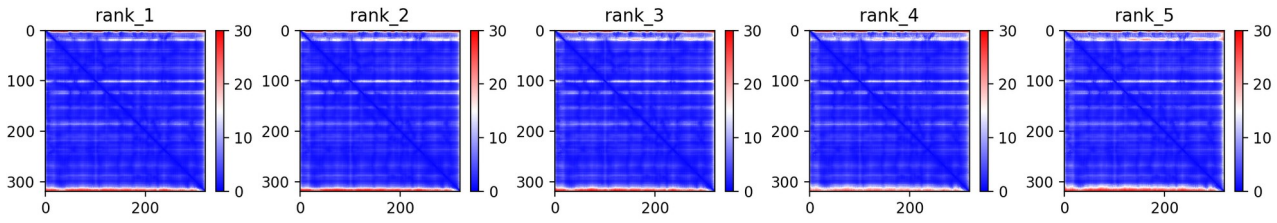
Besides that, it is providing 3 plots that can be really helpful to understand the modeling process and the quality of the models. These are the three type of plots that alphafold returns us:

- Coverage plots:



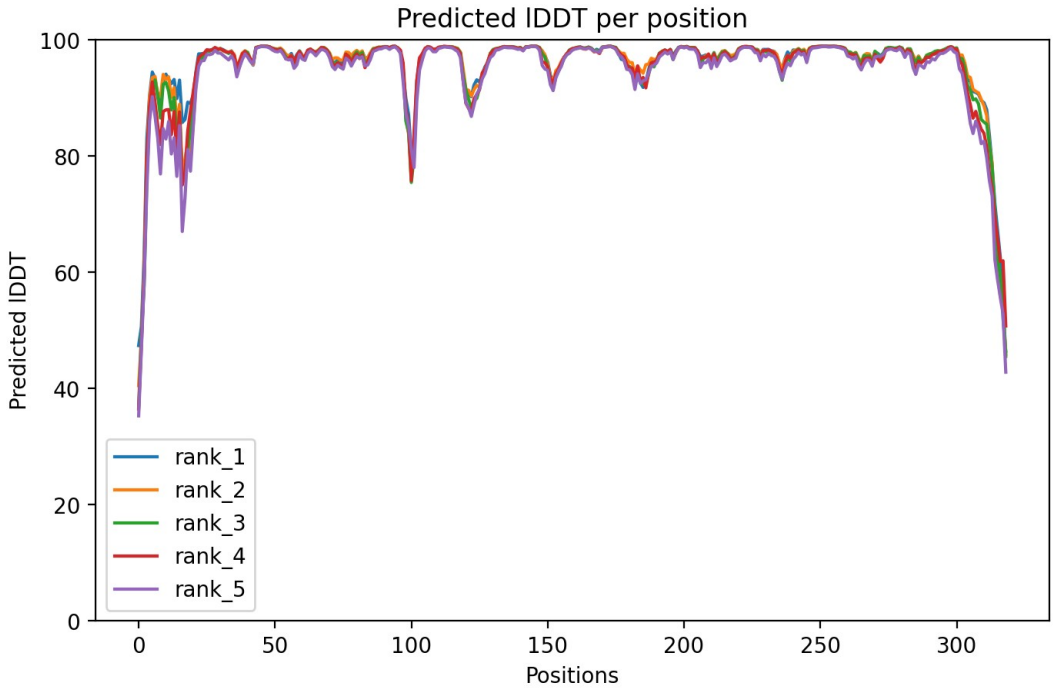
As you remember, the input for alphafold2 is a MSA. In this plot we can see the representation of all the sequences that have been used to create this alignment. The X axis represents the amino acid positions in our protein of interest, while the Y axis indicates the number of homologous sequences found for our target. See that some regions of our target have more coverage than others, as it is represented by the black line in the plot. Finally, the color of the sequences indicates the sequence identity percentage between the homolog and our target. We see that the coverage is quite high for most of the protein, with some exceptions in short regions (if you check these short regions in pymol, you will see that most of these short regions are loops).

- Predicted align error plots:



See that we have 5 plots, where each one of the plots corresponds to one of the models provided by collab fold. In these plots alphafold is showing the predicted aligned error (PAE). The X and the Y axis represent both the amino acid sequence of our target protein, and the color is indicating the predicted aligned error. This predicted aligned error is a value that says how sure is alphafold of the distance between two amino acids. In this case, the PAE is very low for the entire protein, meaning that alphafold is confident for the prediction of this model. In the next executions of alphafold we will see more complicated cases where these PAE plots become more tricky.

