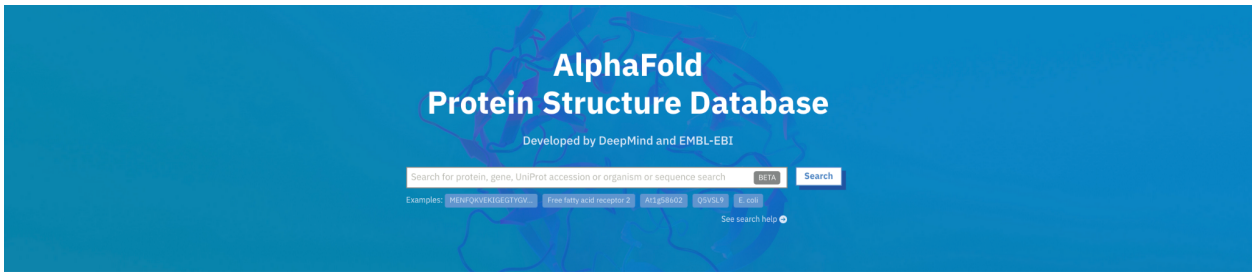
Alphafold2 tutorial

Useful links:

AlphaFold database	https://alphafold.ebi.ac.uk/
Colabfold MMseqs2 colab page	https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb
AlphaFold colab page	https://colab.research.google.com/github/deepmind/alphafold/blob/main/notebooks/AlphaFold.ipynb
ColabFold local Github page	https://github.com/YoshitakaMo/localcolabfold
AlphaFold Github page	https://github.com/google-deepmind/alphafold
AF_cluster Github page	https://github.com/HWaymentSteele/AF_Cluster
CCP4 Cloud page	https://cloud.ccp4.ac.uk/

