**AI in Drug Discovery and Development**

**Prof. Rajnish Kumar**
**Dept. of Pharmaceutical Engineering and Technology**
**IIT-(BHU), Varanasi**

**Week-04**
**Lecture-20**

Welcome to the course "AI in Drug Discovery and Development." In today's session, we will see how we can predict a protein structure using different tools such as AlphaFold or ESMFold. In the theory lectures, we have seen that protein prediction is one of the biggest challenges, which was, you know, partially solved by AlphaFold. and other tools which have been becoming very popular in predicting protein structures. And earlier, the protein structure predictions were done using, you know, homology modeling, where we were trying to search for similar proteins. And on the basis of the similarity in the sequence, the secondary structure or tertiary structure of the proteins was predicted.

And then there are other methods as well, like threading or, you know, ab initio modeling, where the protein structures were predicted from sequence. But AlphaFold, which uses deep learning, and other folds like RosettaFold. So, they, you know, transformed the way protein structures are predicted. In this hands-on session, we will see how we can use AlphaFold to predict the structure of our proteins.

So, AlphaFold has been implemented as ColabFold for use on the web, actually. So, in this case, you can go through this paper, Colab fold, making protein folding accessible to all. So, nature's methods paper. So, the authors have created a Colab notebook where one can predict the protein structure of their interest by just entering the sequence of the protein. So, why do we need to predict protein structure? Because we might be interested in knowing the structure of a protein to utilize its structure for structure-based design, that can be one reason.

And then there are other reasons, like we wanted to see, for example, what the difference will be between a wild type and a mutated protein. For example, if you have identified some mutations in a protein, through sequencing, we wanted to see how that mutation can affect the structure, dynamics, or functions of the protein. So, for that, we sometimes need the protein structure prediction tools. So, and this ColabFold can be utilized for that. So, how does it work? So, at the back end, it is using AlphaFold 2 for solving the protein structure.

So, what it can take is this: it takes the protein sequence in the input sequence either via the web browser, you know, either Chrome or Internet Explorer or Safari; it does not matter

which web browser you are using. Or you can use it in the command line as well; you can use it as a standalone. You can download this tool, and then you can use it in the command line. Or you can also use AlphaFold, which has been implemented in the form of ColabFold in tools such as Chimera as well. In Chimera itself, you can utilize this Colab fold to predict the protein structure directly from the sequence.

So, once you have this input in the form of protein sequences. So, then the next step is the search where MMSEQ2 searches units of 100 and the environmental database using profile-based iteration. So, it searches for similar, you know, proteins, and then an MSA generates a multiple sequence alignment, which is one of the most important steps in predicting the protein structure. So, there are like two MSAs in A3M formats that are generated. And then the next step is the structure prediction.

So, for structure, ColabFold can do both: one is, you know, a single chain where you have just a single protein and the structure of the protein you want to determine. And then you can also predict the complex as well, like if you are interested in how two proteins are interacting with each other. So, how do they make the complex so that you can also use the AlphaFold with this using Colab? So, here in the single chain, the filtered MSAs are input to AlphaFold 2, which are ranked by the PLDDT, and PLDDT is the predicted local distance difference test, which tells how much confidence there is in the prediction. So, if this PLDDT score ranges from 0 to 100 and if it is above 90. So, it means that the structure predicted by AlphaFold is highly confident.

And if it is more than 70 and less than 90, it means that in that range the backbone prediction is correct. Because you know the protein is made up of the backbone atoms, and then it has the side chains, right? So, whenever you are looking at this PLDDT score. So, if the score is between 70 and 90, it means that the backbone is predicted correctly; there is very high confidence in the backbone. However, the side chains that you know are not predicted to that high confidence level. And a score of less than 50 is usually considered bad because it indicates a lack of confidence in the structure, whether or not it is well predicted.