- Predicted LDDT per position:

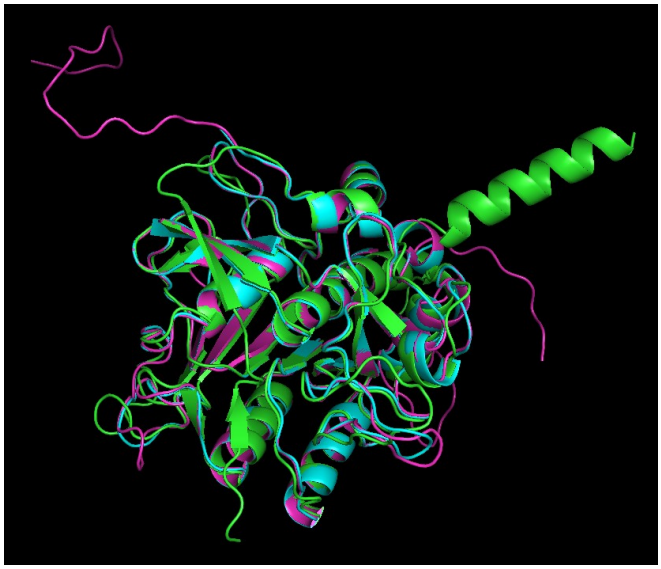


Another metric that alphafold uses to quantify the quality of their models is the predicted LDDT (Local Distance Difference Test). As it name says, this metric is a quality indicator of how good is the model at a local scale. Each amino acid has this score and we can use it to distinguish what regions of the model have been properly modeled and what regions have not. This metric is the score used in the alphafold database to quantify the accuracy of the predictions. As you can see, we have the pLDDT across in the Y axis, while the amino acid sequence is represented in the X axis, with 5 lines representing the 5 models obtained. Regarding this metric, the developers of alphafold provide the following guidelines:

- Regions with pLDDT > 90: Regions modeled with high accuracy.
- Regions with pLDDT between 70 and 90: Regions modeled with moderate accuracy. You can expect a good positioning of the backbone of the protein, but not so much for other features of the protein such as the side chains.
- Regions with pLDDT between 50 and 70: Regions modeled with low cnfidence, they should be treated with caution.
- Regions with pLDDT < 50: Usually are modeled as loops and they should not be interpreted. However, such low values of LDDT can also be a good predictor of disordered protein regions.

We can evaluate this model by comparing it to the models we created in practical 4 and 5. Since alphafold is providing several models, we will use model 1, which is the one with best metrics according to the results given by alphafold. We can do these comparisons by:

- Making superimpositions with a template: Since alphafold models have very long and complicated names, I recommend you to rename them to a more simple name. You can see that the two superimpositions look very well and have low RMSDs (modeller-template RMSD=0.165; alphafold-template RMSD=0.564).

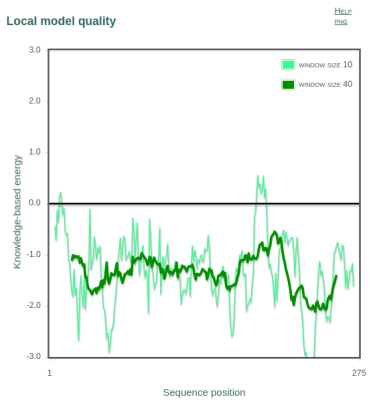


See that the region that was giving us problems during the modeling now it is different. There is a helix close to that region that is modeled as a longer helix.

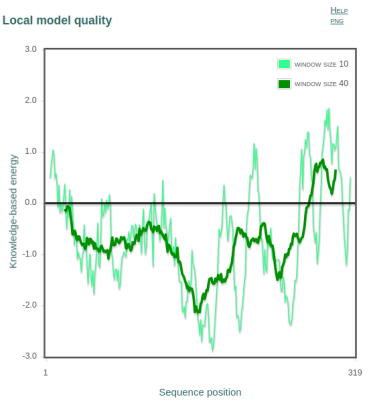
- Analyzing the structures with prosa: This is the best way to asses the quality of the models obtained by alphafold and modeller. When doing this analysis, it seems that the PDB files provided by collab fold have a format that prosa web cannot understand. To fix this, you can use the PDB model for this same protein in the alphafold database: https://alphafold.ebi.ac.uk/entry/P11018

These are the results we get for our models and template:

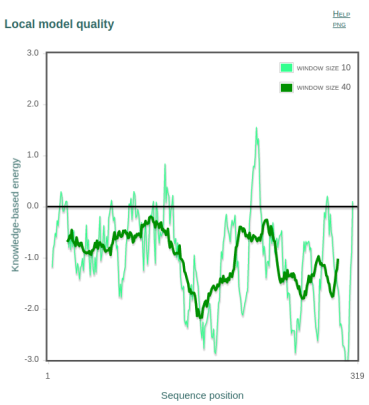
Our template (1meeA)
Z-score = -9.51



Our corrected model (S5)
Z-score = -7.26



Alphafold model
Z-score = -9



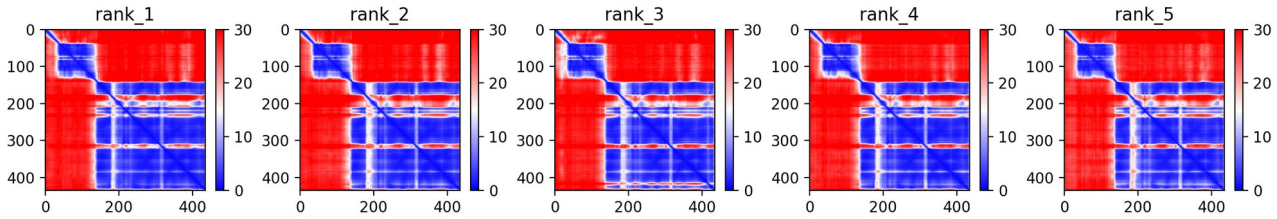
With this information, what model do you think is the best? Was using modeller worth the try?

Step 2: Using alphafold with a two domain protein

In this step we are going to model a nuclear receptor. Nuclear receptors are transcription factors that also bind hormones. This binding to hormones modulates their activity, it is one of the mechanisms by which our cells can react to some of the hormones we produce, such as estrogen.

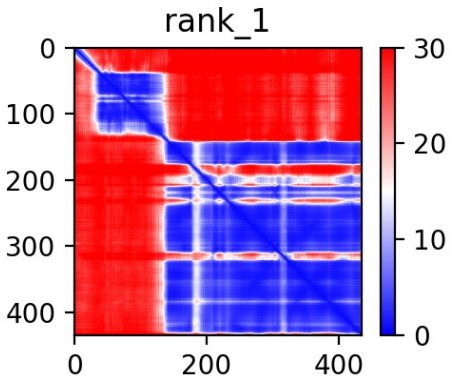
Due to this dual function, nuclear receptors must have (at least) two different domains: one domain to interact with the DNA (remember that it is a transcription factor) and another domain to interact with the hormones. These two domains are independent from each other and are linked by a flexible region of the protein, which can also be called hinge. The protein we will work with (NR1I2) can be found at the following uniprot page (https://www.uniprot.org/uniprotkb/O75469/entry) and has a DNA binding domain (amino acids 41-102) and a hormone binding domain (amino acids 146-433).

If we check the output PAE plots of alphafold we will see that we can clearly distinguish the two domains and the hinge:



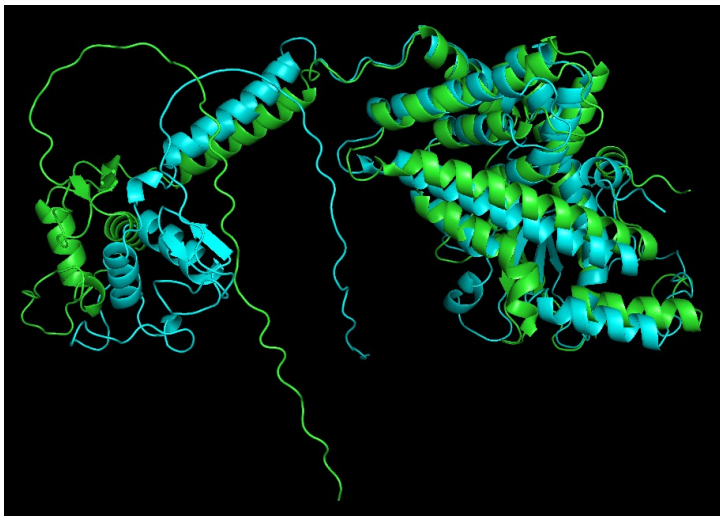
This protein provides a good opportunity to understand these PAE plots. Let's take the first plot and see what is the PAE for the following pairs of amino acids:

- Amino acids 50 and 100
- Amino acids 100 and 200
- Amino acids 300 and 300
- Amino acids 250 and 400



Here we can see how alphafold is very confident of the modeling of the two domains independently. Each one of the blue squares corresponds with each of the two main domains in the protein: the DNA binding domain and the hormone binding domain. The blue regions mean that alphafold is expecting low error regarding the distances between the amino acid positions represented in the X and Y axis. The red regions mean that alphafold is expecting high error regarding the distances between the amino acid positions represented in the X and Y axis. See that alphafold has a clear idea of where are the two domains independently, but it doesn't know where each domain goes relative to the other.

