Class 11*

Protein Structure Prediction with AlphaFold

Barry Grant

2023-11-06

1. Overview

In this hands-on session we will utilize **AlphaFold** to predict protein structure from sequence (Jumper et al. 2021).

Without the aid of such approaches, it can take years of expensive laboratory work to determine the structure of just one protein. With AlphaFold we can now accurately compute a typical protein structure in as little as ten minutes.

This major breakthrough (Figure 1) promises to place Molecular Biology in a new era where we can visualize, analyze and interpret the structures and functions of all proteins.

2. Background:

Proteins are natures robots being responsible for nearly every task in living organisms. Just as robots are programmed to perform specific tasks, proteins, through their distinct 3D shapes (known as their **atomic structures**), carry out diverse functions from catalysis, signaling and transport to replication, division and movement. Indeed, even a subtle alteration in structure can lead to drastic changes in activity and function. Knowledge of protein structures therefore yields a deeper understanding of protein function and disfunction in health and disease.

Unfortunately, experimental structure determination is far more complex and time-consuming than sequence determination. The main PDB repository of protein structures lags far behind sequence databases such as UniProt in terms of size and coverage of known proteins (just 7% coverage as we saw in our last lab). Expanding this coverage would vastly accelerate efforts to understand distant evolutionary relationships, probe the inner-workings of cells and enable more rapid drug discovery.

*http://thegrantlab.org/teaching/

Figure 1: AlphaFold was awarded “breakthrough of the year” award for 2021

3. Structure Prediction

It has been known from the work of Anfinsen in the 1960s that the primary sequence of a protein contains all the information necessary for correct folding and structure formation. This seminal work lead to the foundational **sequence-structure-function** tenet of molecular biology (Figure 2). That is, protein sequence dictates 3D structure, which in turn governs interactions and roles within the cellular milieu. However, the ability to accurately predict structure from sequence has remained elusive until very recently.

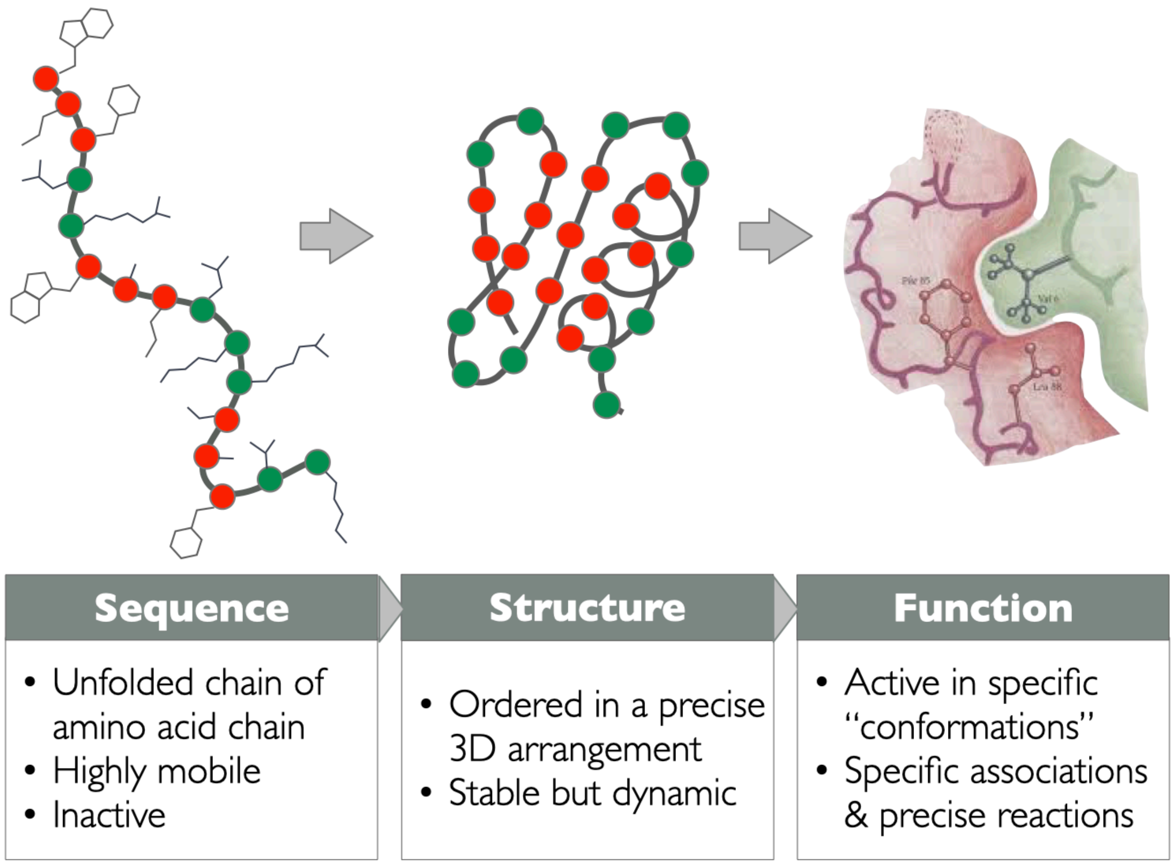

Sequence
• Unfolded chain of amino acid chain
• Highly mobile
• Inactive
Structure
• Ordered in a precise 3D arrangement
• Stable but dynamic
Function
• Active in specific "conformations"
• Specific associations & precise reactions

Figure 2: The sequence-structure-function relationship is akin to the holy trinity of molecular biology

The importance of open community benchmarks

Key to major progress was the establishment of the **Critical Assessment of Structure Prediction** (**CASP**) community benchmark challenge. Established in the early 1990s, CASP is a biennial competition that provides a platform for researchers worldwide to rigorously test and compare their computational methods against yet-to-be-revealed experimental protein structures. By offering a standardized benchmark, CASP has fostered collaboration, transparency, and innovation among scientists, pushing the boundaries of predictive accuracy. The iterative nature of the competition has charted the progress and challenges in the field, highlighting both advancements and areas in need of further research. Crucially, CASP's community engagement and unbiased assessments have been instrumental in validating the true capabilities of various prediction tools, with one method in particular standing out above all others in the most recent CASP14 competition (Figure 3).

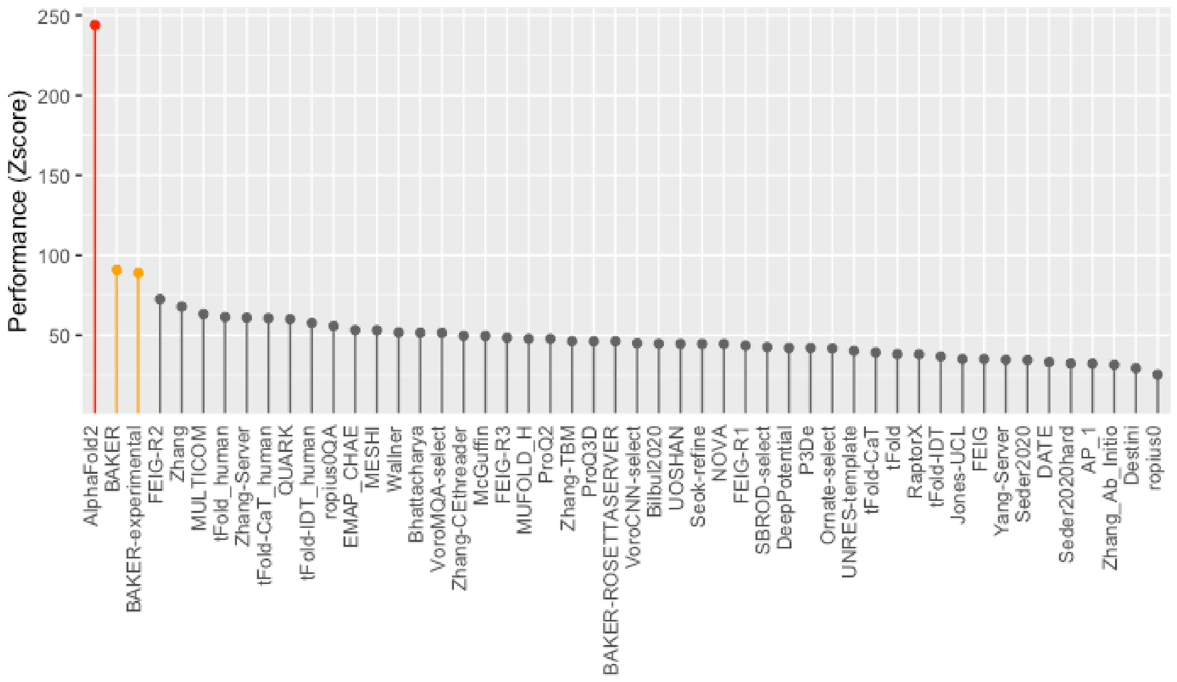
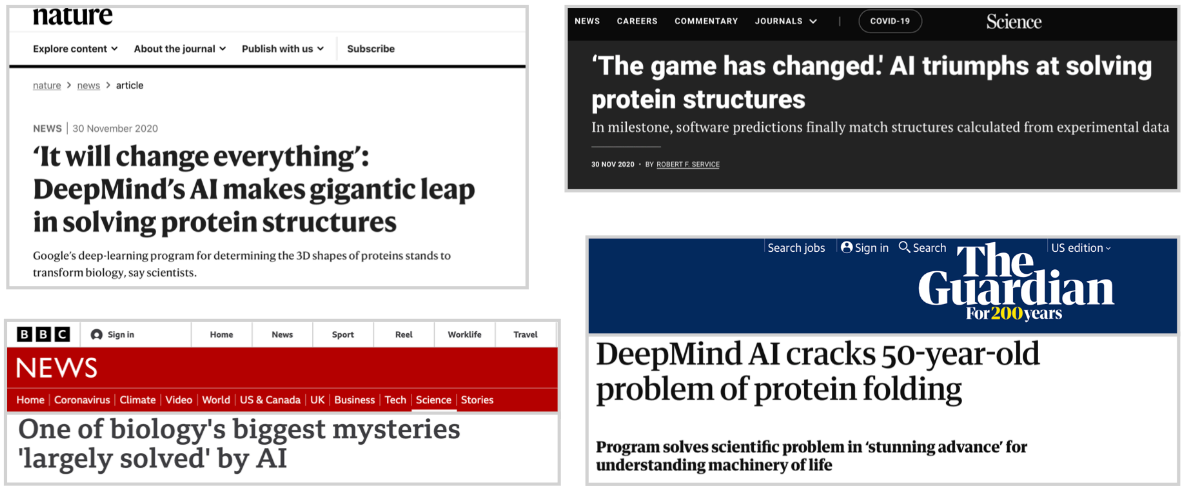


Figure 3: CASP14 in 2020 saw an enormous jump in prediction accuracy (red and orange). This advance is largely the result of the successful application of deep learning methods, particularly by AlphaFold2 and, since that CASP, RosettaFold by the Baker group. Note that only the top 50 ranked groups are shown. For all scores see https://predictioncenter.org/casp14/zscores_final.cgi.

4. AlphaFold and the AI revolution

Predicting protein structure from sequence alone remains an unsolved "grand challenge" problem. However, by utilizing the information contained in very large multiple sequence alignments (MSAs) as input features for AI approaches, AlphaFold2 was able to predict structures with unparalleled accuracy in recent CASP editions. Developed by the Google owned DeepMind company, AlphaFold2 (the second version of AlphaFold) employs advanced deep learning techniques, together with physics based refinement, to predict structures with an accuracy that is often within the error margin of experimental methods (Figure 4).



