ColabFold v1.5.5: 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 Nature Protocols.

Old versions: v1.4, v1.5.1, v1.5.2, v1.5.3-patch

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

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

query_sequence: " MRTLNTSAMDGTGLVVERDFSVRILTACFLSLLILSTLLGNTLVC "

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

jobname: " group7_DRD1_multimer "

num_relax: 1

- 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)

Show code

```
jobname group7_DRD1_multimer_17ee1
sequence MRTLNTSAMDGTGLVVERDFSVRILTACFLSLLILSTLLGNTLVCAAVIRFRHLRSKVTNFFVISLAV
length 1249
```

Install dependencies

Show code

```
installing colabfold...
installing conda...
installing amber...
CPU times: user 6.35 ms, sys: 170 µs, total: 6.52 ms
Wall time: 1min 6s
```

MSA options (custom MSA upload, single sequence, pairing mode)

msa_mode: mmseqs2_uniref_env

pair_mode: unpaired_paired

- "unpaired_paired" = pair sequences from same species + unpaired MSA, "unpaired" = seperate MSA for each chain, "paired" - only use paired sequences.

Show code

Advanced settings

model_type: alphafold2_multimer_v3

- if `auto` selected, will use `alphafold2_ptm` for monomer prediction and `alphafold2_multimer_v3` for complex prediction. Any of the mode_types can be used (regardless if input is monomer or complex).

num_recycles: 3

- if `auto` selected, will use `num_recycles=20` if `model_type=alphafold2_multimer_v3`, else `num_recycles=3` .

recycle_early_stop_tolerance: auto

- if `auto` selected, will use `tol=0.5` if `model_type=alphafold2_multimer_v3` else `tol=0.0`.

relax_max_iterations: 200

- max amber relax iterations, `0` = unlimited (AlphaFold2 default, can take very long)

pairing_strategy: greedy

- `greedy` = pair any taxonomically matching subsets, `complete` = all sequences have to match in one line.

`calc_extra_ptm:` ☐

- return pairwise chain iptm/actifptm

Sample settings

- enable dropouts and increase number of seeds to sample predictions from uncertainty of the model.
- decrease `max_msa` to increase uncertainity

`max_msa:` auto

`num_seeds:` 1

`use_dropout:` ☐

Save settings

`save_all:` ☐

`save_recycles:` ☐

`save_to_google_drive:` ☐

- if the save_to_google_drive option was selected, the result zip will be uploaded to your Google Drive

`dpi:` 200

- set dpi for image resolution

Don't forget to hit `Runtime` -> `Run all` after updating the form.