Interestingly, we obtain a similar situation if we try to superimpose the models that we obtain. Open the two first models in pymol and try to superimpose them, you should get something similar to this:



The superimposition is quite bad, the RMSD is 6.265 Å, and yet the models are very similar if we look at the plots provided by alphafold regarding its predictive performance. What do you think that it is happening?

The problem here is the hinge region linking the two domains. This hinge is flexible, and alphafold is quite bad predicting flexible regions. This makes sense, since flexible regions are hard to crystalize and they are scarce in the PDB (which is the training set used to train alphafold). Also, flexible regions are by definition flexible. Therefore, it doesn't makes much sense to try to predict a fix structure for something that is going to be under continuous change.

See that the angle that the flexible hinge takes is going to define the orientation of one domain towards the other. This makes the superimposition go wrong, but if we superimpose the domains separatelly, we will see how they superimpose perfectly. You can do so with the following pymol commands:

- To superimpose the DNA binding domain:

```
super model1 and resi 41-102, model2 and resi 41-102
```

See that now the two domains fit way better and the RMSD is smaller (0.130).

- To superimpose the hormone binding domain:

```
super model1 and resi 146-433, model2 and resi 146-433
```

See that now the two domains fit way better and the RMSD is smaller (0.161).

See that when we are superimposing the two domains separately, the RMSD that we get are extremely low, and the domain that is not superimposed is located in very different positions. The idea here is that the domains are properly modeled, but the connection and the orientation of these domains is prone to error.

Step 3: Using alphafold with a chimeric protein

Since alphafold only takes as input a sequence, we can put anything we want as input, as long as it is made of amino acids. This means that we can input proteins that don't exist in nature, but that could be created in the lab with the appropriate molecular biology tools. One of these cases are chimeric proteins.

A chimeric protein is a protein that results from the fusion of two or more proteins. It is a common procedure in molecular biology and usually involves fusing one protein of interest with another protein that we can easily detect. One classic example of a protein that can be easily detected is the green fluorescent protein from Aequorea victoria (also known as GFP). As its name says, it is a protein that can release green light via fluorescence.

We are going to model a chimeric protein made of two parts:

- The serine protease of Bacillus subtilis we modeled in step 1 (also in seminars 4 and 5).

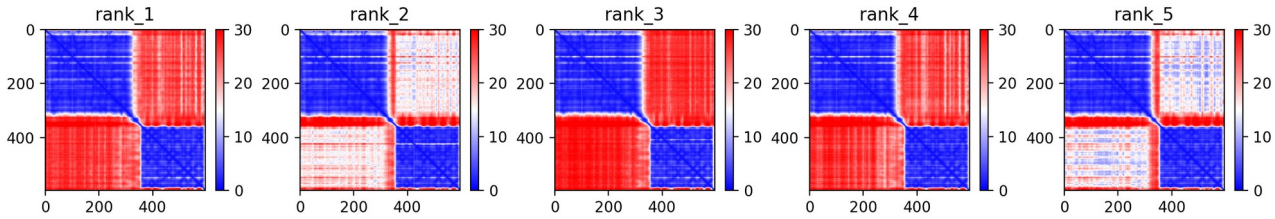
- The green fluorescent protein from Aequorea victoria.

In between the two proteins we will place the hinge region of the nuclear receptor we modeled in the previous step. This will enable that the two domains don't clash with each other and that they have space to operate independently. Thus, our input sequence for alphafold looks like this:

MNGEIRLIPYVTNEQIMDVNELPEGIKVIKAPEMWAKGVKGKNIKVAVLDTGCDTSHPDLK
NQIIGGKNFTDDDGGKEDAISDYNGHGTHVAGTIAANDSNGGIAGVAPEASLLIVKVLGGE
NGSGQYEWIINGINYAVEQKVDIISMSLGGPSDVPELKEAVKNAVKNGVLVVCAAGNEGD
GDERTEELSYPAAYNEVIAVGSVSVARELSEFSNANKEIDLVAPGENILSTLPNKKYGKLTG
TSMAAPHVSGALALIKSYEEESFQRKLSESEVFAQLIRRTLPLDIAKTLAGNGFLYLTAPDEL
AEKAEQSHLLTLKKEMIMSDEAVEERRALIKRKKSERTGTQPLGVQGLTEMSKGEELFTG
VVPILVELDGDVNGHKFSVSGEGEGDATYGKLTLKFICTTGKLPVPWPTLVTTFSYGVQCF
SRYPDHMKQHDFFKSAMPEGYVQERTIFFKDDGNYKTRAEVKFEGDTLVNRIELKGIDFK
EDGNILGHKLEYNYNSHNVYIMADKQKNGIKVNFKIRHNIEDGSVQLADHYQQNTPIGDG
PVLLPDNHYLSTQSALSKDPNEKRDHMVLLEFVTAAGITHGMDELYK

The amino acids without background color belong to the serine protease, the ones with yellow color belong to the hinge, and the ones with green background belong to GFP.