NEWS | 30 November 2020

‘It will change everything’: DeepMind’s AI makes gigantic leap in solving protein structures

Google’s deep-learning program for determining the 3D shapes of proteins stands to transform biology, say scientists.

‘The game has changed.’ AI triumphs at solving protein structures

In milestone, software predictions finally match structures calculated from experimental data

30 NOV 2020 • BY ROBERT F. SERVICE

DeepMind AI cracks 50-year-old problem of protein folding

Program solves scientific problem in ‘stunning advance’ for understanding machinery of life

One of biology's biggest mysteries 'largely solved' by AI

Figure 4: AlphaFold makes headlines in the wake of it’s standout performance in recent CASP editions. These headlines underscore AlphaFolds transformative impact on the discipline.

AlphaFold2 system architecture

At its core AlphaFold employs a deep residual neural network that predicts inter-residue distances and angles from co-evolutionary patterns in MSAs of related proteins (Figure 5). These predictions are then transformed into a spatial representation of the protein, similar to a distance matrix. The system then refines this representation, leveraging both the predicted geometric constraints and physical principles derived from known structures, to arrive at the most probable 3D structure. Another key component is the so-called “recycling” procedure, where the network is iterated multiple times to further refine predictions. A full description of the system architecture and details of the neural network training process are given by Jumper *et al* (Jumper et al. 2021).

5. The EBI AlphaFold database

AlphaFold structure predictions for over 200 million proteins are now available in the EBI AlphaFold Database (AFDB) (Figure 6). This is a result of an ongoing a collaboration between DeepMind and the European Bioinformatics Institute (EBI). The online interface allows for

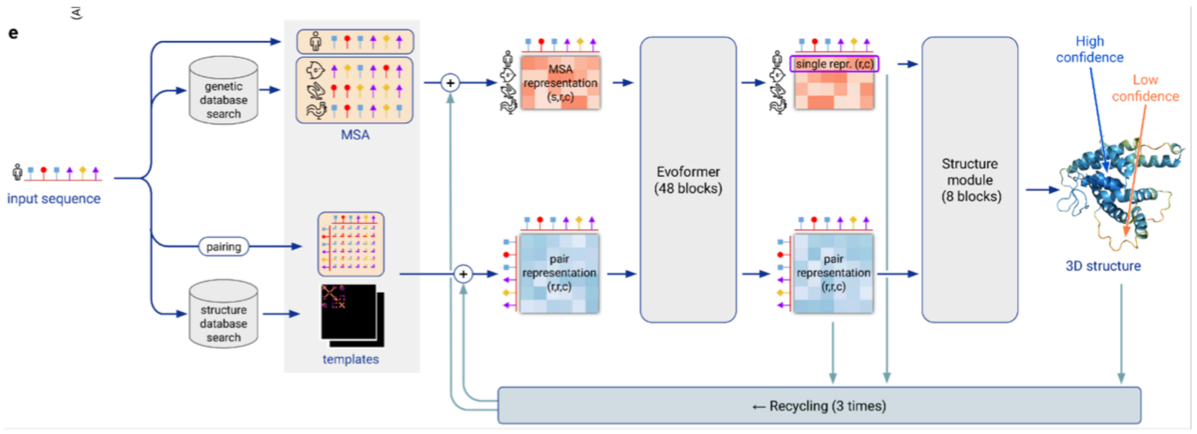


Figure 5: AlphaFold2 system architecture as described in (Jumper et al 2021). Multiple sequence alignments (MSAs containing large numbers of related sequences) together with structure templates (essentially related structures identified from the PDB) from the major inputs to the first of two deep neural networks (called Evoformer). This in essence identifies interacting pairs of amino acids that can be further processed by the second "structure module" to initial 3D atomic coordinates. This output is then further refined with molecular mechanics (Amber99sb force field minimization). Finally, a "recycling" procedure incorporates the output as additional inputs to Evoformer for another three iterations.

searching based on protein names, sequences, or accession numbers. Additionally, the database offers basic visualization of structure models with Mol* and downloadable PDB files (Figure 7). If your protein of interest is there then you don't need to proceed with the timely step of running AlphaFold yourself (next section).

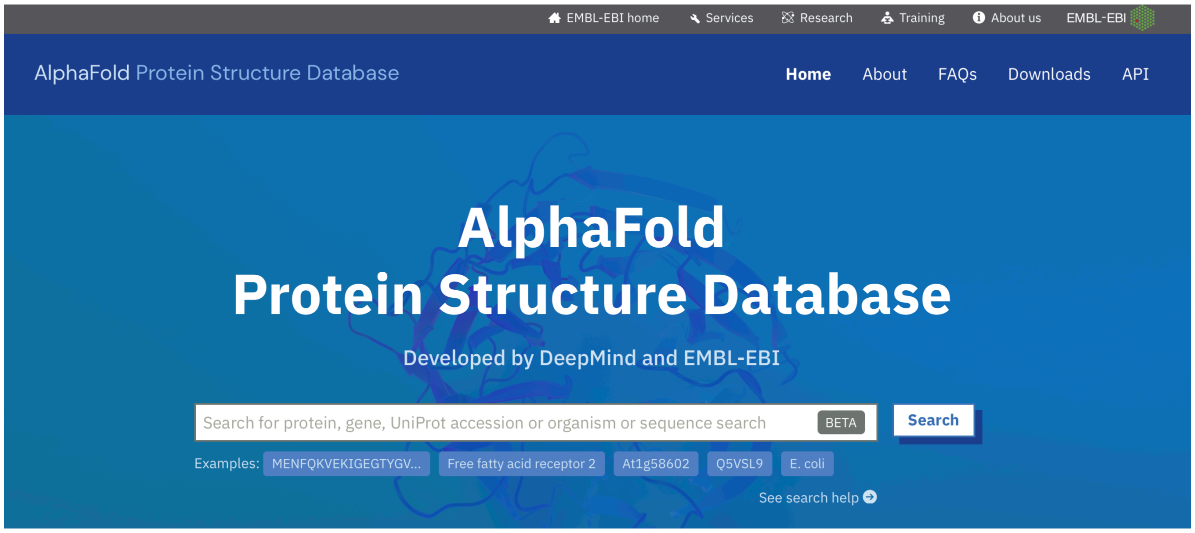

EMBL-EBI home
Services
Research
Training
About us
EMBL-EBI
AlphaFold Protein Structure Database
Home
About
FAQs
Downloads
API
AlphaFold
Protein Structure Database
Developed by DeepMind and EMBL-EBI
Search for protein, gene, UniProt accession or organism or sequence search
BETA
Search
Examples:
MENFQKVEKIGEGTYGV...
Free fatty acid receptor 2
At1g58602
Q5VSL9
E. coli
See search help

Figure 6: The AlphaFold database now contains over 200 million protein structure models.

Querying the AlphaFold database

Lets look in the AlphaFold Database (AFDB) for a structure model of HIV-Protease similar to the one we viewed extensively in the last lab session.

- 1. Use the following sequence to search AFDB:

```
>HIV-Pr
PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQ
```

- 2. Examine one of your top hits. Note that at the time of writing HIV virus structures are not included in the current AFDB. However, there are a number of very closely related sequences from other species with informative models that we can learn from.

For me the top hit was named "Peptidase A2 domain-containing protein" from *Thalassobius mangrovi* with id A0A6L8LSL6.

- 3. Visit the corresponding AFDB structure page. It shows basic information about the protein (drawn from UniProt), and three separate outputs from AlphaFold.

The first is the 3D coordinates displayed with Mol* and colored by a per-residue confidence metric called **pLDDT**.

pLDDT stands for "*predicted local distance difference test*" and is a confidence score that is computed for each amino acid residue. This score is a measure of how well the method has converged (*i.e.*, how well the predicted structure agrees with MSA and PDB structure information).

> pLDDT values range between 0 and 100. Values above 70 are considered high confidence whereas those scored below 50 are low confidence and hence unreliable.

Note that pLDDT scores can vary greatly along a chain so it is important to consult these confidence estimates when interpreting structural features. The lower confidence bands may be associated with disorder particularly apparent at the C-terminus of this model, which looks very unreliable in this case.

The third major output is a matrix plot of **Predicted Aligned Error (PAE)** values (Figure 7). PAE is an important metric for assessing domain packing confidence and larger-scale topology reliability.

On the AFDB page you can click and drag on this PAE matrix plot to highlight corresponding regions of structure. Note again that the C-terminus has much lower PAE scores and is thus unreadable in its packing with the high confidence N-terminal portion of the structure.

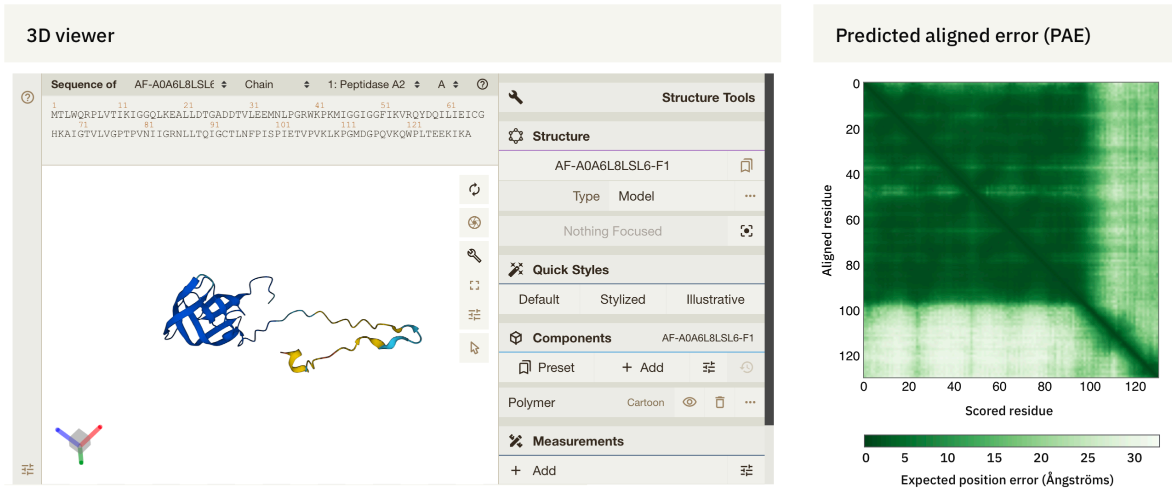
We will discuss the interpenetration of PAE values and this plot in person (ask Barry now if I have not already done this). We will also analyze PAE values further below in the context of our own models in Section 8.

6. Generating your own structure predictions

If you can't find an existing structure of interest in the PDB or AFDB then you can generate your own structure predictions with ColabFold (Mirdita et al. 2022).

> **Side-note**: You can also obtain the AlphaFold source code to install on your own hardware. However, installation is not straightforward as it requires GPU or TPU access and large sequence databases to be available. Running AlphaFold via ColabFold (that uses Googles cloud infrastructure) therefore makes sense for most applications.