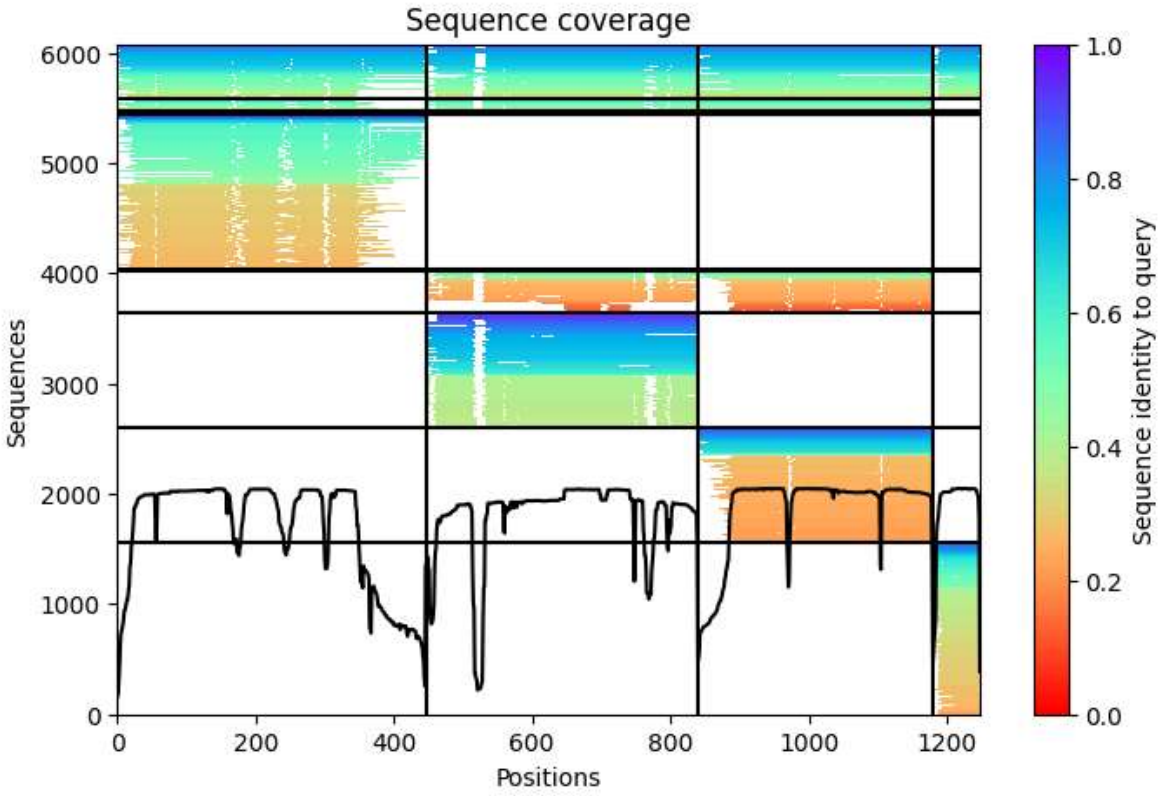
Show code

Run Prediction

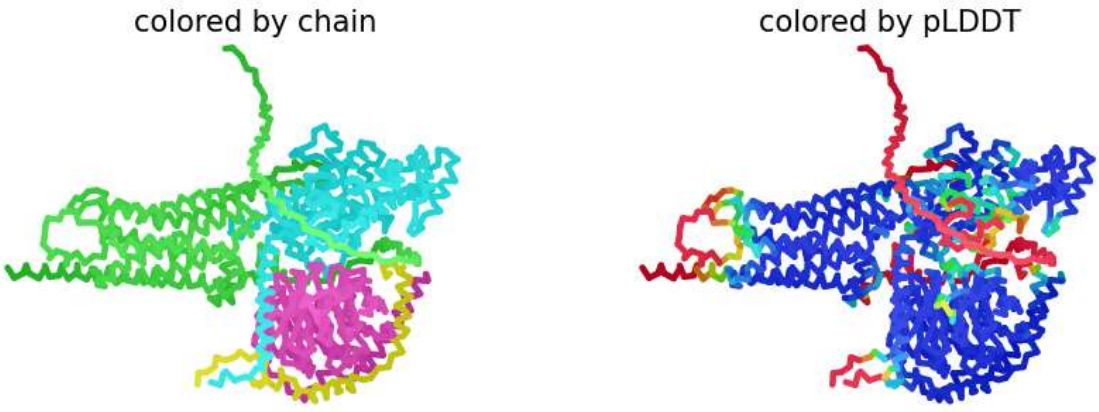
`display_images:` ☐

Show code

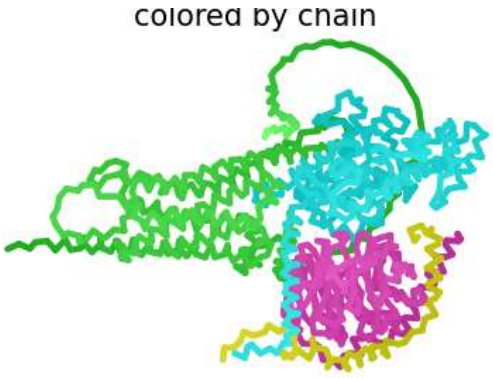
```
Downloading alphafold2_multimer_v3 weights to .: 100%|██████████| 3.82G/3.82G
2025-12-04 09:49:17,916 Running on GPU
2025-12-04 09:49:18,264 Found 6 citations for tools or databases
2025-12-04 09:49:18,264 Query 1/1: group7_DRD1_multimer_17ee1 (length 1249)
COMPLETE: 100%|██████████| 600/600 [elapsed: 00:02 remaining: 00:00]
COMPLETE: 100%|██████████| 600/600 [elapsed: 00:00 remaining: 00:00]
```