And then, you can see here, for example, it has this; you know, for a single chain, it produces several models, and those models are scored by the pLDDT score, okay? And then you have the multimer complexes between different chains, so it again produces several, how to say, several models. And then those are scored by the predicted TM-score, okay? And then it visualizes the output as well, like the MSA diversity or the PLDDT and PAE, which are shown to assess the prediction confidence. So, this is how you know the CoLab fold works. So, now we will see how we can use it to predict the structure of our protein. So, now let us open this link in our browser, okay? So, now that you have opened

this link in your browser.

So, what you will see here is this CoLab fold version 1.5.5 alpha fold 2 using MMSeq-2. So, it is an easy-to-use protein structure and complex prediction using AlphaFold2 and AlphaFold2 Multimer. Sequence alignments or templates are generated through MMseq2 and HHsearch.

For more details, we can go to the bottom of this notebook and check the Colab GitHub account as well as the Nature Protocol which I just shared with you. So let us just quickly go through it; here you can see that we have to give the input sequences. So, it is pretty simple; actually, the only thing you have to do is type your sequence here. So, you can just enter your sequence here in this case, and then after entering your sequence, you can give it a job name. You can share the name of the job here, whatever your protein is, and whatever your project is; you can give that.

And then you can just go to the runtime, and then you can click on the run all, okay? So, then what it will do is first install the dependencies, and after that, it will use the MSA options. One of the possibilities here is that you can also upload your custom MSAs as well. So, for that, you just click on this one, and then you can see that you can click on custom, and then you will have the possibility to upload the MSA as well, multiple sequence alignment for your research project. Otherwise, you can just use one from the UNIREF, or you can go after the single sequence as well. And then there are some advanced settings, like the model type, which model type you want to use.

Here, you can see the auto; it will automatically identify which model is suitable for your research. For your sequence, what you can do here is go with AlphaFold 2 or DeepFold version 1. Or AlphaFold 2_PTMs, which is for post-translational modification, because recently they have released a tool for predicting proteins with post-translational modifications as well. And then you can either use the multimer where you are or if you are predicting a protein structure with more than one chain. So, you can go after this multimer as well.

Multimer 1, multimer version 2, and version 3. OK, but the best way is to just keep it on auto. You don't need to choose any of that, you know. It will automatically select whatever is best for your project or problem. And then you don't need to change this number of recyclables as well.

So, if auto is selected, it will use the number of recycles 20. If the model type selected is Alpha Fold 2 Multimer version 3, then the default value is 3. And then there is recycle every stop tolerance. So, if auto is selected, it will use a tolerance of 0.5 if the model type

is AlphaFold to multimer; otherwise, it is 0.

0. And then relax maximum iterations 200. So, it is maximum amber relaxation; you know, unlimited alpha fold to default can take very long. And then, the pairing strategy uses the greedy method to pair any taxonomically matching subsets, or if you are using complete, then all sequences have to match in one line. And then, you know, sample settings enable dropouts and increase the number of seeds to sample predictions from the uncertainty of the model and decrease the max MSA to increase the uncertainty. So, max MSA should also be kept on auto so that it can pick the most suitable settings for your problem.

And then you can also save all those details in the settings. And then the next thing is to run the prediction, so when you click on this, it will go through that MSA and process your sequence. And then identify which proteins have a similar sequence to your protein, and based on that, it downloads the weights from AlphaFold to a model. And then, using that, it predicts the structure and gives it along with the PLDDT score as well. And then you can display the 3D structure, and then you can display the plots, which are again the PLDDT score or the TM score if you are using it for the multimer as well.

Okay. And then finally, you can download all these results because, as I said earlier, if you are using it in Colab. So, once you are disconnected from the Google server, all data will be gone because it does not save your data locally. So, the best thing is to always download all your data, no matter what you are doing on Colab. So, with this, by running this cell, you download all the data that is generated using this Colab notebook.