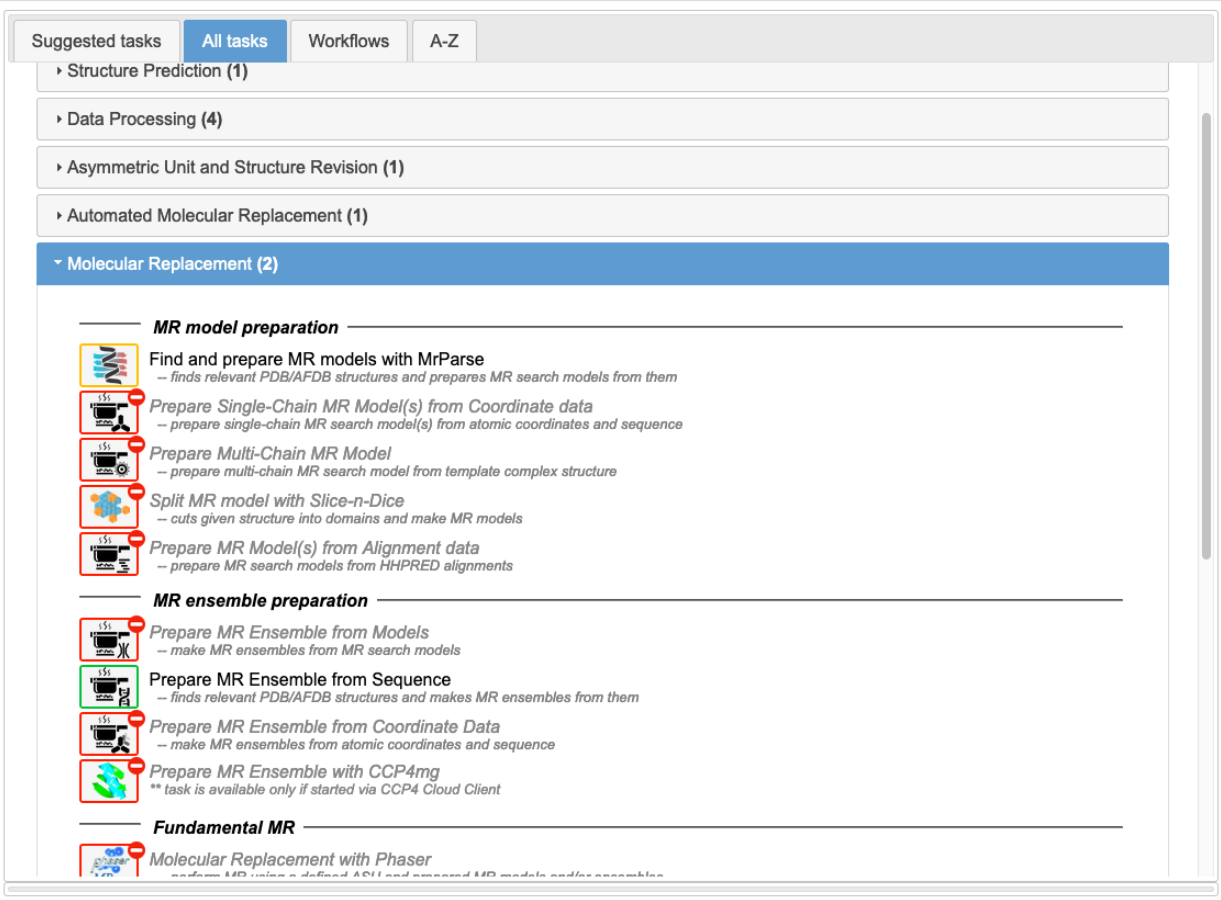
Models from the AlphaFold database

One option is to search for models directly, the AlphaFold database will allow you to search it using keywords (e.g. gene name), a UniProt ID, and more recently sequence.



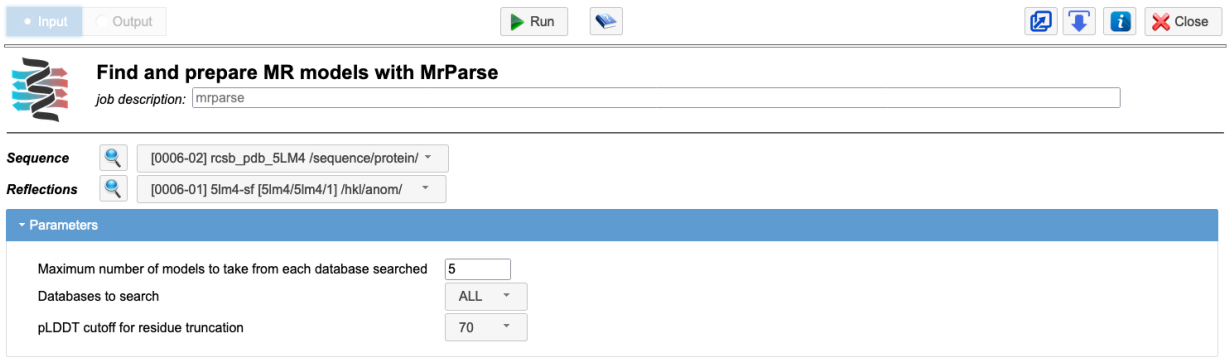
Another option in CCP4 is MrParse. This is a program designed to identify both homologs in the PDB and predicted models in the AlphaFold database.

MrParse can be found on CCP4 cloud under All tasks → Molecular Replacement → MR model preparation → Find and prepare MR models with MrParse



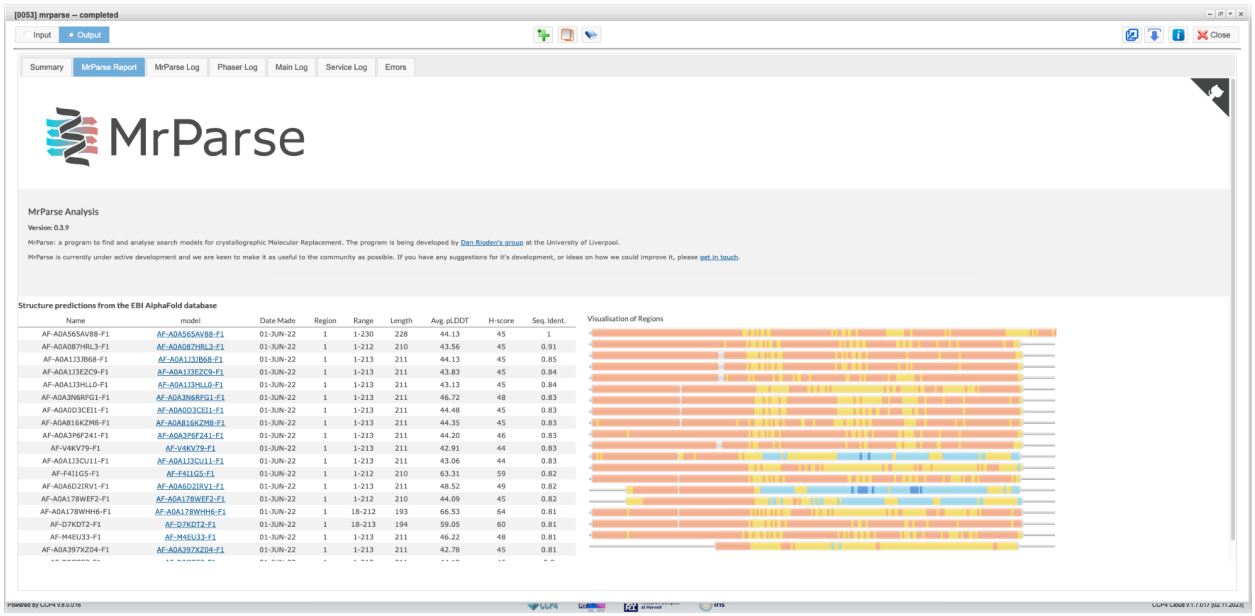
MrParse only requires a sequence, however if an MTZ is provided it will also provide information on crystal pathologies (e.g. twinning) and will calculate an eLLG for anything found in the PDB. eLLG is expected Log Likelihood Gain in Phaser given the quality of the data (e.g. resolution) and information about the PDB model (e.g. size and sequence identify). A high eLLG (>60) will indicate that a PDB entry will likely succeed in MR. If you have a good model in the PDB, this will often outperform an AF2 model (https://doi.org/10.1101/2022.11.21.517405)