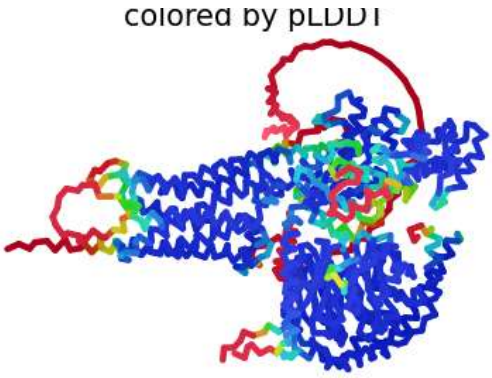



```
2025-12-04 09:49:30,598 Setting max_seq=508, max_extra_seq=2048
/content/alphafold/model/modules_multimer.py:114: UserWarning: Explicitly req
  iota = jax.lax.broadcasted_iota(jnp.int64, logits.shape, axis)
2025-12-04 09:56:56,799 alphafold2_multimer_v3_model_1_seed_000 recycle=0 pLD
2025-12-04 10:03:32,233 alphafold2_multimer_v3_model_1_seed_000 recycle=1 pLD
2025-12-04 10:09:32,671 alphafold2_multimer_v3_model_1_seed_000 recycle=2 pLD
2025-12-04 10:15:34,079 alphafold2_multimer_v3_model_1_seed_000 recycle=3 pLD
2025-12-04 10:15:34,101 alphafold2_multimer_v3_model_1_seed_000 took 1556.0s
```

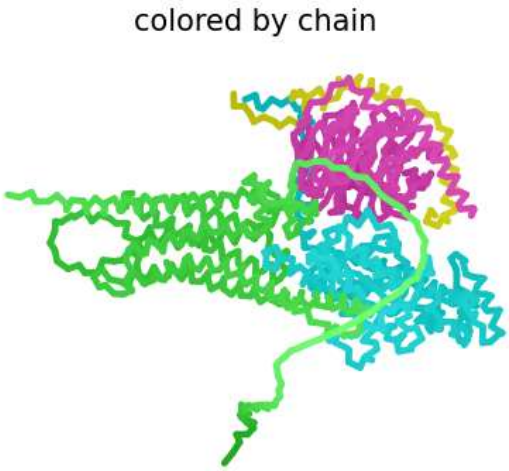


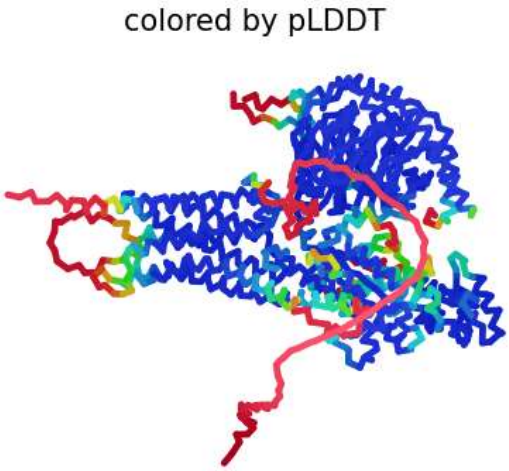

```
2025-12-04 10:21:35,700 alphafold2_multimer_v3_model_2_seed_000 recycle=0 pLD
2025-12-04 10:27:36,534 alphafold2_multimer_v3_model_2_seed_000 recycle=1 pLD
2025-12-04 10:33:36,399 alphafold2_multimer_v3_model_2_seed_000 recycle=2 pLD
2025-12-04 10:39:36,734 alphafold2_multimer_v3_model_2_seed_000 recycle=3 pLD
2025-12-04 10:39:36,751 alphafold2_multimer_v3_model_2_seed_000 took 1441.0s
```