Okay. And these instructions are given here. So, these are exactly the same, you know, as whatever we discussed. And then the resulting zip file contains the PDB formatted structures sorted by average PLDDT, and complexes are sorted by PTM score, unrelaxed and relaxed if you are using Amber. And then it also contains the plots of the model quality, the plots of the MSA coverage, the parameter log file, and the A3M formatted input MSA. and a predicted aligned error version 1.

0 JSON file using AlphaFold database format and a score dot JSON for each model, which contains an array list of lists for PAE, a list with the average PLDDT, and the PTM score. A BibTeX file with the citation for all used tools and databases is very important whenever you are using these tools, so you have to properly cite them. At the end of the job, a modal box will pop up with a job name results.zip file; additionally, if the "Save to Google Drive" option was selected, the job name results.zip will be uploaded to your Google Drive.

So you can always, you know, save it to your Google Drive as well. For that, you have to mount your Google Drive, and then it will save results into your Google Drive as well.

Okay, so yeah, this is some detail about how MSAs are generated for complexes and how to use the custom MSA. And about the PDB 100 database, which is being used in prediction, and then how to use the custom templates. Okay, some troubleshooting as well where we check that the runtime type is set to GPU at runtime; change the runtime type because it is using, you know, deep learning, so it is better if we use GPU, as it will be faster.

And try to restart the session, we go to runtime and factory reset runtime. And if there is a problem, we need to check the input sequence as well. And there are some known issues, like Google Colab assigning different types of GPUs with varying amounts of memory. Some may not have enough memory to predict the structure for a long sequence. And the browser can block the pop-up for downloading the results file, so you can choose to save it to Google Drive.

Instead of manually downloading the results file, click on the little folder here and then download the file. So, the limitations are like computing resources; their MM-Seq2 API can handle only 20,000 to 50,000 requests per day. In the MSA, the MM-Seq2 is very precise and sensitive, but might find fewer hits compared to the HH-BLIT or HMMER search against BFD or Magnify. So, they recommend additionally using the full AlphaFold in the pipeline. And then the description of plots is also given: the number of sequences per position; we want to see at least 30 sequences per position, and ideally 100 sequences for best performance.

And then predicted IDDT per position, the model's confidence out of 100 at each position; the higher, the better. And the predicted alignment error for homo oligomers, this could be a useful metric to assess how confident the model is about the interface; the lower, the better. Okay, and then the license information is also given, and finally, acknowledgment. This is how the AlphaFold, you know, the Colab Fold is working. So now let us try our hands at predicting a protein structure by using this Colab Fold.

So, for that, we will be using the small structure; we will be predicting a small structure, and that structure is coming from, you know, E. coli bacteria, and this is a structure of the efflux transporter EMRE1. So, you know that the efflux transporters play a very important role in bacteria by making them resistant to antibiotics. So, when the antibiotics enter the bacteria, the efflux transporters push the drugs out of the bacterial cell. So, for that, it is important that this structure is already resolved using structural NMR, so you can see the NMR ensemble here.

And you can also have a look at the structure in 3D as well, so if you click on "Explore in 3D" and "Structure." So, you can see here, and then you can see it is actually a dimer, okay?

So, it is a dimer containing chain A and chain B. So, what we will do here in this case is just pick up the sequence. So, what we will do here is take up the sequence of this protein, and then we will predict the structure of the monomer first. And then for the dimer, see how well the ColabFold can predict the structure.

Okay, since the structure is already known, of course the prediction will be highly confident, but let us check it. So, for that, in order to get the sequence, we need to go to the sequence here and click on it, so you can see the sequence for chain A and then for chain B as well. And one more thing is that it is a heterodimer. Okay. So, it means that both the chain, chain A, and chain B are different.

So, usually the homodimers have the same chains, but this is a heterodimer. So, it has a different sequence for each chain. And then we have to click on the displayed files, and we go for this FASTA sequence. And when we click on this, you can see we get the sequence for chain A and then the sequence for chain B. So, what we have to do here is click on the sequence for chain A and then copy it.