Any predicted models found by MrParse will be prepared for Molecular replacement. This means that the poorly predicted regions (default: pLDDT <70) will be removed and the pLDDT scores in the B-factor column will be converted into a predicted B-factor.



You can select options including how many models to retrieve from each database, which databases to search and you can adjust the pLDDT threshold for truncation poorly predicted regions.

Running MrParse will output something like this:

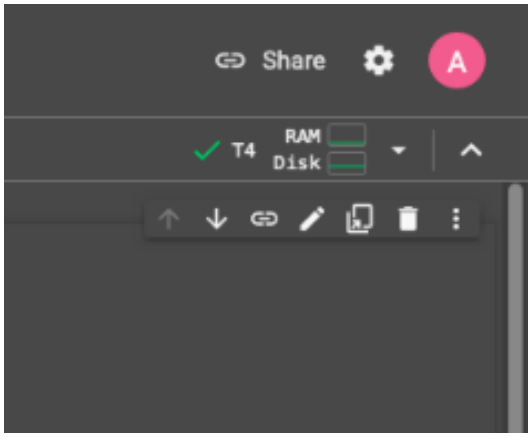


In this example I've only searched the AFDB and I've looked for 20 models. Here you can see some of the lower sequence identity hits make much better models than the 100% hits.

Generating models with colab pages

The easiest way to generate AlphaFold models is using the Colab pages. Depending on the GPU assigned to you by Colab, you can model up to 2000 residues (Tesla T4 & P100) or 1000 residues (Tesla K80). You can check the GPU assigned to you by looking in the top right hand corner. If you need to model something >1000 residues but only have a K80 assigned to you, keep refreshing the page until you get a T4/P100. Note: This tends to become relevant if you're

modelling complexes or multimers as this counts the total number of residues. E.g. a 300 residue protein in a hexamer would be 1800 residues.

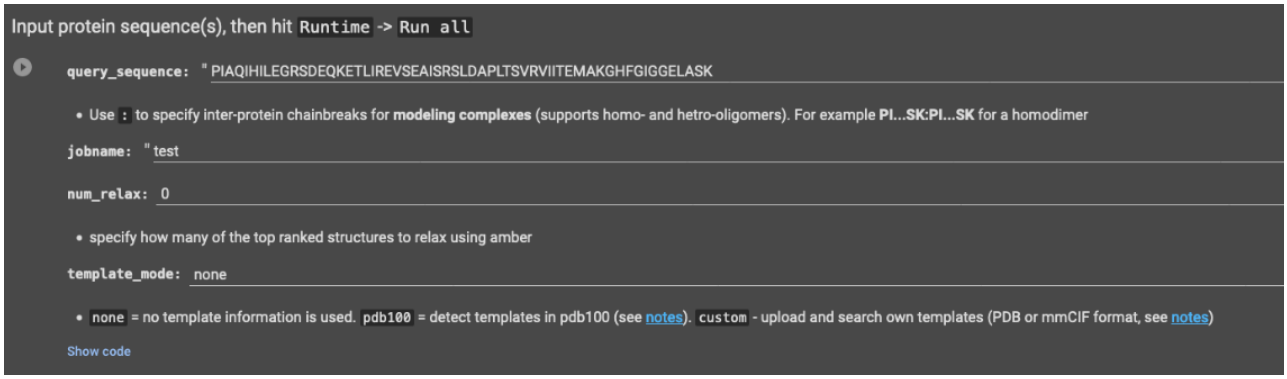


Both ColabFold and AlphaFold colab pages will output the results of their jobs as a zipped file once they have finished.