There are a number of different AlphaFold notebooks, for running slightly different versions of the software. This includes one made available by DeepMind itself. This original version uses the excellent but rather slow HMMer and HHblits hidden Markov models based sequence search and alignment methods we covered back in class 4.



```
1         11        21        31        41        51        61
MTLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMNLPGRWKPKMIGGIGGFIKVRQYDQILIEICG
    71        81        91        101       111       121
HKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFPISPIETVPVKLKPGMDGPQVKQWPLTEEKIKA
```

Figure 7: AFDB entry for the Peptidase A2 domain-containing protein from *Thalassobius mangrovi*. The structure is displayed in Mol* colored by pLDDT scores. The PAE plot is also interactive and can be clicked on to highlight coresponding regions in the structure in Mol*

For this lab we will use the ~40 fold faster MMSeqs2 version. This version, like all the these "notebooks", is composed of 'cells' that typically run the code required for the next cell. You can run each individual cell with a Run button next to that cell. Alternatively, you can run all the cells in order with a pull-down menu item called **"Run all"** under the heading **"Runtime"** at the top of the page. Before doing this be sure to put in your input sequence and job name before clicking **"Run all"** option.

Getting Started with ColabFold

- 1. First, check the AFDB for the protein of interest. If your structure has already been predicted there, you can download the AFDB PDB file and skip to the **Interpreting Results** section below.

- 2. Otherwise obtain the sequence of your protein of interest, e.g. at UniProt. Click on the FASTA button above the sequence in UniProt. Copy only the sequence, excluding the FASTA header line that begins with ">".

- 3. For your first time through this lab I would like you to use the HIV-Pr sequence to generate a single chain model. After your first run you can experiment with generating the biologically relevant homodimer (the monomer is unstable in reality and the dimer is the functional unit):

```
>HIV-Pr
PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQ
```

And for your **2nd run** the dimer input. As it is a homodimer this consists of the same sequence twice with a colon between chains:

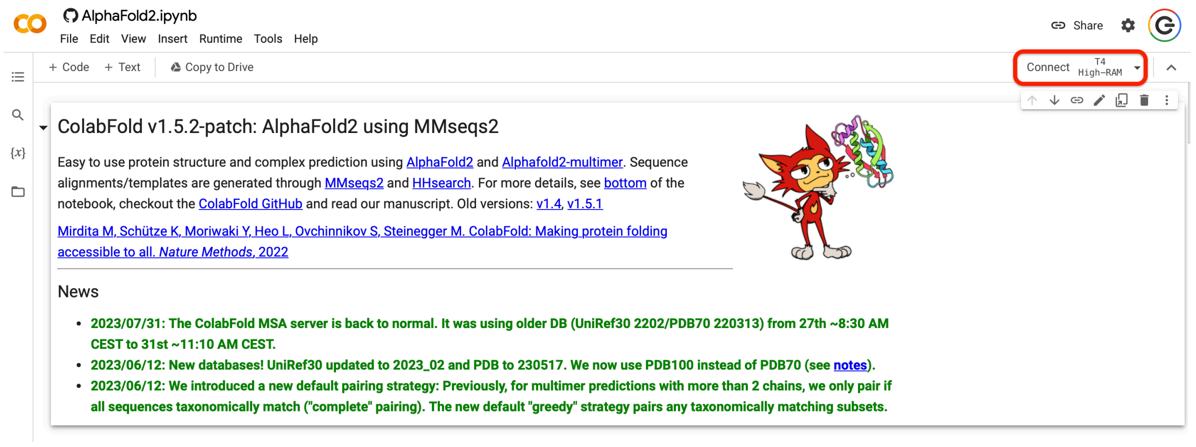
```
>HIV-Pr-Dimer
PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYD
QILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF:PQITLWQRPLVTIKIGGQLK
EALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPT
PVNIIGRNLLTQIGCTLNF
```

You can also experiment with your "find-a-gene" project sequence as you will need this result for your project work).

- 4. Visit AlphaFold2_mmseqs2 Colab notebook (at the time of writing this is currently the preferred AlphaFold version for our current prediction tasks.

You can always check for updates and alternate versions here)

4. Click on **"Connect"** on top right toolbar to obtain access to computing resources to run AlphaFold on. If successful this will connect you to Google Compute Engine cloud resources with a GPU. A green tick should appear along with RAM and Disk space graphics (Figure 8).



ColabFold v1.5.2-patch: AlphaFold2 using MMseqs2

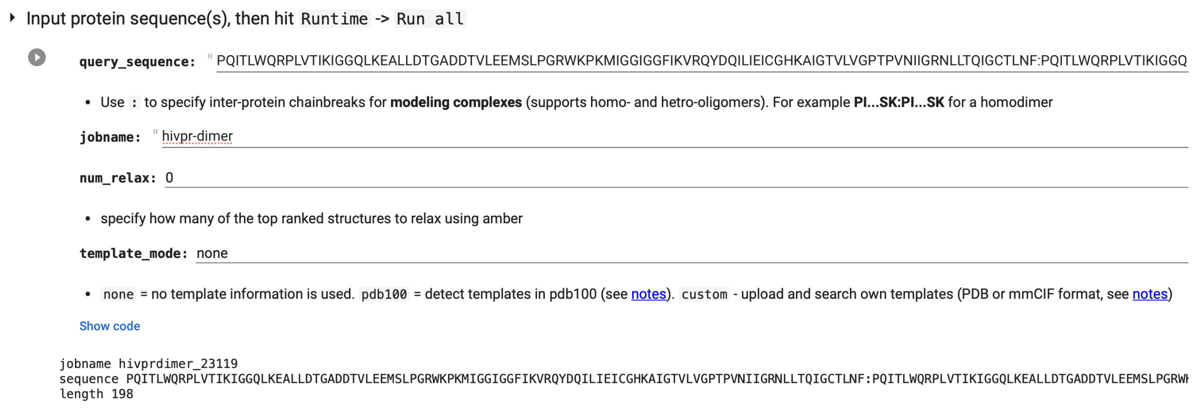
Easy to use protein structure and complex prediction using AlphaFold2 and Alphafold2-multimer. Sequence alignments/templates are generated through MMseqs2 and HHsearch. For more details, see bottom of the notebook, checkout the ColabFold GitHub and read our manuscript. Old versions: v1.4, v1.5.1

Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: Making protein folding accessible to all. *Nature Methods*, 2022

News

- 2023/07/31: The ColabFold MSA server is back to normal. It was using older DB (UniRef30 2202/PDB70 220313) from 27th ~8:30 AM CEST to 31st ~11:10 AM CEST.
- 2023/06/12: New databases! UniRef30 updated to 2023_02 and PDB to 230517. We now use PDB100 instead of PDB70 (see notes).
- 2023/06/12: We introduced a new default pairing strategy: Previously, for multimer predictions with more than 2 chains, we only pair if all sequences taxonomically match ("complete" pairing). The new default "greedy" strategy pairs any taxonomically matching subsets.

Figure 8: The ColabFold interface for running AlphaFold on Google cloud computing resources. Note the red rectangle in the upper right highlighting the **"Connect"** button that must be activated before any computation can take place.



Input protein sequence(s), then hit `Runtime` -> `Run all`

query_sequence: PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF:PQITLWQRPLVTIKIGGQ

- Use `:` to specify inter-protein chainbreaks for **modeling complexes** (supports homo- and hetro-oligomers). For example **PI...SK:PI...SK** for a homodimer

jobname: hivpr-dimer

num_relax: 0

- specify how many of the top ranked structures to relax using amber

template_mode: none

- `none` = no template information is used. `pdb100` = detect templates in pdb100 (see notes). `custom` - upload and search own templates (PDB or mmCIF format, see notes)

```
jobname hivprdimer_23119
sequence PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF:PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRW
length 198
```

Figure 9: The first code cell of the ColabFold interface allows you to paste any protein **query_sequence** and enter a **jobname**.

- 6. In the first main page code cell paste in your query_sequence, making sure to completely replace the default sequence (Figure 9):

This input cell can accept sequences >1,000 amino acids, even though it is only one line. Sequence lengths of ~1,000 amino acids, or longer, may cause it to run out of compute resources and fail.

- 7. Enter a descriptive **jobname** (replacing the default "test" value). Note that the `results.zip` filename obtained at the end of the full computation will begin with this jobname (but none of its contents include the jobname).

- 8. For now leave all other parameters (i.e. the code cells) at their default values. We can explore them later after completing our first successful run.

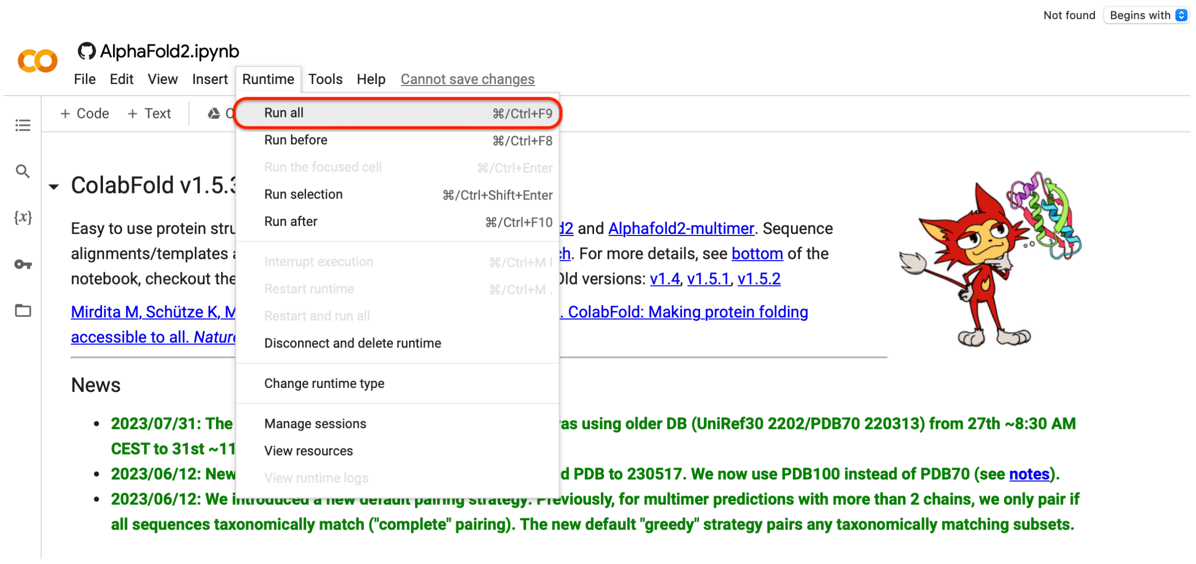
8. Back at the very top of the page where we have the "File"/"Edit"/"View" toolbar menu items click **"Runtime"** > **"Run All"** (Figure 10).

Don't worry about the "Warning". It is just Google's disclaimer that they did not write the code you are about to execute. Click Run anyway.

The currently running step is indicated by a circle with a stop sign next to it. Once complete a green tick indicates step success.

What is ColabFold doing? Note that the overall pipeline consists of multiple steps/code cells.

There is the first sequence input cell. By default this cell has `template_mode` set to **None**. This does not use any protein structure templates as input to AlphaFold. Hence only the MSA information (generated blow) will be used for this prediction. In a future run you can change this to PDB to include structure templates (usually always good idea).



ColabFold v1.5.3

Easy to use protein stru… …d2 and Alphafold2-multimer. Sequence alignments/templates … …h. For more details, see bottom of the notebook, checkout the … …Old versions: v1.4, v1.5.1, v1.5.2

Mirdita M, Schütze K, M… … ColabFold: Making protein folding accessible to all. Natur…

News

- 2023/07/31: The … …as using older DB (UniRef30 2202/PDB70 220313) from 27th ~8:30 AM CEST to 31st ~11…
- 2023/06/12: New … …d PDB to 230517. We now use PDB100 instead of PDB70 (see notes).
- 2023/06/12: We introduced a new default pairing strategy. Previously, for multimer predictions with more than 2 chains, we only pair if all sequences taxonomically match ("complete" pairing). The new default "greedy" strategy pairs any taxonomically matching subsets.