If we check the PAE plots that we obtain from this modeling, we see a similar pattern to the one in the previous step: the individual models are properly modeled, but the connection between them is not.



We can further prove this idea by superimposing independently each domain of the resulting model with a template:

- For the serine protease region we can superimpose on the chain A of 1mee, the template we used in seminar 4 and 5:

```
fetch 1mee
```

```
super model1 and resi 1-319, 1mee and chain A
```

See that we obtain a very good RMSD value (0.556).

- For the GFP region we can superimpose on the chain A of the 1gfl PDB structure, which is the structure of a GFP protein:

```
fetch 1gfl
```

```
super model1 and resi 350-595, 1gfl and chain A
```

Again, we obtain a very good RMSD value (0.381).

Step 4: Using alphafold with a protein with poor structural representation

Remember that alphafold is a program that requires a training to provide predictions. If it has to model a protein that is different from what it was in the training set, it is likely that it will model it wrong. This is the case for the Saccharomyces cerevisiae Sec3 protein, which we are going to model in this section of the tutorial.

Sec3 belongs to a huge protein complex called the exocyst. This protein complex is involved in the transport of vesicles within the cell and it is fundamental for vesicles to fuse with membranes and release their cargo to the extracellular space. There is only one structure of Sec3 in the PDB. It was obtained in 2017 using a technique called Cryo-EM, and in that experiment they obtained the entire structure of the exocyst. Cryo-EM is an excelent experimental technique to determine the structure of big protein complexes, but it has a drawback: the structures generated by this technique have low resolution (although they are getting better). Besides, in the structure of the exocyst, Sec3 is not complete, actually most of the protein is missing.

Now we are going to challenge alphafold to predict a protein for which there is no entire structure in the PDB, and the only partial structure available has low resolution. After obtaining this model we will evaluate its quality by superimposing it to he available Sec3 structure and by making a prosa analysis.

- Superimposition:

```
fetch 5YFP
```

Now remove all chains but chain A, which is Sec3. Also, we recommend you to rename the alphafold model to a simpler name, such as model1. Then you can perform a first superimposition with the super command:

```
super model1, 5YFP
```

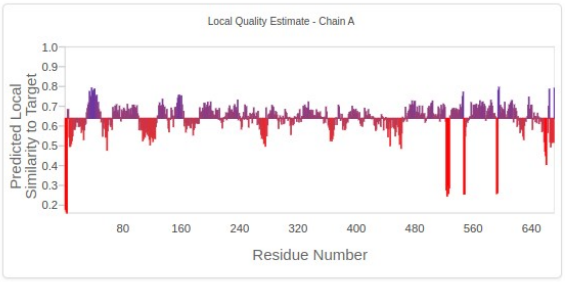
See that this superimposition is quite bad, with an RMSD of 9.860 Angstroms. Since the two proteins that we are superimposing are the same and have identical sequence, we can also use the align command:

```
align model1, 5YFP
```

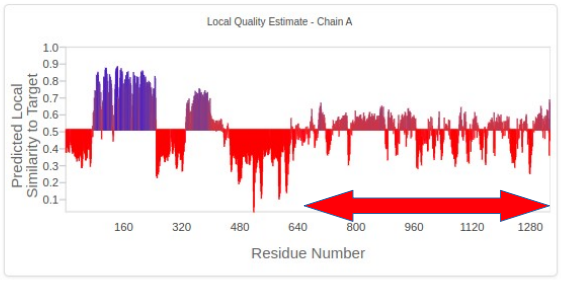
However, the align command provides an even worse superimposition with an RMSD of 10.470 Angstroms. From these superimpositions we can conclude that the alphafold model doesn't resembles the experimental structure.

- Statistical potentials (QMEAN): Prosa-web is not working for the alphafold models, that is why we will make this analysis with QMEAN, another program that uses statistical potentials to score how reliable is a protein structure. See that the statistical potentials values have been reescalated to a 0 to 1 scale, where 1 means good quality and 0 bad quality. The results for the experimental structure can be seen here: https://swissmodel.expasy.org/qmean/tB6Apb. The results for the alphafold model can be seen here: https://swissmodel.expasy.org/qmean/LUZjUA.