ColabFold

Generally I would recommend using ColabFold. The major change in ColabFold is that it uses MMSeqs2 to perform the MSA search instead of Jackhmmer. MMSeqs2 sacrifices some of the sensitivity of Jackhmmer but is an order of magnitude quicker.

ColabFold has a lot of user adjustable parameters. For the majority of cases, leaving the default parameters will be sufficient to make a good model.
To run ColabFold, we simply need to add our query sequence and navigate to Runtime → Run all



If your target is a complex/multimer, we need to adjust our target sequence to represent that. You need to separate each chain with a colon, e.g. for a homodimer:

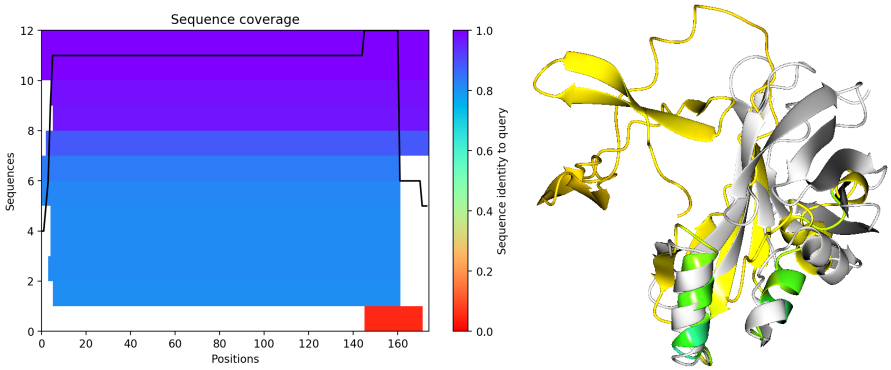
PIAQIHILEGRSDEQKETLIREVSEAISRSLDAPLTSVRVIITEMAKGHFGIGGELASK:PIAQIHILE
GRSDEQKETLIREVSEAISRSLDAPLTSVRVIITEMAKGHFGIGGELASK

For complexes/multimers, I would recommend modelling the individual components and the complexes and testing both in Molecular Replacement.

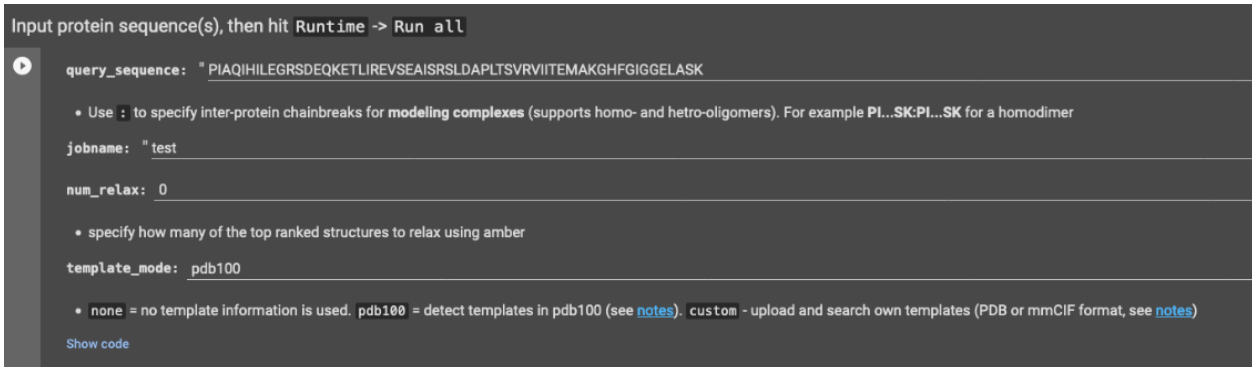
What to do if my model is bad?

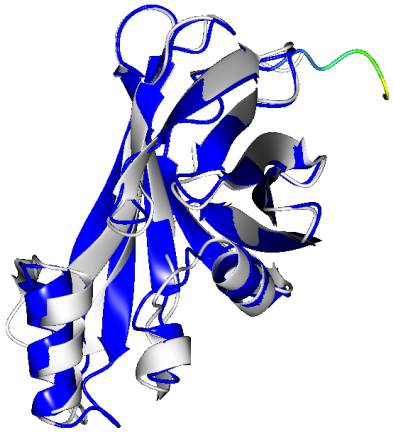
The first thing you need to do is try and work out *why* your model is bad. The most common reason will be due to a poor MSA.