Figure 10: After you have successfully connected to compute resources and entered your sequence and identifying jobname you can select Runtime > Run all to execute all stages of the AlphaFold prediction pipeline.

There is also a `num_relax` option, which by default is set to zero. Setting this to one or more will have the effect of running an Amber based energy minimization on the models output from AlphaFold. Again, this is usually always a good idea as it corrects potential stereochemical and atom clash errors but it does increase compute time substantially.

The next cell installs software dependencies (i.e. downloads AlphaFold, conda, amber and hhsuite etc.). Recall that we are running on a fresh google compute cloud instance that does not have these required tools installed by default.

The **MSA options** cell allows you to upload your own MSA for use and optionally pair sequences from the same species across chains if you are running a multi-chain complex prediction.

The **Advanced settings** cell allows you to chose a different version of AlphaFold (there are 3 at the time of writing), run more recycles (usually a good idea), increase the amount of Amber energy minimization, and optionally save intermediate results.

The remaining cells either run the predictions and display the calculation status or display the results. The later includes a **3D structure** display colored by so-called lDDT (Figure 11). This is a confidence score called "predicted local distance difference test" (pLDDT) that is computed for each amino acid residue to estimate how well the method has converged (i.e., how well the predicted structure agrees with multiple sequence alignment data and PDB structure information).

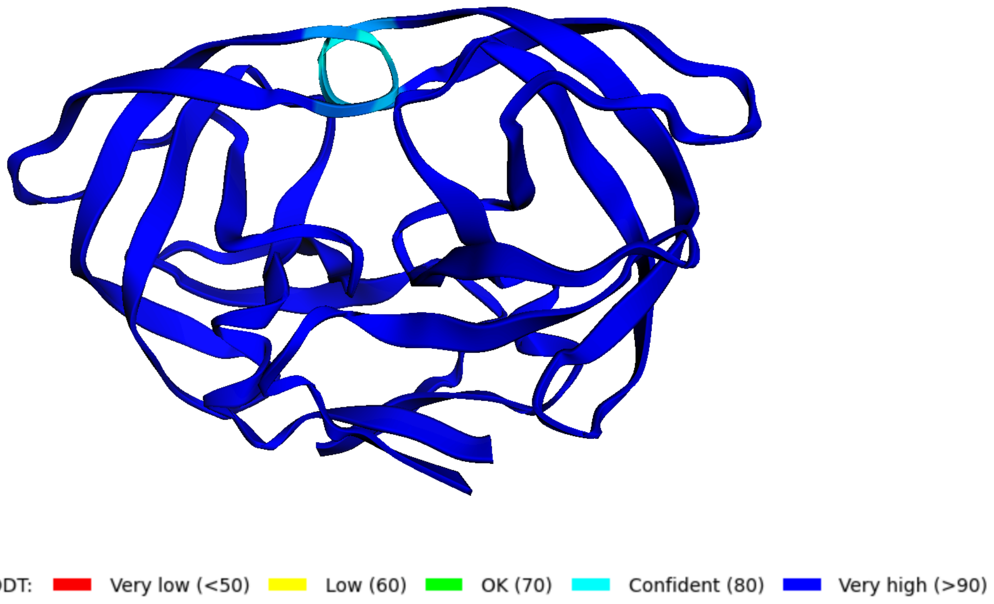


Figure 11: The top AlphaFold structure model colored by predicted IDDT values.

Also shown are diagnostic plots for MSA sequence coverage (i.e. how many sequences span the amino acids along the query sequence and IDDT per position (Figure 12).

- 10. Downloading Results **N.B. Do NOT close your browser tab until the job is completed**. You will lose your work if you close the browser tab. In some browser you will be warned if you inadvertently try but others, like Chrome, will not warn you.

When the job is finally completed (typically in under 10mins), a dialog to download a **zip file** will appear automatically. Sometimes you will be asked for permission to enable download first.

7. Interpreting Results

Once your zip file has been downloaded you can uncompress it to begin result interpretation. On a Mac or PC simply click on the file to expand it. At the command line you can use the `unzip` command.

Inside the resulting folder/directory you will have a number of .txt, .json, and .pdb files (Figure 13). The later are your new structures with `rank_001` being the top ranked model when they are sorted by avg. pLDDT score. These contain the coordinates in regular PDB

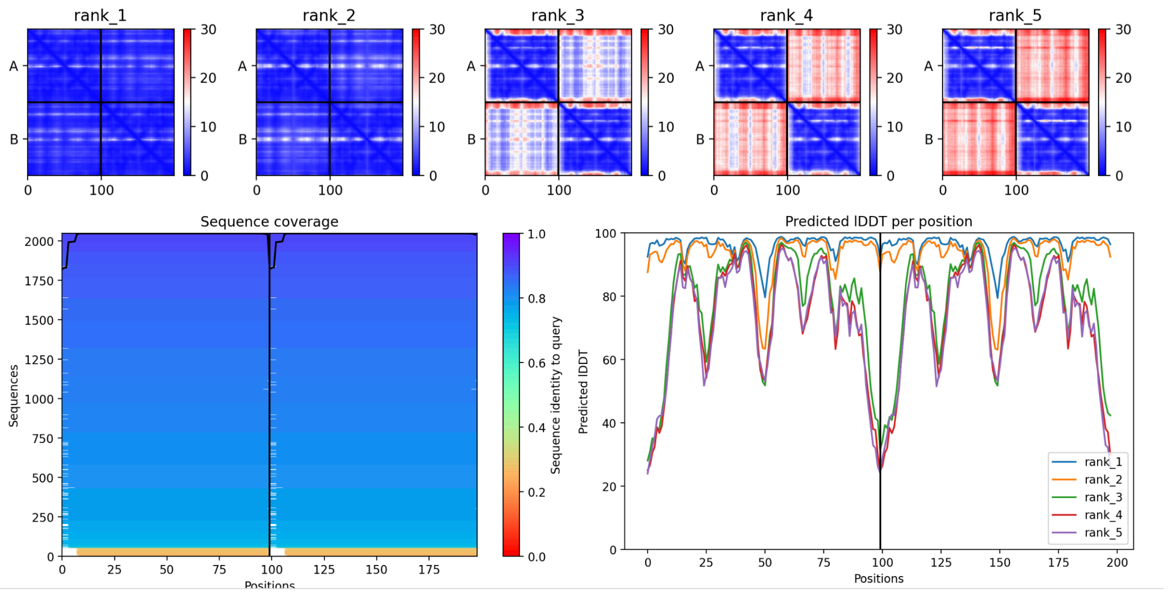


Figure 12: Sequence coverage in the MSA and the predicted local distance difference test (pIDDT) score for the top 5 ranked HIV-Pr dimer models from AlphaFold2. Note that if you are predicting for the monomer first your plots will look diferent obviously.

format along with pLDDT scores in the B-factor column. Also returned is the large MSA in a3m format and a CSV both of which we will work with later on (Figure 13).

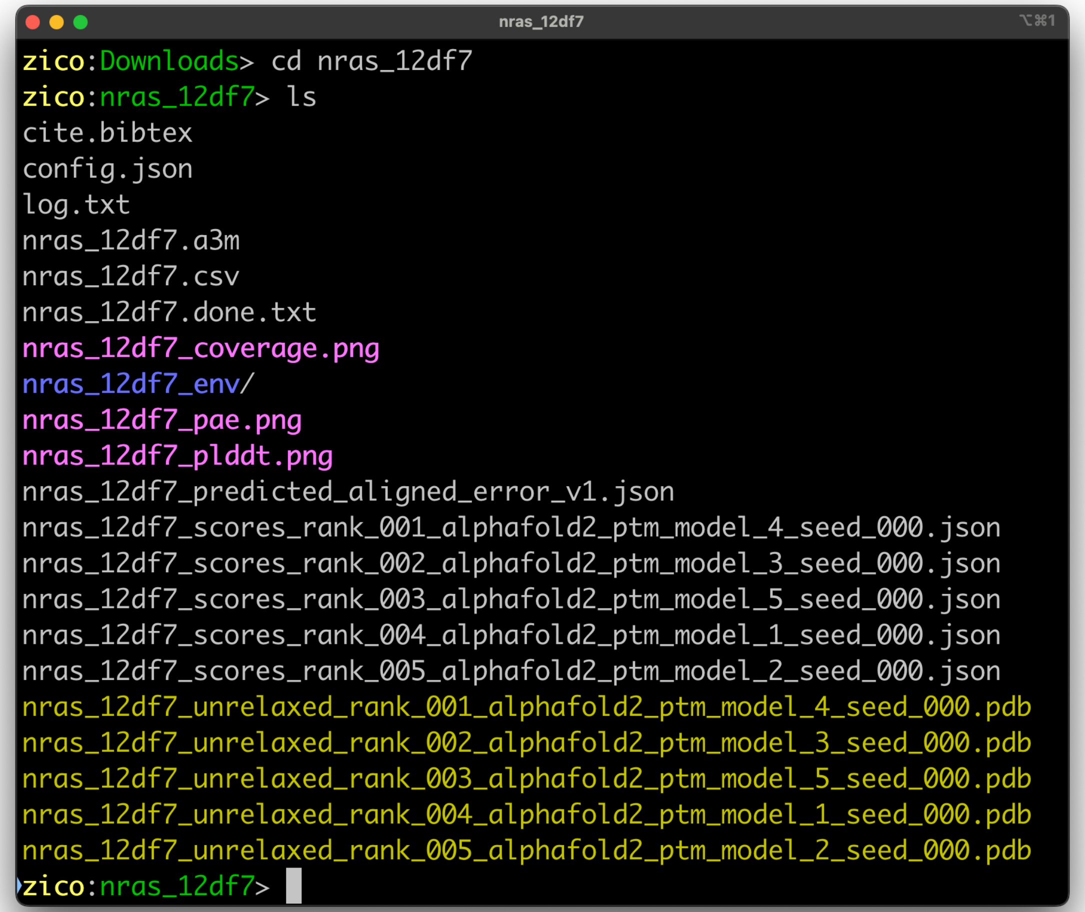
Visualization of the models and their estimated reliability

We can use Mol* for visualization of your model PDB files.

> Mol* (pronounced “molstar”) is a new web-based molecular viewer that is rapidly gaining in popularity and utility. At the time of writing it is still some way from having the full feature set of stand-alone molecular viewer programs like VMD, PyMol or Chimera. However, it is gaining new features all the time and does not require any download or complicated installation.

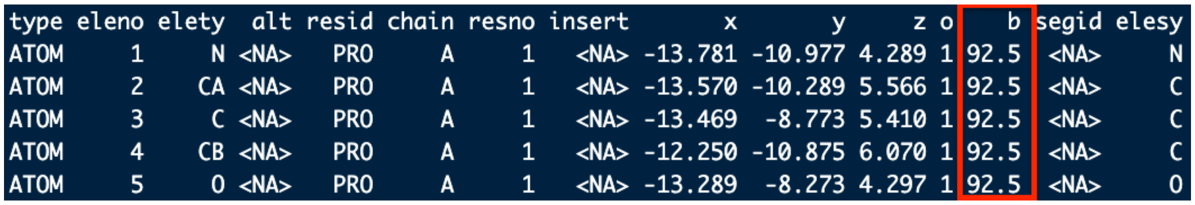
Regardless of which molecular viewer you use it is important to note that the B-factor column of each AlphaFold produced PDB file contains the **per-residue pLDDT score** - essentially a measure of the estimated reliability, or confidence score per-residue (Figure 14).