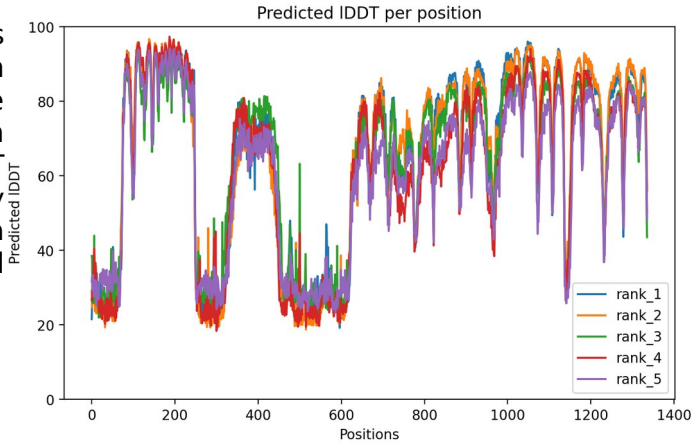
Experimental structure (5yfp, chain A)
Global quality: 0.64/1.0



Alphafold model
Global quality: 0.51/1.0



Here the superimposition and the analysis with statistical potentials have consensus in their results: both of them indicate that the alphafold models are not correct. We can also get this idea by checking the pLDDT plot, that looks quite bad. If we pay attention, we will see that there is a correlation between the QMEAN score and the pLDDT provided by alphafold:

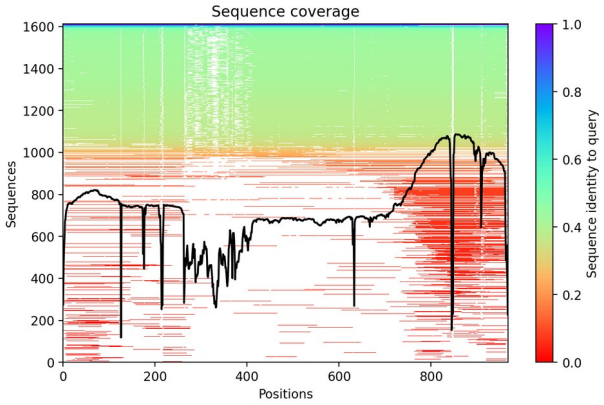


Step 5: Using alphafold on transmembrane proteins

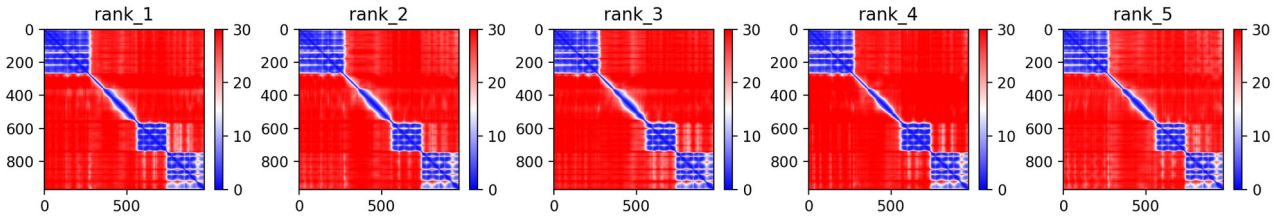
One of the main limitations of alphafold is that is not taking into account biological knowledge of the proteins that is modeling. Aspects such as the function of the protein or where is this protein located within the cell are not considered to make this modeling. This becomes a clear problem when modeling transmembrane proteins. In this step we are going to model the protein Sla2 from Saccharomyces cerevisiae. This is a huge protein with a transmembrane domain in the N-terminal region.

If we look at the plots provided by alphafold, we will get hints that the modeling didn't go as well as we would like:

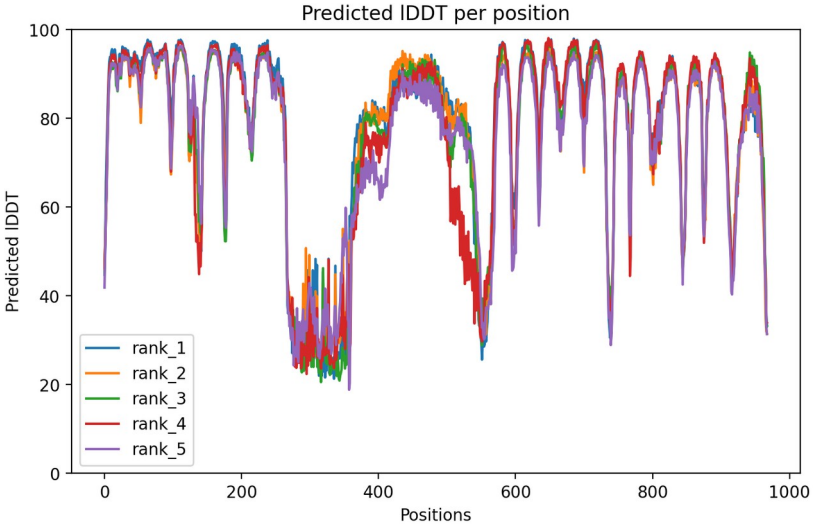
- Coverage plot: We see that the sequence coverage for most of the protein is not as high as it was for the other proteins we have analyzed.