Example. 7SNR: only 12 sequences found and a bad corresponding model



In this instance, turning on the template search in ColabFold makes a huge difference. In the first box, select the template_mode drop down and change it to pdb100:





If you have an unpublished structure of a close homologue, this could also improve the ColabFold modelling. You can change the template mode to custom and then upload your own template if this applies to you.

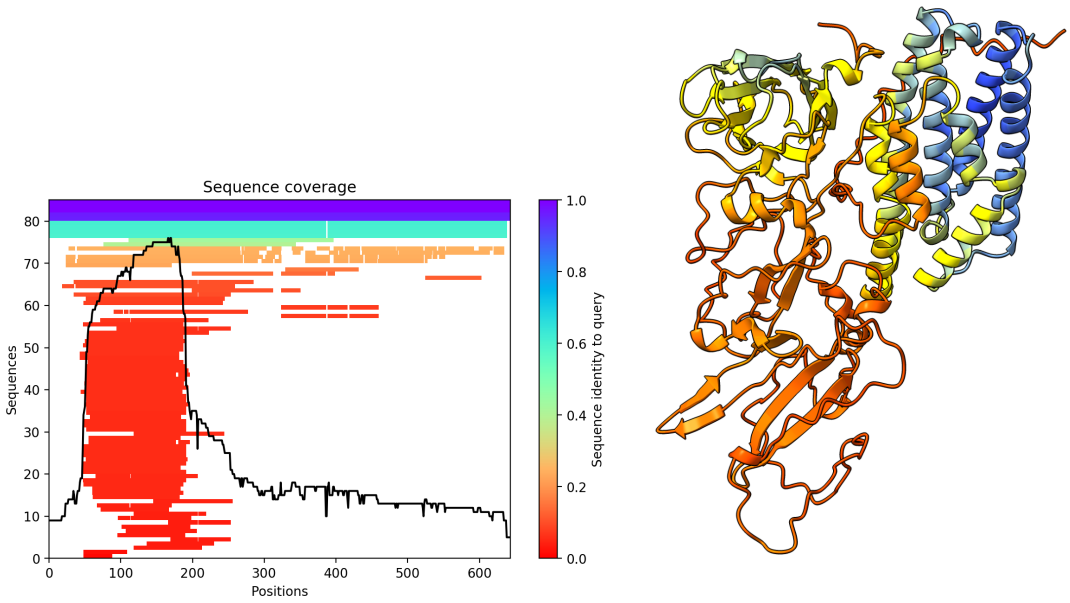
AlphaFold

AlphaFold's colab page has far fewer options than ColabFold and doesn't use templates. It is also significantly slower than ColabFold as it uses Jackhmmer to perform the MSA search. Nonetheless, this deeper MSA can prove useful in some cases.

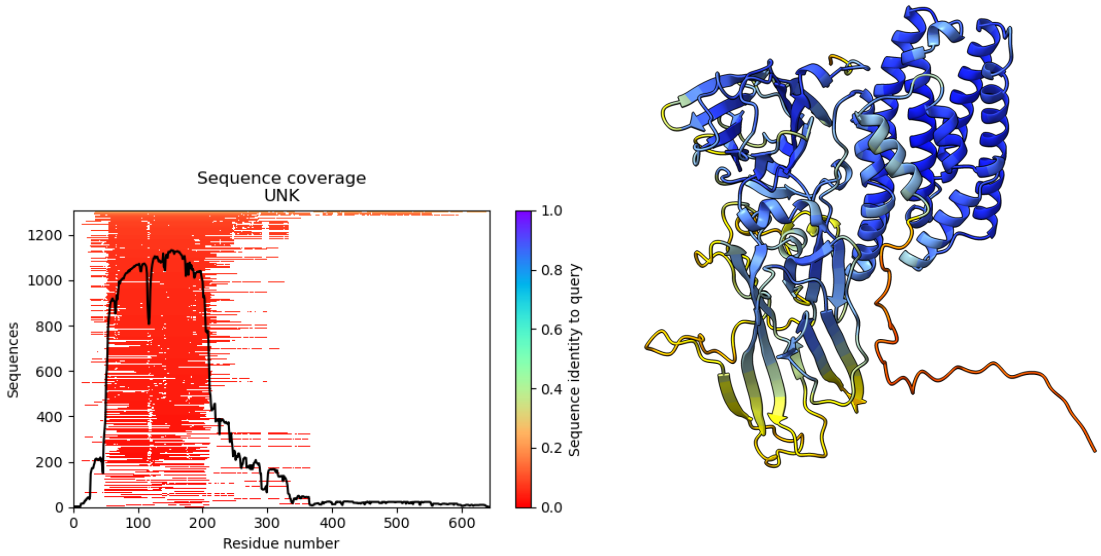
The main difference to ColabFold is that for multimers/complexes, instead of separating chains with colons, you enter each sequence on a separate line.