As previously noted these pLDDT values range between 0 and 100. Values above 70 are considered high confidence whereas those scored below 50 are low confidence and hence unreliable. Circumstantial evidence suggests that these low confidence regions may be unstructured in isolation (we will talk more about their interpretation below).



```
zico:Downloads> cd nras_12df7
zico:nras_12df7> ls
cite.bibtex
config.json
log.txt
nras_12df7.a3m
nras_12df7.csv
nras_12df7.done.txt
nras_12df7_coverage.png
nras_12df7_env/
nras_12df7_pae.png
nras_12df7_plddt.png
nras_12df7_predicted_aligned_error_v1.json
nras_12df7_scores_rank_001_alphafold2_ptm_model_4_seed_000.json
nras_12df7_scores_rank_002_alphafold2_ptm_model_3_seed_000.json
nras_12df7_scores_rank_003_alphafold2_ptm_model_5_seed_000.json
nras_12df7_scores_rank_004_alphafold2_ptm_model_1_seed_000.json
nras_12df7_scores_rank_005_alphafold2_ptm_model_2_seed_000.json
nras_12df7_unrelaxed_rank_001_alphafold2_ptm_model_4_seed_000.pdb
nras_12df7_unrelaxed_rank_002_alphafold2_ptm_model_3_seed_000.pdb
nras_12df7_unrelaxed_rank_003_alphafold2_ptm_model_5_seed_000.pdb
nras_12df7_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb
nras_12df7_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
zico:nras_12df7>
```

Figure 13: Result files from ColabFold include PDB structures and JSON format pLDDT scores



```
type eleno elety  alt resid chain resno insert       x       y     z o    b segid elesy
ATOM     1     N <NA>   PRO     A     1   <NA> -13.781 -10.977 4.289 1 92.5  <NA>     N
ATOM     2    CA <NA>   PRO     A     1   <NA> -13.570 -10.289 5.566 1 92.5  <NA>     C
ATOM     3     C <NA>   PRO     A     1   <NA> -13.469  -8.773 5.410 1 92.5  <NA>     C
ATOM     4    CB <NA>   PRO     A     1   <NA> -12.250 -10.875 6.070 1 92.5  <NA>     C
ATOM     5     O <NA>   PRO     A     1   <NA> -13.289  -8.273 4.297 1 92.5  <NA>     O
```

Figure 14: AlphaFold PDB files have computed per-residue pLDDT scores in their B-factor column (denoted “b”) above. This allows molecular viewers such as Mol* to color atoms by these values. Later in this lab we will map residue conservation to the “o” (occupancy) column to again allow color-codding of the structure by this data.

- Visit the main Mol* page and in the left panel click **Open Files** > **Select Files...** and chose all your downloaded PDB files. Then click **Apply** to load all these structures (a total of 5 structures if you ran with ColabFold with the default values above. Figure 15).

- Note that these structures are not in the same reference frame. To fix this we will **superpose** all structures. First, using the *selection arrow icon* on the top right side of the viewing window (red circle in Figure 16). Change the selection mode from "residue" to "chain instance" (red line toward the top of Figure 16).

- Once in chain selection mode, click on each chain/molecule in the main display window to select all 5 structure models. They should now appear highlighted with a green glow (Figure 16).

- On the right panel click on the **Superpose** menu and its "Chains" entry to expand it (Figure 16). Now click on superpose (red rectangle in Figure 16). Once they are fitted on top of each other turn the selection menu mode back to residue and turn off selection mode by clicking the "x" icon. **N.B. It is important you exit "selection mode" or things will not work as expected below.**

- Examine the superposed structures (Figure 17) via rotation, zoom and pan features to focus on variable and invariable structural regions. Can you identify the most variable regions by eye?

- Let's now color each structure by their computed pLDDT scores contained in the B-factor column (a.k.a "Uncertainty/Disorder" in Mol* parlance). To do this find the "Polymer Cartoon" entry under **Components** in the right side panel. Click the three dots (...) to expand the cartoon style settings and select **Set Coloring** > **Atom Property** > **Uncertainty Disorder** (Figure 18).

This will set a default color pallet of red-white-blue (red for high confidence, blue for low confidence). You can change this if you wish under Cartoon Representation > Color Theme.

Note any areas of lower model quality. These are typically surface exposed loop regions and extend termini or other linking chain segments between otherwise well predicted domains. It is common for models to have high confidence in a folded domain, and low confidence in a segment that is not part of a compact domain. Low-confidence segments may be intrinsically disordered. It is useful to compare predictions of disorder with AlphaFold reliability estimates.

Other visualizations to explore include coloring the **Residue property** of *Accessible Surface Area*, *Secondary structure*, and *Hydrophobicity*. Note the distribution of hydrophobic vs. polar residues. Integral membrane proteins will have large hydrophobic surfaces while soluble proteins will have hydrophobic cores.

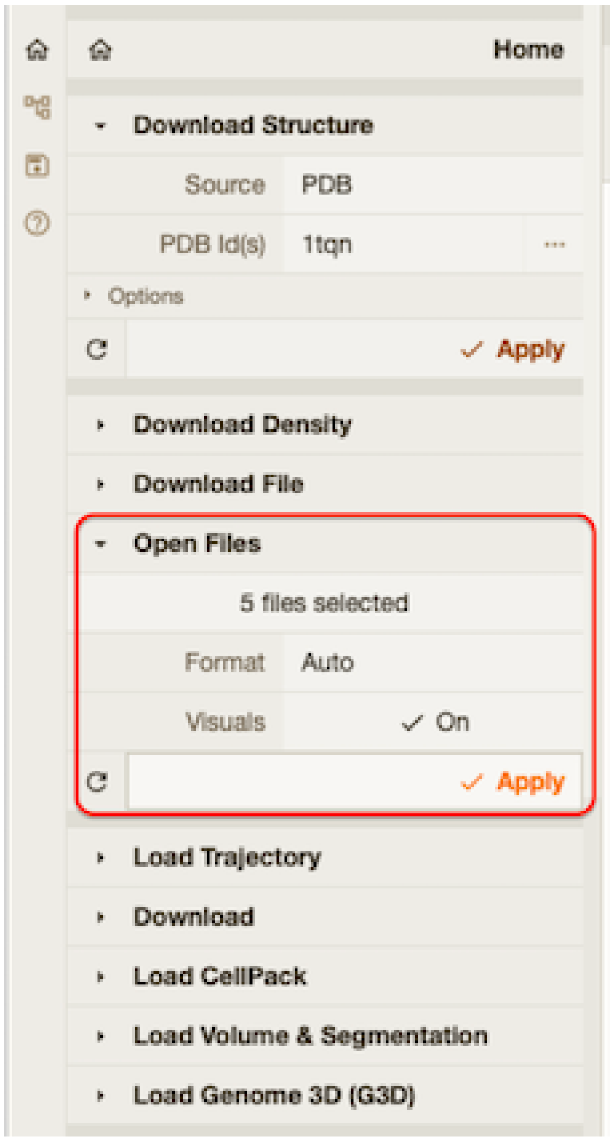
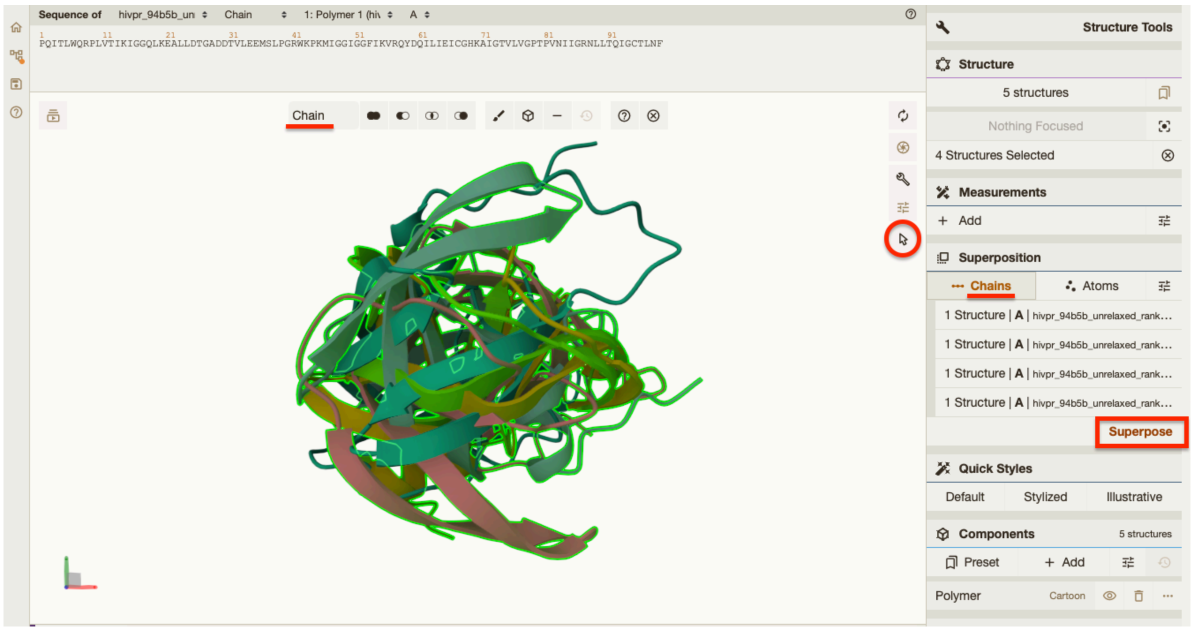
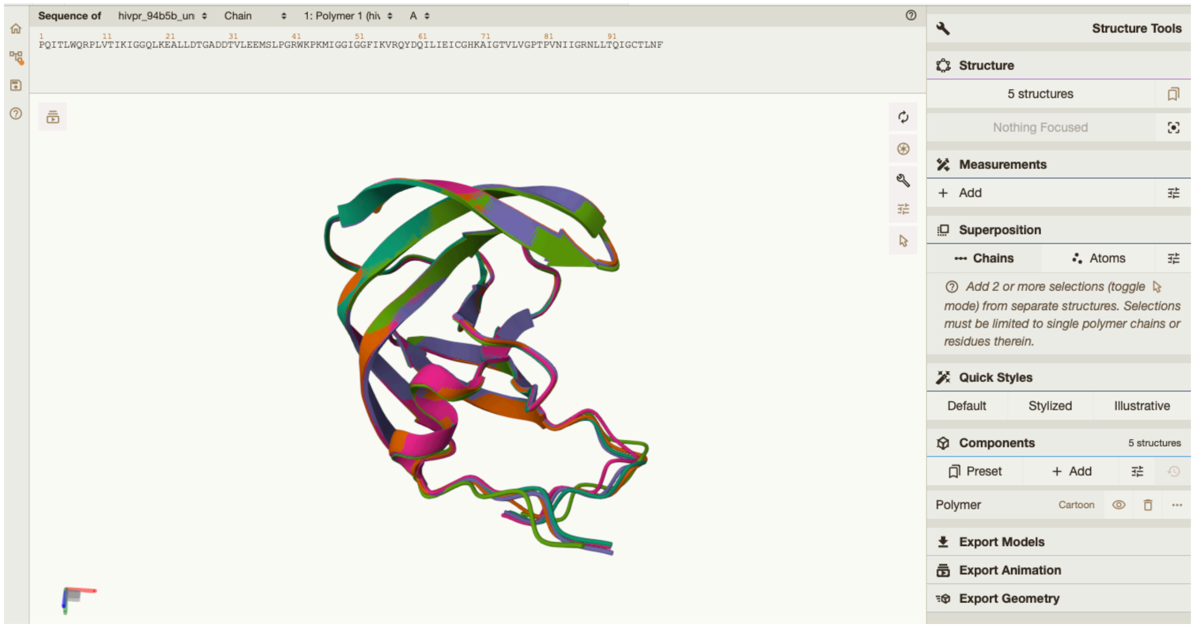