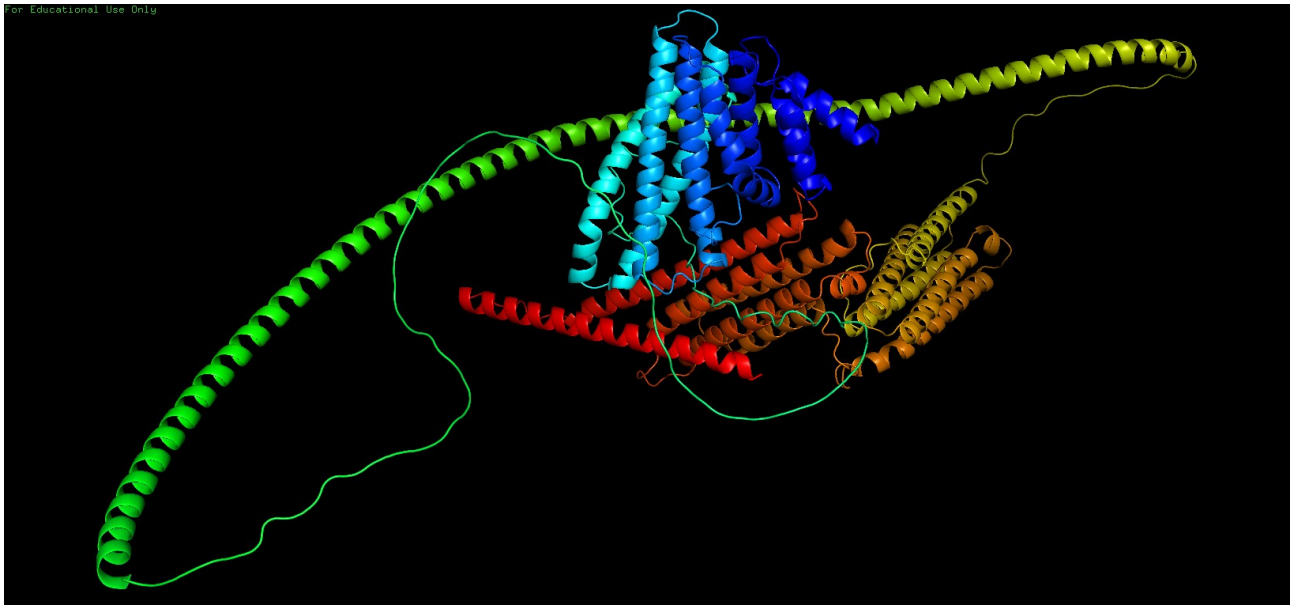
- Predicted align error plots: Most of the plots are in red, with small blue regions indicating folds that have been properly modeled. There are regions of the protein where we only see a blue diagonal, this means that the modeling in this region is not reliable.



- Predicted LDDT per position: Some regions have high values of pLDDT, but also we find regions with extremely low Plddt. Regions with low pLDDT are not reliable (or are intrinsically disordered). Usually these regions with low pLDDT are loops, and if they are not properly modeled they are going to affect the whole model. This happens because loops can turn, while helices and sheets can't. Therefore, the loops determine the orientation of sheets and helices within the structure. If the orientation of a loop is not correct, the helices or sheets that are connected to it will not be correct either.



Also, if we know that in the N-terminal there is a transmembrane domain, we can visualize the protein with pymol and try to identify at what position the membrane would be. In the following image you can see the model of Sla2 with rainbow coloring going from blue in the N-terminal to red in the C-terminal. In the center of the image we see the N-terminal in blue with several helices that could constitute the transmembrane domain. If this blue region is the transmembrane domain, do you think is there anything wrong with this model?



As you can see, if this model is alright it means that the huge helix we see is inside the membrane, parallel to the surface, and this doesn't happen in nature. The point here is that alphafold is not assuming that there is a membrane somewhere in the structure, and since it is missing this information, it is allowing the model to roam free in regions where a membrane should be placed. This is not a dramatic error, since loops are flexible, we could obtain a more suitable conformation using a molecular dynamics simulation.