There is also an option to use the multimer modelling mode for single sequences, this tends to improve very large models.

Example: 7QX4 - the deeper MSA produced by Jackhmmer results in a better model



Approximately 80 sequences were found by ColabFold's MMSeqs2 and one domain was very poorly modelled.



Approximately 1200 sequences were found by AlphaFold's Jackhmmer and both domains were well modelled.

Generating models locally

There are some limitations to running jobs on colab pages, most notably, the size of the protein that you can predict. If you have access to a good computing system, e.g. a university computing cluster, installing AlphaFold and running it through that should allow you to generate much larger models. Installing it locally on a laptop will give you no benefit over the colab pages.

The key thing for running AlphaFold to generate large models quickly is that you have a relatively recent Nvidia GPU with a large amount of VRAM. If the modelling isn't time sensitive you can run it using your CPU, but this will take a very long time.

Finally, if you know anyone else with a good computer, you can always ask them to generate the models for you (People working on Cryo-EM are often a good bet!)

Installing ColabFold

Installing ColabFold is significantly quicker and requires a lot less space. This is largely because the MSA search is performed using the MMSeqs2 API and so you don't need to store the sequence databases locally.

Full instructions for installing ColabFold locally on Linux, MacOS and Windows are given here: https://github.com/YoshitakaMo/localcolabfold

Installing AlphaFold2

As mentioned above, AlphaFold2 is more effort to install because the MSA search is performed locally. On that note, because the MSA search is performed locally on a CPU, AlphaFold2 benefits from having a good CPU in addition to a good GPU.

Full instructions for installing AlphaFold locally on Linux are given here: https://github.com/google-deepmind/alphafold

Note: You will need 2.62Tb of storage for the databases.

Exploring conformations

Many proteins are flexible and therefore the model that you produce may be in a different conformation to your crystal structure. Subsequently, the model you produce may not succeed in molecular replacement.

AF_cluster

One option is to try and generate your model in multiple conformations. AF_cluster works by clustering the MSA generated by ColabFold and outputting multiple sub-sampled MSA files. The idea is that the MSA contains information on multiple conformational states, but the combined MSA will only output a single dominant conformation.
By sub-sampling the MSA, AF_cluster is able to separate out the conformational states, although the reduced MSA depth in each cluster may result in slightly worse models.

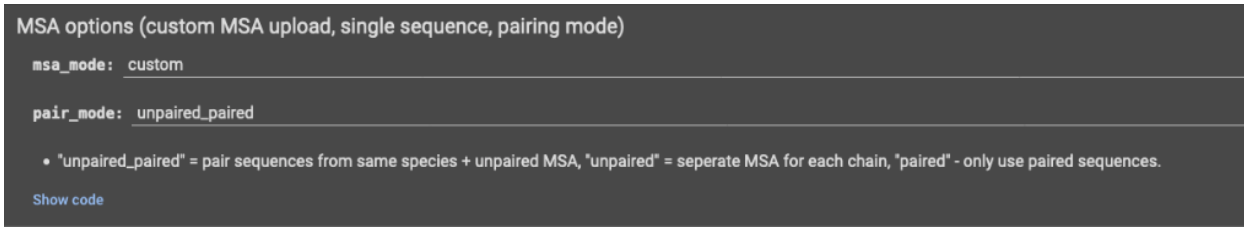
To run AF_cluster you need an MSA file. This is a .a3m file output by ColabFold.

The code can be retrieved from here: https://github.com/HWaymentSteele/AF_Cluster and can be run as follows:

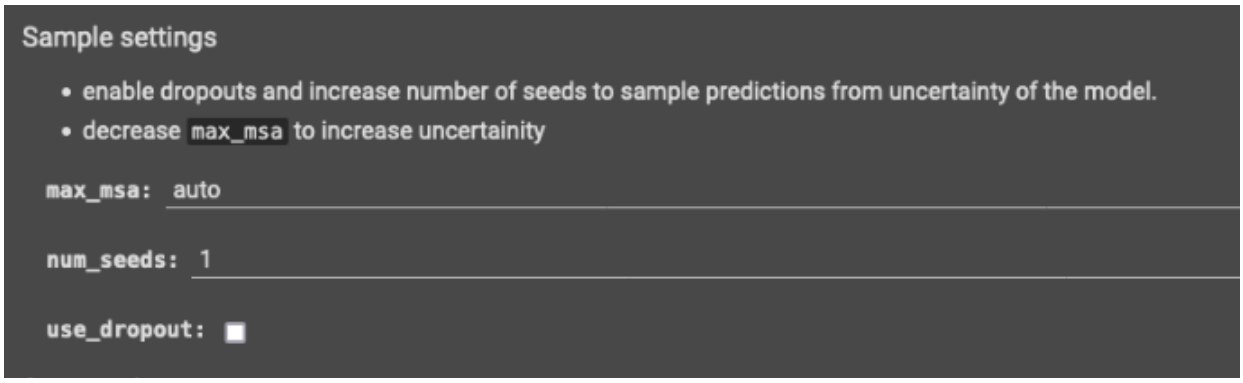
```
python scripts/ClusterMSA.py EX -i initial_msa.a3m -o msas
```

Where initial_msa.a3m is the file output by ColabFold and msas is a directory created with the sub-sampled MSAs.

We can then re-run ColabFold using these sub-sampled MSAs in turn by selecting the custom MSA option:



You may additionally benefit by altering the sample options in ColabFold:



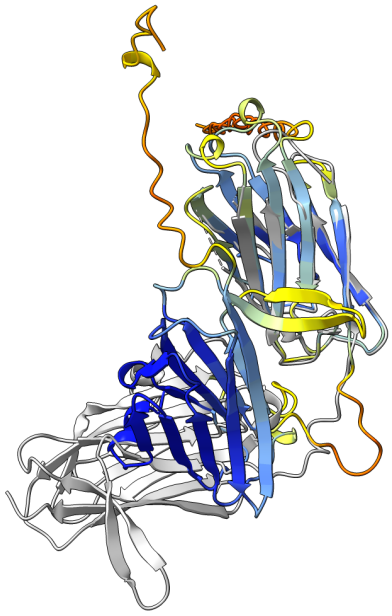
Increasing the number of seeds and enabling use_dropout increase the variability of the output ColabFold model.
We can ignore max_msa here as we're already modifying the MSAs using AF_cluster.

Slice’N’Dice

It can be tricky to get AlphaFold to correctly adopt the conformation of your target when you don’t know the conformation of your target.

Slice’N’Dice is a method developed to break a predicted model up into district structural units (or slices) and then place these individually. This can often result in successful MR when the conformation adopted by a protein is completely wrong.

E.g. 8GSX



Here our ColabFold model is aligned to the crystal structure. You can see that the two domains look well modelled but the conformation is well off.

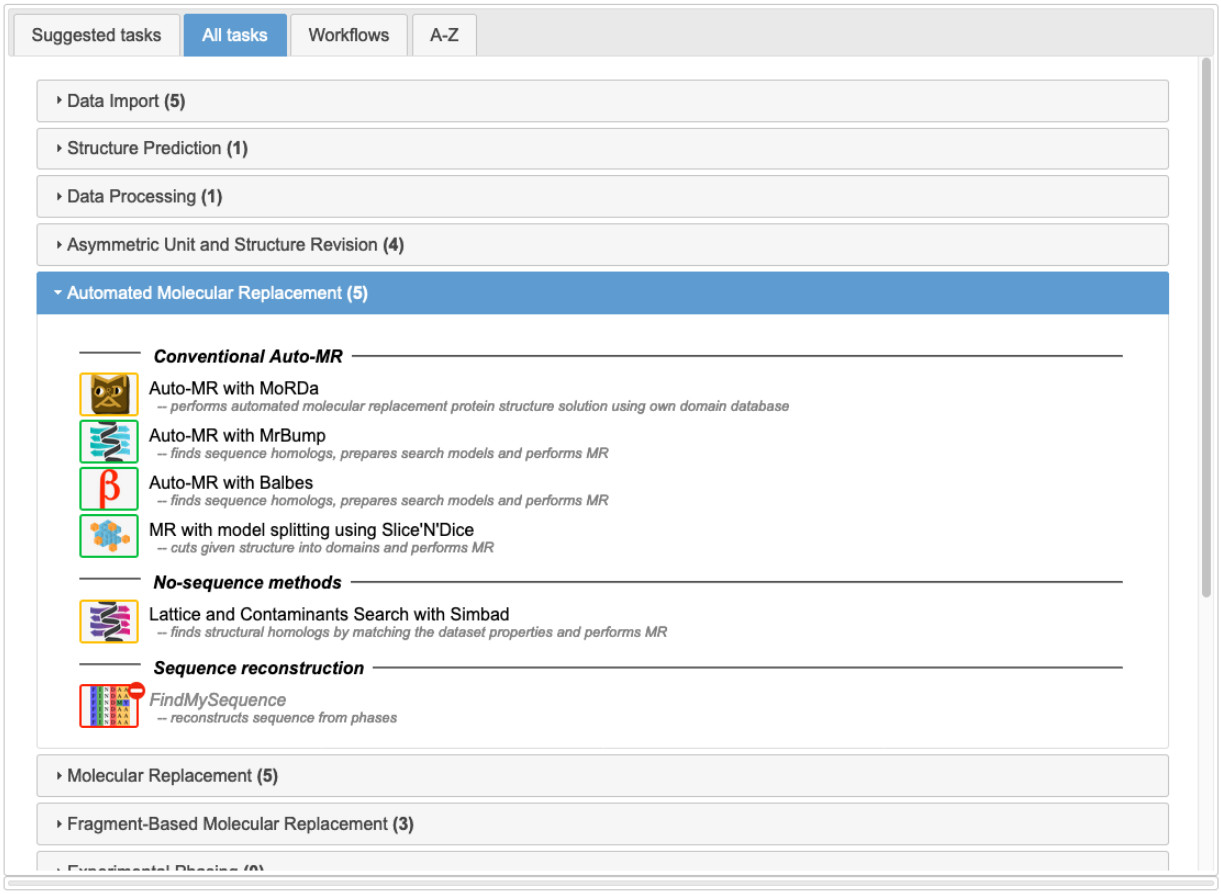
This one is deceptively tricky, because even though it is incorrect, Phaser still gives excellent scores (LLG: 305, TFZ: 22.5) because of the correctly placed domain. The R-scores are a little high (R-fact: 0.43, R-free: 0.44) but it looks like there is a potential solution here.