And once we have copied it, we go back to the Colab folder here, and then in this query sequence, we just paste it. So, we remove this sequence, and then we paste it. Okay, and then we can give a job name since the prediction takes time. So, I already ran this, you know, Colab for demonstration purposes. But what you have to do is paste the sequence and then give it a job name.

Like here, I have given EMRE, and then after that, you just have to click on run time and then run all, okay? After pasting the sequence and giving it a job name, you click on the run time and then run all. Since I have already run the job, I will not run it again because it will take some time. So, what it will show you here is the job name EMRE_8fdca, which might be different for you because every run is different. And then it will show you the sequence that you have entered, and it will also show you the length, which contains 110 amino acid residues. So when you run the runtime, it will execute all the cells, and in the next cell, you will see that it will install the dependencies.

And then you can see here it is; once it is done, it will show "installing colab fold," and then it will show the CPU times: user 138 milliseconds, system this, and blah blah blah; all time was 43 seconds. And then it will run the next cell, and then it will just pick up MSA. And if you are using the custom MSA, then you have to upload the custom MSA yourself. And then there are some advanced settings; you don't need to alter those.

And the next cell runs the prediction. So, when it is run, what you will see here is that first it will download the AlphaFold PTM weights. So, it will download the weights. So, once

it has downloaded the weights, it might take some time because it's quite a heavy file. So, then it will show you running on GPU because we are already running it on GPU, and you can check that from here as well. To change the runtime type, go to Runtime, and you can see here that Google has assigned us a T4 GPU.

So we are using that for it, and that is why it will be faster as well. And then you can see that it has shown that the sequence coverage actually looks like this: this is the sequence coverage plot. So, this is sequence coverage; here on this left side, you see the sequences, and then here you see the positions. Our sequence, which we input for the prediction, contained 110 amino acid residues.

So, these are 0 to 110 amino acid residues for that protein. And then you can see that it has evaluated more than 10,000 sequences. So, it could, you know, find the coverage, actually, how much this input sequence is covered by different sequences. And here you can see the scores. So, the red one shows 0, and then the cyan one, or the blue one, is showing more than 0.

8. So, after that, you see that these are the predicted models. So, it is colored from N to C, N terminal to the C terminal, and this one is colored by the PLDDT. So, this one is colored by PLDDT. So, if you look at this, you can see that the red part is, you know, less confident, okay. The green part is highly confident, and yellow is actually in between.

So, it means that this protein contains 4 alpha helices. So, these 4 alpha helices are predicted quite well. However, the loops and the terminal parts, C and N terminal residues, are predicted with a little bit less confidence. So, after that, you can just display the 3D structure. So, it will display the 3D structure, and then you can see that you can have all these rank numbers as well. And which structure to display, you can do that here, and then you can color them with the IDDT, or rainbow, or the chain.

Okay. So, it will show it like this, and then you can interact with it as well; you can rotate it. And then here you can see that PIDDT is okay. So, it is like it is red. So, it is very low, less than 50, and as we discussed, if it is red in this case, you can see here that the terminal amino acid residues are predicted not very well.

So, the structure is actually flexible. They might be unstructured as well. They do not need to be structured because many proteins they have are unstructured as well. And then the yellow is showing low, at 60. For example, in these loops, you can see that this loop is shown in yellow, which indicates that the PIDDT score is 60. And then the green one is okay, and then the cyan one is like confident, which is 80.

So, you can see that those alpha helices are predicted with very high confidence, okay. And then, if it is purple, then it is very high, but this structure is, you know, reliable, actually. Okay, and then it also plots all those, you know, for all five of these models that it has predicted. So, it plots all these things as well, like the sequence coverage, the positions, and the predicted IDDT. So, you can see here all those positions and how much their prediction score is also being given.

And then finally, it will automatically download all the data here, and then you can open those PDB files in some other visualizer, a structure visualizer, or some other programs or tools as well; you can analyze them fine. So, this is how we can predict a monomer structure, okay? And then going back to that structure. So, in this case, we have a heterodimer, and then this heterodimer had a chain B, which was this. So, let us predict the heterodimer structure as well.