On the other hand, the long loops in this model that correlate with low pLDDT values suggest that this model is not fully correct, and that these long loops should not be trusted. We can see a long loop from amino acid 270 to amino acid 357 that correlates with very low values of pLDDT. Also, the long helix that comes after (from amino acid 357 to amino acid 546) has lower pLDDT values than most of the domains that we have predicted correctly during this tutorial.

This point suggests that for transmembrane proteins, alphafold is not the best option and experimental methods such as X-ray crystallography can still provide structures that wouldn't be available without it. Also, if templates are available, traditional homology modeling methods such as modeller are still useful.

Step 6: Using alphafold with an intrinsically disordered protein

Intrinsically disordered proteins (or protein regions) don't have secondary structure. They are just long loops which are very flexible and variable. In some cases, these proteins can adopt a specific protein folding, but they require the interaction with other proteins for this to happen.

Intrinsically disordered proteins are hard to study. Since they are in continuous motion, they cannot be crystallized, and the main experimental method to characterize them is nuclear magnetic resonance (NMR). Also, when we try to model them, most of the times we get long loops (which is the same that you get with modeller of with alphafold when the modeling goes wrong).

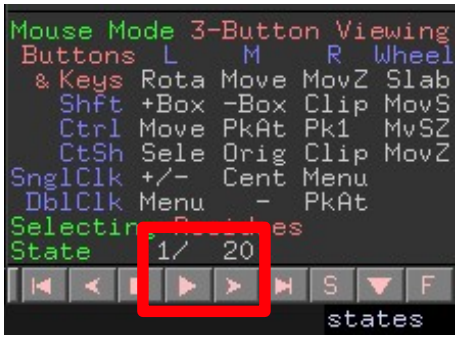
Now we are going to model the thylakoid soluble phosphoprotein (TSP) from Spinacia oleracea (Spinach). This is an intrinsically disordered protein for which we have an available structure in the PDB obtained by nuclear magnetic resonance (https://www.rcsb.org/structure/2FFT).

NMR structures provide an ensemble of structures. We can see that this ensemble is very diverse in the case of TSP, as its loops have a lot of freedom to move. We see something similar in the models obtained with alphafold. There is only one region that doesn't change: a small helix located at the center of the protein.

We are going to analyze this experimental structure with pymol and then compare it to the models of alphafold:

```
fetch 2fft
```

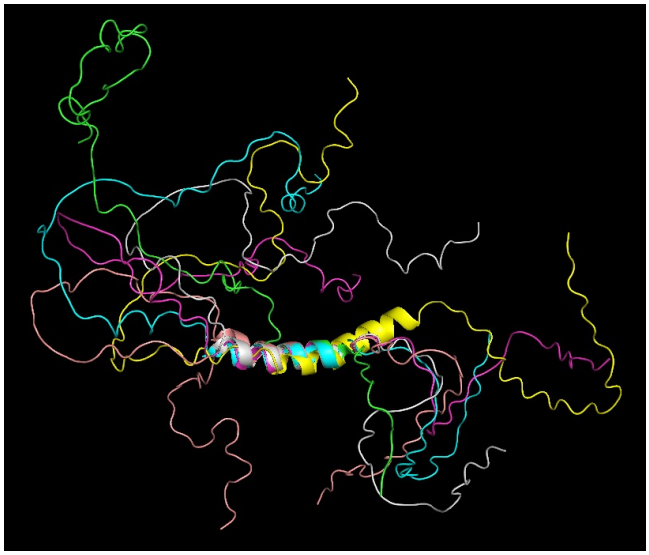
You can see the different conformations detected by NMR bi clicking into the play button you have in the low right corner of your screen. In total, 20 conformations have been obtained for this structure.



Now, open the alphafold first model and rename it to model1. Then superimpose it to the experimental structure:

```
super model1, 2fft
```

As you can see, the only regions that are superimposing properly are the helices at the center of the protein. If you load the 4 other models generated by alphafold you will see how their central helices superimpose perfectly in the experimental structure, while their loops adopt random conformations. Load the 4 models left and superimpose them to the experimental structure, you should get something like this:



Interestingly, alphafold is predicting this structure correctly, because it is a disordered protein. The problem is that it is hard to know when alphafold is predicting a disordered protein correctly and when it is doing so because the model it makes is wrong. In both situations the confidence values are very low. That is why, the use of NMR and other experimental structural methods can be useful in many scenarios despite the huge success of alphafold.