Slice’N’Dice breaks the model in 2 and automatically removes the low-pLDDT regions and performs a pLDDT → predicted B-factor conversion. You can see the individual domains fit the target a lot better:

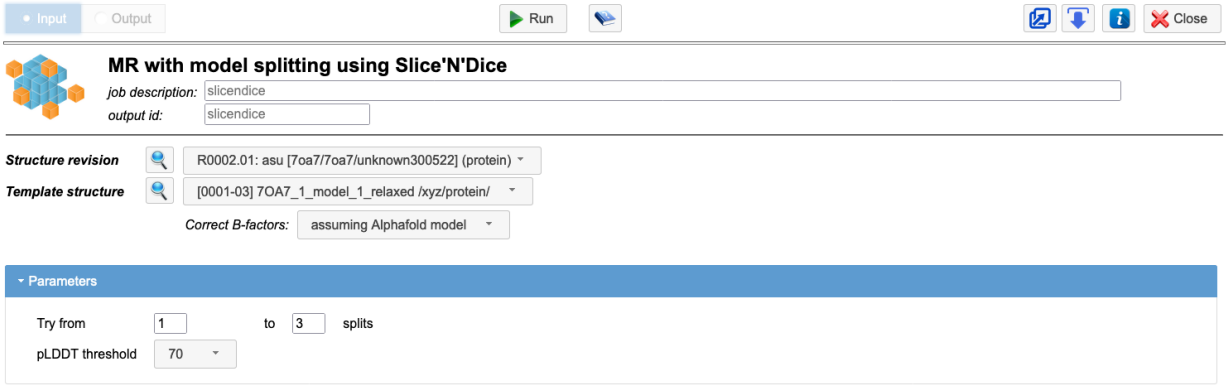
As expected, the scores all improve massively (LLG: 1407, TFZ: 28.7, R-fact 0.34, R-free: 0.38) and an ambiguous solution becomes an obvious one.

Whenever you see a flexible linker as in this case, it’s well worth using Slice’N’Dice.

To run Slice’N’Dice, navigate to Automated Molecular Replacement → Converional Auto-MR → MR with model splitting using Slice’N’Dice



Here you can select your data/model from you cloud project and select some options.



If the model is straight from AlphaFold, select correct B-factors assuming AlphaFold model. If the model has already had B-factor correction (e.g. obtained from MrParse), change this to “do not correct”.
Slice’N’Dice will try a range of splits for your model, by default ranging from 1 (no splitting) to 3. You can play about with these numbers as and if you feel it’s required.
Finally you can change the pLDDT threshold value. This can be useful if you think your model looks decent but you don’t have many sequences in the MSA (and therefore low pLDDT scores).