Figure 15: Mol* can open multipe PDB files from your computer at the same time. Be sure to select all your models and click the correct Apply button to have these load properly.



```
1         11        21        31        41        51        61        71        81        91
PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
```

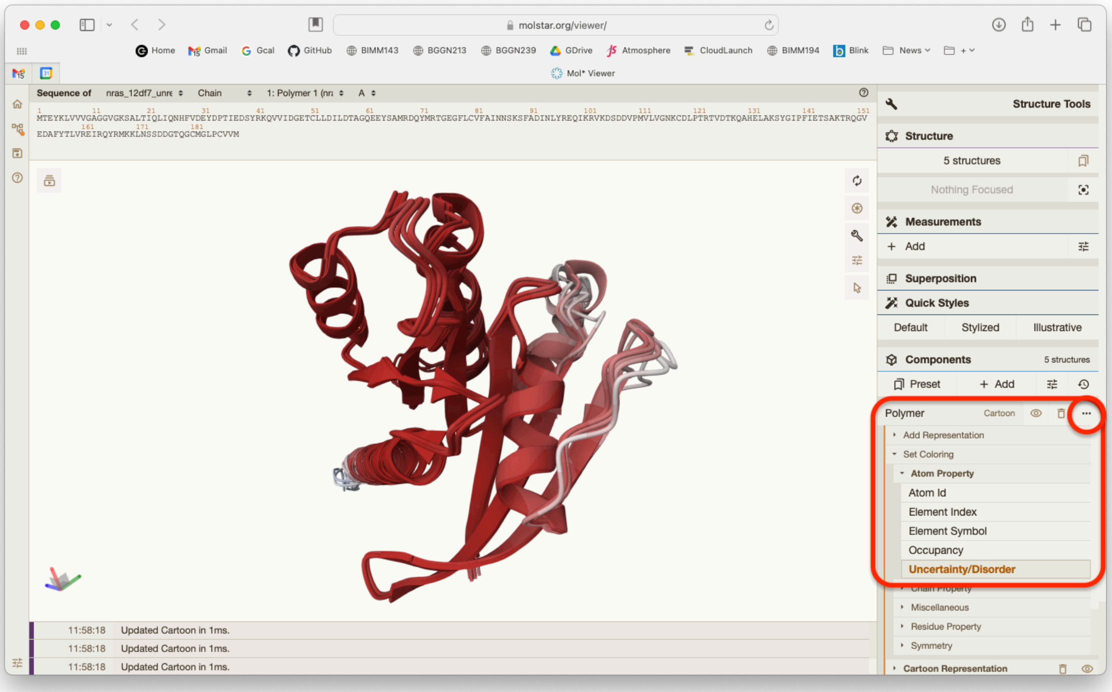
Figure 16: Performing superposition (a.k.a. fitting) in Mol* is a multi-step process that entails first selecting a single chain from each structure you wish to fit and then performing the superposition based on sequence alignment of these chains.



```
1         11        21        31        41        51        61        71        81        91
PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
```

Add 2 or more selections (toggle mode) from separate structures. Selections must be limited to single polymer chains or residues therein.

Figure 17: Supperposed monomer models



```
MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAGQEEYSAMRDQYMRTGEGFLCVFAINNSKSFADINLYREQIKRVKDSDDVPMVLVGNKCDLPTRTVDTKQAHELAKSYGIPFIETSAKTRQGV
EDAFYTLVREIRQYRMKKLNSSDDGTQGCMGLPCVVM
```

```
11:58:18    Updated Cartoon in 1ms.
11:58:18    Updated Cartoon in 1ms.
11:58:18    Updated Cartoon in 1ms.
```

Figure 18: Fitted and pLDDT colored Nras structure models. Your HIV-Pr monomers (and later dimers) will look somewhat less intresting as the entire model is generally high quality but note that there are structural regions of the monomer that do not make much sense in the absence of it's partner chain - this is because AlphaFold does not know any physics as we have tun it here.

8. Custom analysis of resulting models

In this section we will read the results of the more complicated HIV-Pr dimer AlphaFold2 models into R with the help of the Bio3D package. You can do the same thing for the monomer models if you wish but again they will be less interesting as the monomer is not physiologically relevant.

For tidiness we can move our AlphaFold results directory into our RStudio project directory. In this example my results are in the director `results_dir` (set below). You should change this to match your directory/folder name.

```
# Change this for YOUR results dir name
results_dir <- "hivprdimer_23119/"
```

```
# File names for all PDB models
pdb_files <- list.files(path=results_dir,
                        pattern="*.pdb",
                        full.names = TRUE)

# Print our PDB file names
basename(pdb_files)
```

```
[1] "hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

```
library(bio3d)

# Read all data from Models
#  and superpose/fit coords
pdbs <- pdbaln(pdb_files, fit=TRUE, exefile="msa")
```

If your `pdbaln()` function gives an error message then you likely do not have the `msa` package from BioConductor installed correctly. You will need to run `install.packages("BiocManager")` and then `BiocManager::install("msa")` in your console.

A quick view of model sequences - this should be a boring alignment in the sense that all sequences are the same.

```
pdbs
```

```
                                1        .         .         .         .         50
[Truncated_Name:1]hivprdimer   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]hivprdimer   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:3]hivprdimer   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]hivprdimer   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:5]hivprdimer   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                               **************************************************
                                1        .         .         .         .         50

                               51        .         .         .         .         100
[Truncated_Name:1]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:3]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               **************************************************
                               51        .         .         .         .         100

                              101        .         .         .         .         150
[Truncated_Name:1]hivprdimer   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]hivprdimer   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:3]hivprdimer   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:4]hivprdimer   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:5]hivprdimer   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                               **************************************************
                              101        .         .         .         .         150

                              151        .         .         .         .       198
[Truncated_Name:1]hivprdimer   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]hivprdimer   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]hivprdimer   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]hivprdimer   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]hivprdimer   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                               ************************************************
                              151        .         .         .         .       198

Call:
  pdbaln(files = pdb_files, fit = TRUE, exefile = "msa")

Class:
  pdbs, fasta

Alignment dimensions:
  5 sequence rows; 198 position columns (198 non-gap, 0 gap)

+ attr: xyz, resno, b, chain, id, ali, resid, sse, call
```

RMSD is a standard measure of structural distance between coordinate sets. We can use the `rmsd()` function to calculate the RMSD between all pairs models.

```r
rd <- rmsd(pdbs, fit=T)
```

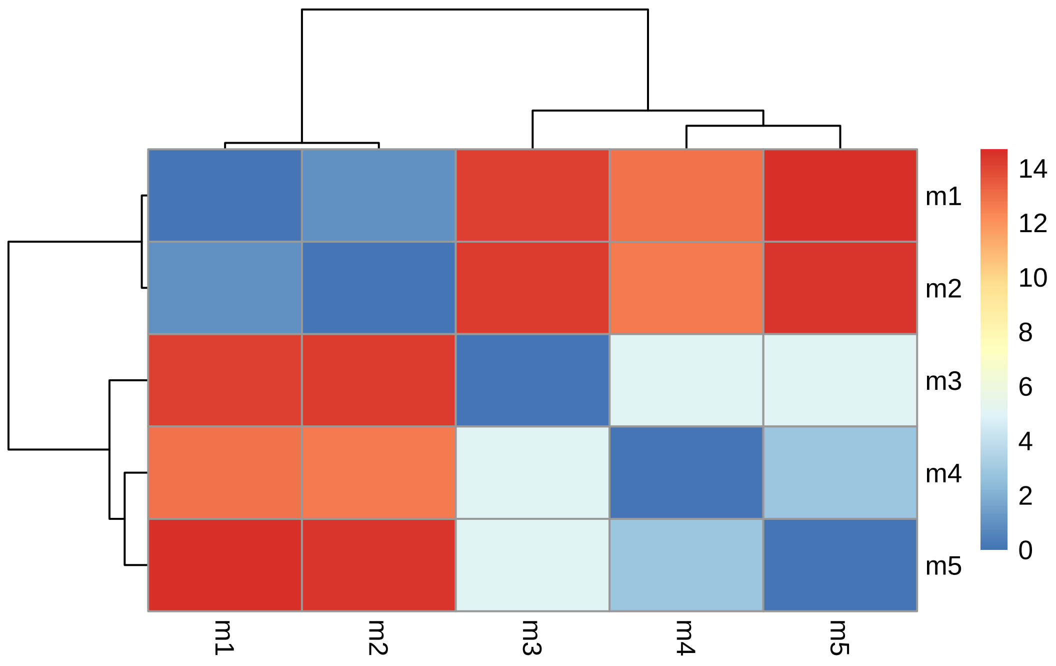
```r
range(rd)
```

```
[1]  0.000 14.689
```

Draw a heatmap of these RMSD matrix values

```r
library(pheatmap)

colnames(rd) <- paste0("m",1:5)
rownames(rd) <- paste0("m",1:5)
pheatmap(rd)
```



Here we can see that models 1 and 2 are more similar to each other than they are to any other model. Models 4 and 5 are quite similar to each other and in turn more similar to model 3 than to models 1 and 2. We will see this trend again in the pLDDT and PAE plots further below.

Now lets plot the pLDDT values across all models. Recall that this information is in the B-factor column of each model and that this is stored in our aligned `pdbs` object as `pdbs$b` with a row per structure/model.

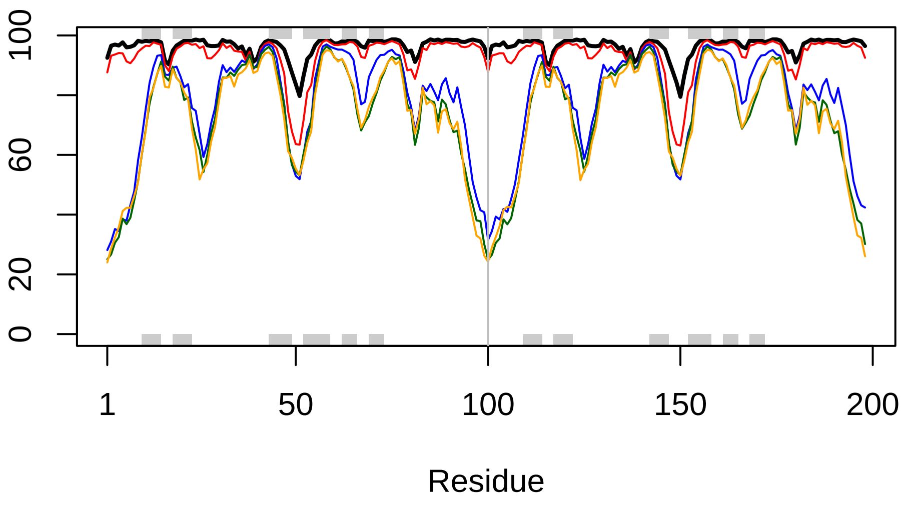
```r
# Read a reference PDB structure
pdb <- read.pdb("1hsg")
```

```
  Note: Accessing on-line PDB file
```

You could optionally obtain secondary structure from a call to `stride()` or `dssp()` on any of the model structures.