So, first we have to copy the chain A again. So, we copy the chain A, and then we paste it here in this cell, okay? And then we go back and copy the chain B, okay? And then once we have copied the chain B, how will we specify it? So, it is specified by a colon, actually. So, you just add a colon, and then you just add these double dots, actually, and then. You paste the sequence B, okay for the sequence for chain B, right? Yeah, so once you have pasted the sequence for chain A and chain B. So, the next thing is you give it a job name like EMRE multimer, and then you just go to runtime and run it all okay.

It is the same as we did for the monomer; we do it for the multimer. The only difference is that whenever you can see here, it has specified inter-protein chain breaks for modeling complexes that support homo- and hetero-oligomers. For example, PISK double dot to PISK for a homodimer. So, you just have to add these double dots between these two chains, and then it will automatically recognize that these are two different chains. And with those chains, it will automatically make a, you know, a multimer, actually, whether it is a dimer, trimer, or tetramer.

So, based on your input sequence, it will automatically predict the structure. Okay, so in this case, again you go to runtime and then run all. So, you can see here, since I said it would take some time, I didn't; I already ran it before. So, I'll just walk through it. So, then you can see here that the job name is this and the sequence is given here.

You can see the double dot, and it has given the length of 220 now. So, there are like 110, 110, two chains. So, the total length is 220, and then it will install dependencies, MSA options, advanced settings, and finally come to the predictions. So, since I started it, you can see that it has. So, what it will do first is download the weights again.

So, it has used the alphafold2_multimer_v3 weights here. So, you can see the difference, like in the earlier one when we were running the monomer. So, it used the, you know, the alphafold2_ptm weights, I think. And here it is, you know, using it is the multimer weights because it has detected that we are now predicting the multimer. So, for that, it has used the suitable weights. Ok, and then you can see here the sequence coverage as well; like this is for, you know, the chain A and then chain B.

And then this is the sequence identity to the query, colored by colors; this is the color map, and then you can see that most of the, you know, part is being covered by the sequences that have been searched, okay. And then you see the models; for example, this model is colored by chain, and this one is colored by the PLDDT. So, when you look at this PLDDT color, So, you can immediately see that the alpha helices are again similar in the case to what we saw in the monomer case. So, these are predicted with high confidence: the alpha helical structure and the terminal part, especially the C-terminal and N-terminal. So, these are predicted with a very low confidence, and the loops are also predicted with a little bit of low confidence.

Because loops are usually flexible, you cannot predict them very reliably, as they can exist in multiple configurations or multiple orientations. So, you get all these, you know, models, and then you can quickly look at them in 3D as well. So, now you can see we actually have a complex. So, earlier we predicted the monomer structure, and now we have this complex made up of two different chains, chain A and chain B, okay. And then the colors here are colored by the PLDDT, indicating whether the structure is reliable or not, and which parts of the structure are reliable that you can see.

And then you can have these maps and plots as well. So, here you can see that the terminal part is predicted a little bit with less confidence. However, all those five predictions were, you know, quite good in this case, while in the monomer case, some of them had a low prediction score. Okay, and then finally you download them. Okay, so this is how you can predict both the monomers and multimers using Colab Fold. So, this is a very easy to use and highly reliable resource that can be used for predicting protein structure.

And as I said, one of the possible problems can be if you want to know how a mutated protein is different from the wild-type protein. So, in that case, it can be used very efficiently. So, in this case, we just enter your wild-type sequence, and then you enter your sequence with the mutation. For example, you have, you know, this amino acid residue mutated with an alanine.

So, you just replace it and then you run the prediction. So, that you can see how that mutation is affecting the secondary structure those alpha helices or beta sheets or the loops

or leading to the change in the functions of those proteins. So, I hope that you will utilize this tool for your research and for learning as well. And with that, thank you.