```r
plotb3(pdbs$b[1,], typ="l", lwd=2, sse=pdb)
points(pdbs$b[2,], typ="l", col="red")
points(pdbs$b[3,], typ="l", col="blue")
points(pdbs$b[4,], typ="l", col="darkgreen")
points(pdbs$b[5,], typ="l", col="orange")
```

```
abline(v=100, col="gray")
```



We can improve the superposition/fitting of our models by finding the most consistent "rigid core" common across all the models. For this we will use the `core.find()` function:

```
core <- core.find(pdbs)
```

We can now use the identified core atom positions as a basis for a more suitable superposition and write out the fitted structures to a directory called `corefit_structures`:

```
core.inds <- print(core, vol=0.5)
```

```
# 80 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  25     16
2    27  48     22
3    53  94     42
```

```
xyz <- pdbfit(pdbs, core.inds, outpath="corefit_structures")
```

The resulting superposed coordinates are written to a new director called `corefit_structures/`. We can now open these in Mol* and color by the **Atom Property** of **Uncertainty/Disorder** (i.e. the **B-factor** column that contains the pLDDT scores Figure 19):

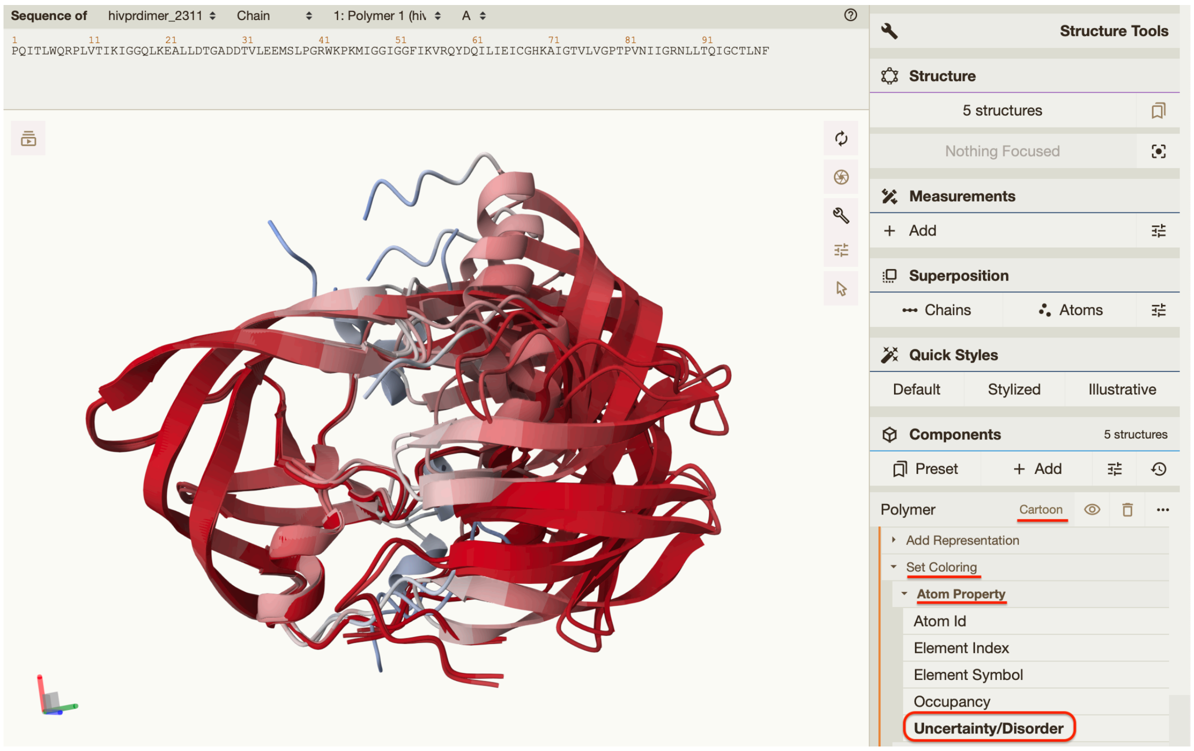
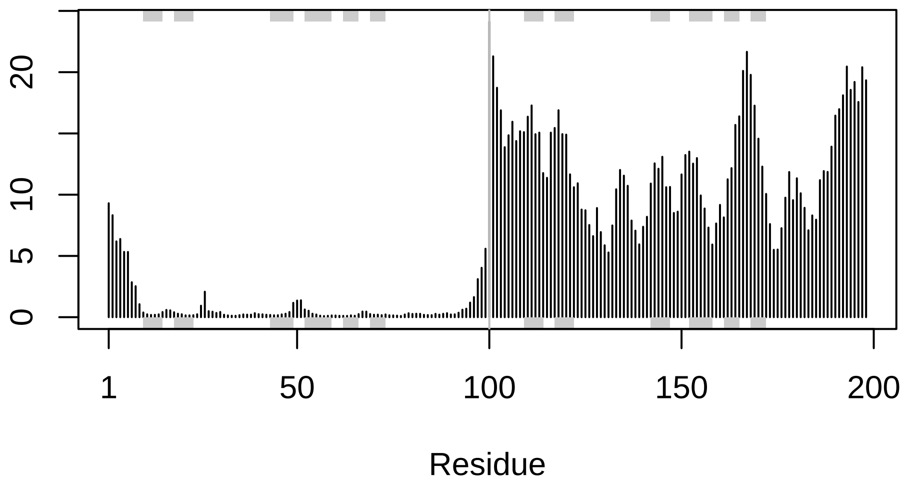


Figure 19: Core superposed structures colored by B-factor i.e. pLDDT. Note that you can toggel the display of the various models to examine each one in turn. Note that models 3, 4, and 5 are quite dissimilar and of questionable quaility as we will see with thir PAE scored below.

Now we can examine the RMSF between positions of the structure. RMSF is an often used measure of conformational variance along the structure:

```r
rf <- rmsf(xyz)

plotb3(rf, sse=pdb)
abline(v=100, col="gray", ylab="RMSF")
```



Here we see that the first chain is largely very similar across the different models. However, the second chain is much more variable - we saw this in Mol* previously (Figure 19).

Predicted Alignment Error for domains

Independent of the 3D structure, AlphaFold produces an output called **Predicted Aligned Error** (**PAE**). This is detailed in the JSON format result files, one for each model structure.

Below we read these files and see that AlphaFold produces a useful inter-domain prediction for model 1 (and 2) but not for model 5 (or indeed models 3, 4, and 5):

```
library(jsonlite)

# Listing of all PAE JSON files
pae_files <- list.files(path=results_dir,
                        pattern=".*model.*\\.json",
                        full.names = TRUE)
```

For example purposes lets read the 1st and 5th files (you can read the others and make similar plots).

```r
pae1 <- read_json(pae_files[1],simplifyVector = TRUE)
pae5 <- read_json(pae_files[5],simplifyVector = TRUE)

attributes(pae1)
```

```
$names
[1] "plddt"   "max_pae" "pae"     "ptm"     "iptm"
```

```r
# Per-residue pLDDT scores
#  same as B-factor of PDB..
head(pae1$plddt)
```

```
[1] 92.50 96.56 96.94 96.62 97.69 96.00
```

The maximum PAE values are useful for ranking models. Here we can see that model 5 is much worse than model 1. The lower the PAE score the better. How about the other models, what are thir max PAE scores?

```r
pae1$max_pae
```

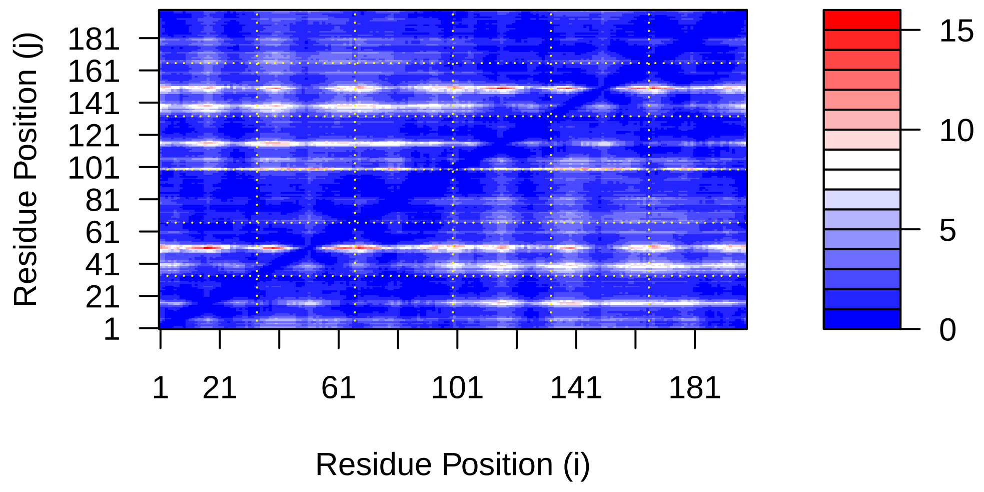
```
[1] 15.54688
```

```r
pae5$max_pae
```

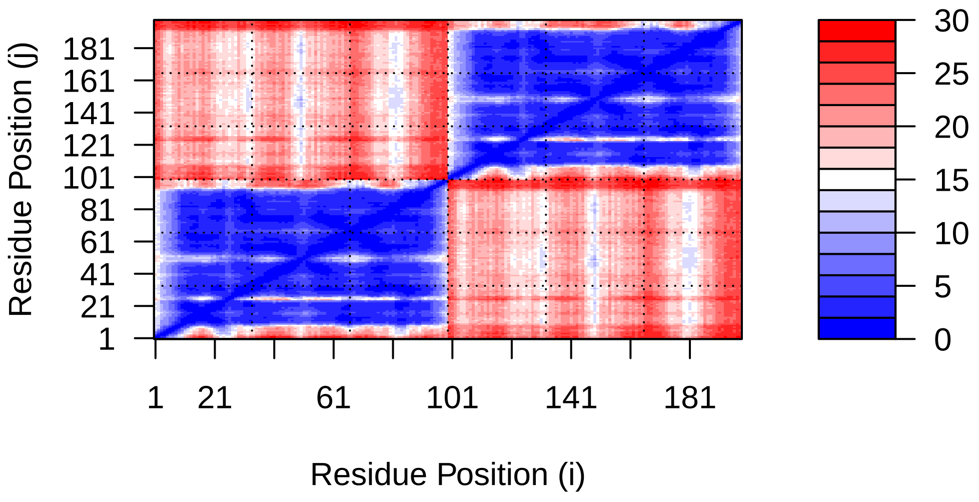
```
[1] 29.29688
```

We can plot the N by N (where N is the number of residues) PAE scores with ggplot or with functions from the Bio3D package:

```r
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```

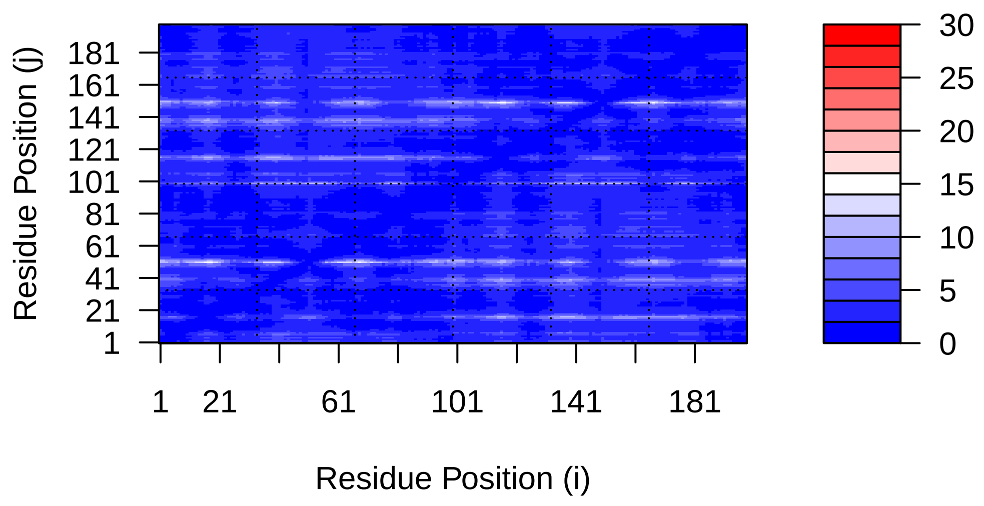


```r
plot.dmat(pae5$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



We should really plot all of these using the same z range. Here is the model 1 plot again but this time using the same data range as the plot for model 5:

```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



Residue conservation from alignment file

```
aln_file <- list.files(path=results_dir,
                        pattern=".a3m$",
                         full.names = TRUE)
aln_file
```

```
[1] "hivprdimer_23119//hivprdimer_23119.a3m"
```

```
aln <- read.fasta(aln_file[1], to.upper = TRUE)
```

```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

How many sequences are in this alignment

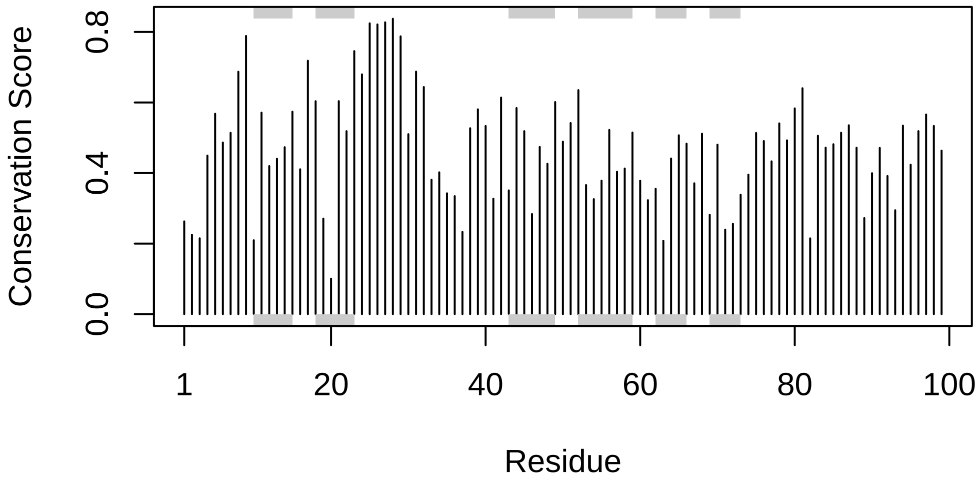
```
dim(aln$ali)
```

```
[1] 5378  132
```

We can score residue conservation in the alignment with the `conserv()` function.

```r
sim <- conserv(aln)
```

```r
plotb3(sim[1:99], sse=trim.pdb(pdb, chain="A"),
       ylab="Conservation Score")
```



Note the conserved Active Site residues D25, T26, G27, A28. These positions will stand out if we generate a consensus sequence with a high cutoff value:

```r
con <- consensus(aln, cutoff = 0.9)
con$seq
```

```
  [1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [19] "-" "-" "-" "-" "-" "-" "D" "T" "G" "A" "-" "-" "-" "-" "-" "-" "-" "-"
 [37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-"
```

For a final visualization of these functionally important sites we can map this conservation score to the Occupancy column of a PDB file for viewing in molecular viewer programs such as Mol*, PyMol, VMD, chimera etc.

```
m1.pdb <- read.pdb(pdb_files[1])
occ <- vec2resno(c(sim[1:99], sim[1:99]), m1.pdb$atom$resno)
write.pdb(m1.pdb, o=occ, file="m1_conserv.pdb")
```

Here is an image of this data generated from and Mol* using coloring by Occupancy. This is done in a similar manor to the pLDDT coloring procedure detailed above (Figure 20).

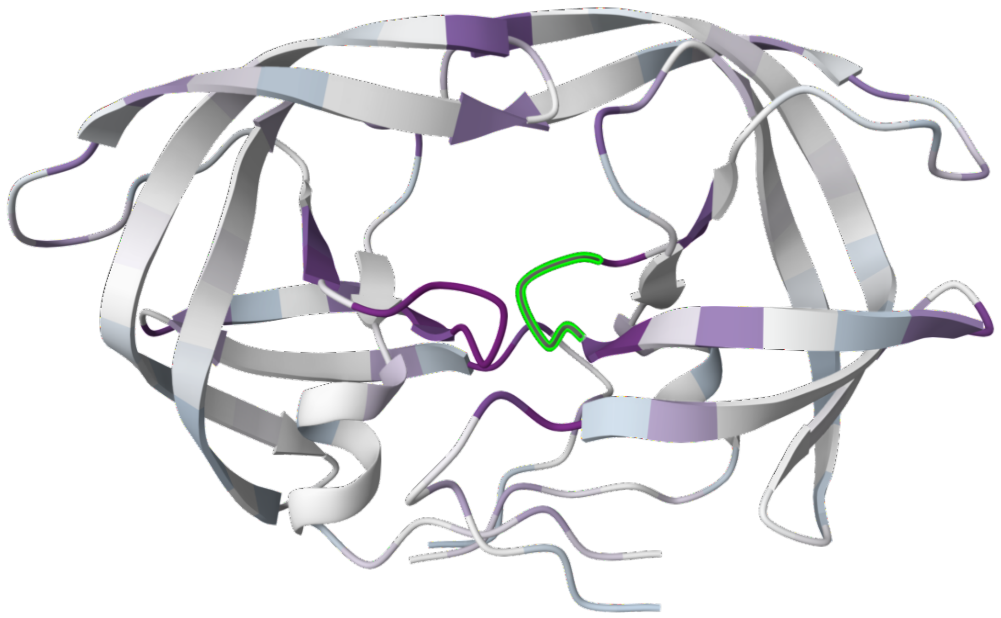


Figure 20: Top ranked dimer model colored by sequence conservation. Conserved positions in a darker purple. The DTGA motif of one chain is highlighted in green

Note that we can now clearly see the central conserved active site in this model where the natural peptide substrate (and small molecule inhibitors) would bind between domains.

9. Curent limitations and potential problems

If something goes wrong with your ColabFold run, you only real option is to load the site over again (i.e. refresh the webpage and start from the beginning). Colab notebooks can crash or time-out at any time. If you are running multiple predictions you could therefore lose a lot of work.

You can mitigate this potential lose by manually downloading results as they appear (using the folder icon on the left side of the notebook, selecting a .zip file to show a download menu, and downloading the file).

Multimer vs single chain predictions

AlphaFold performs best with single chains, which may include one or a few domains. Whilst we have used here the AlphaFold2-multimer notebooks for complex prediction and it gives reasonable results - this is not always the case. Currently these multimer approaches are known to be much less successful. Accurately predicting hetero-complexes is still an area of active research. As is placing small-molecule and other interface site prediction.

Predicting diferent conformational states

The most important limitation of AlphaFold predictions is that only a single state is predicted, even if hints for multiple states and dynamic behavior are in the data, like for our Nras predictions above with the active site “switch regions” clearly shown as more variable regions.

It is also hard to tell which state of a protein will be captured by the AI - it all depends what was in the training dataset. For now at least AF only predicts a single state and does not explain functional dynamic behavior.

Relative Positions of Domains

If the predicted model has more than one domain, each domain may have high confidence, yet the relative positions of the domains may not. The estimated reliability of relative domain positions is in graphs of predicted aligned error (PAE) which are included in the downloadable zip file and analyzed in R above.

Poorely predicted regions, intrinsic disorder or “bugs”

There are also protein regions that AF cannot predict, and it will be important to find out why. Which regions will have stable folds, in isolation or in complexes, that AlphaFold has missed? Which fraction will consist of truly intrinsically disordered regions (IDRs) that are used for instance in phase separation? As more structures and sequences become available, and as the methods further improve, it seems likely that the fraction of poorly predicted protein will decrease.

10. Summary

In a sense AF provides all biologists with a new technique, bringing the fun of structure-gazing without the effort of experimental structure determination work.

In my view it is crucial that we educate the next generation of biologists to learn how to critically analyze predicted structures, notice new interactions, and to get to know each protein of interest in sufficient detail, so as to differentiate between "bugs" and "features".

At the time of writing this lab the current AlphaFold implementation does not yet have the accuracy that is necessary for drug discovery with traditional computational docking approaches. This should not be a supervise (as we will discuss in class). However, new versions of AlphaFold, and the AlphaFold inspired RosseTTAFold (Baek et al. 2021), are rumored that will more explicitly include ligand interactions (and potentially biological therapeutics, such as antibodies or nanobodies) in their training data. It is an exciting time for structural biology and drug discovery.

About this document

Here we use the `sessionInfo()` function to report on our R systems setup at the time of document execution.

```r
sessionInfo()
```

```
R version 4.1.2 (2021-11-01)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] jsonlite_1.8.4  pheatmap_1.0.12 bio3d_2.4-4
```

```
loaded via a namespace (and not attached):
 [1] Rcpp_1.0.10            msa_1.26.0             rstudioapi_0.15.0
 [4] knitr_1.44             XVector_0.34.0         zlibbioc_1.40.0
 [7] BiocGenerics_0.40.0    IRanges_2.28.0         munsell_0.5.0
[10] colorspace_2.1-0       R6_2.5.1               rlang_1.1.1
[13] fastmap_1.1.1          GenomeInfoDb_1.30.1    tools_4.1.2
[16] parallel_4.1.2         grid_4.1.2             gtable_0.3.4
[19] xfun_0.39              cli_3.6.1              htmltools_0.5.5
[22] yaml_2.3.7             digest_0.6.31          lifecycle_1.0.3
[25] crayon_1.5.2           GenomeInfoDbData_1.2.7 RColorBrewer_1.1-3
[28] S4Vectors_0.32.4       bitops_1.0-7           RCurl_1.98-1.12
[31] glue_1.6.2             evaluate_0.22          rmarkdown_2.25
[34] compiler_4.1.2         scales_1.2.1           Biostrings_2.62.0
[37] stats4_4.1.2
```

References:

If you use a model from the AlphaFold CoLab notebook you should be sure to cite the following two publications in your work:

Baek, Minkyung, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, et al. 2021. "Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network." *Science* 373 (6557): 871–76. https://doi.org/10.1126/science.abj8754.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, et al. 2021. "Highly Accurate Protein Structure Prediction with AlphaFold." *Nature* 596 (7873): 583–89. https://doi.org/10.1038/s41586-021-03819-2.

Mirdita, Milot, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. 2022. "ColabFold: Making Protein Folding Accessible to All." *Nature Methods* 19 (6): 679–82. https://doi.org/10.1038/s41592-022-01488-1.