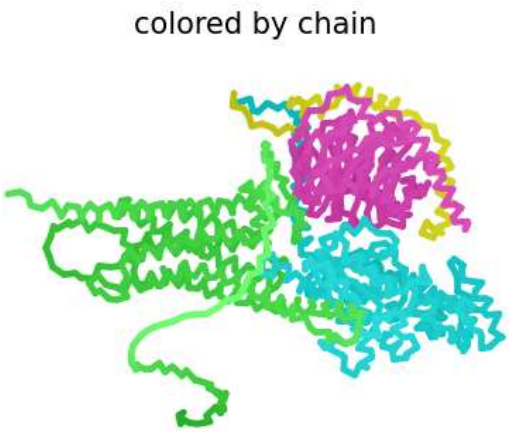


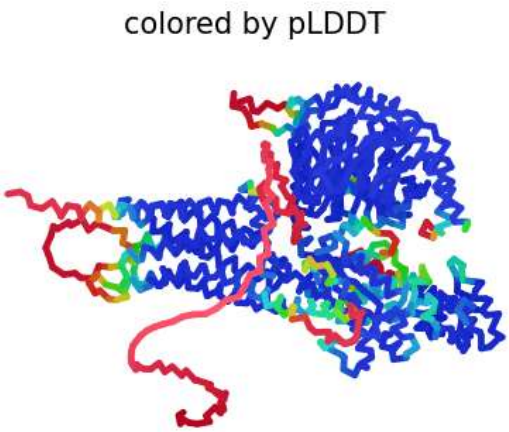

```
2025-12-04 10:45:37,890 alphafold2_multimer_v3_model_3_seed_000 recycle=0 pLD
2025-12-04 10:51:38,180 alphafold2_multimer_v3_model_3_seed_000 recycle=1 pLD
2025-12-04 10:57:37,707 alphafold2_multimer_v3_model_3_seed_000 recycle=2 pLD
2025-12-04 11:03:37,401 alphafold2_multimer_v3_model_3_seed_000 recycle=3 pLD
2025-12-04 11:03:37,412 alphafold2_multimer_v3_model_3_seed_000 took 1438.9s
```






```
2025-12-04 11:09:37,899 alphafold2_multimer_v3_model_4_seed_000 recycle=0 pLD
2025-12-04 11:15:38,058 alphafold2_multimer_v3_model_4_seed_000 recycle=1 pLD
2025-12-04 11:21:36,736 alphafold2_multimer_v3_model_4_seed_000 recycle=2 pLD
2025-12-04 11:27:36,161 alphafold2_multimer_v3_model_4_seed_000 recycle=3 pLD
2025-12-04 11:27:36,170 alphafold2_multimer_v3_model_4_seed_000 took 1437.1s
```






```
2025-12-04 11:33:36,524 alphafold2_multimer_v3_model_5_seed_000 recycle=0 pLD
```

```
2025-12-04 11:39:36,064 alphafold2_multimer_v3_model_5_seed_000 recycle=1 pLD
2025-12-04 11:45:34,875 alphafold2_multimer_v3_model_5_seed_000 recycle=2 pLD
2025-12-04 11:51:34,000 alphafold2_multimer_v3_model_5_seed_000 recycle=3 pLD
```

Display 3D structure

rank_num: 1

color: IDDT

show_sidechains: ☐

show_mainchains: ☐

Show code

Plots

Show code

Package and download results

If you are having issues downloading the result archive, try disabling your adblocker and run this cell again. If that fails click on the little folder icon to the left, navigate to file: `jobname.result.zip`, right-click and select "Download" (see screenshot).

Show code

Instructions

For detailed instructions, tips and tricks, see recently published paper at Nature Protocols

Quick start

1. Paste your protein sequence(s) in the input field.
2. Press "Runtime" -> "Run all".
3. The pipeline consists of 5 steps. The currently running step is indicated by a circle with a stop sign next to it.

Result zip file contents

1. PDB formatted structures sorted by avg. pLDDT and complexes are sorted by pTMscore. (unrelaxed and relaxed if `use_amber` is enabled).
2. Plots of the model quality.

3. Plots of the MSA coverage.
4. Parameter log file.
5. A3M formatted input MSA.
6. A `predicted_aligned_error_v1.json` using AlphaFold-DB's format and a `scores.json` for each model which contains an array (list of lists) for PAE, a list with the average pLDDT and the pTMscore.
7. BibTeX file with citations for all used tools and databases.

At the end of the job a download modal box will pop up with a `jobname.result.zip` file. Additionally, if the `save_to_google_drive` option was selected, the `jobname.result.zip` will be uploaded to your Google Drive.

MSA generation for complexes

For the complex prediction we use unpaired and paired MSAs. Unpaired MSA is generated the same way as for the protein structures prediction by searching the UniRef100 and environmental sequences three iterations each.

The paired MSA is generated by searching the UniRef100 database and pairing the best hits sharing the same NCBI taxonomic identifier (=species or sub-species). We only pair sequences if all of the query sequences are present for the respective taxonomic identifier.

Using a custom MSA as input

To predict the structure with a custom MSA (A3M formatted): (1) Change the `msa_mode`: to "custom", (2) Wait for an upload box to appear at the end of the "MSA options ..." box. Upload your A3M. The first fasta entry of the A3M must be the query sequence without gaps.

It is also possilbe to provide custom MSAs for complex predictions. Read more about the format here.

As an alternative for MSA generation the HHblits Toolkit server can be used. After submitting your query, click "Query Template MSA" -> "Download Full A3M". Download the A3M file and upload it in this notebook.

PDB100

As of 23/06/08, we have transitioned from using the PDB70 to a 100% clustered PDB, the PDB100. The construction methodology of PDB100 differs from that of PDB70.

The PDB70 was constructed by running each PDB70 representative sequence through HHblits against the Uniclust30. On the other hand, the PDB100 is built by searching each PDB100 representative structure with Foldseek against the AlphaFold Database.

To maintain compatibility with older Notebook versions and local installations, the generated files and API responses will continue to be named "PDB70", even though we're now using the PDB100.

Using custom templates

To predict the structure with a custom template (PDB or mmCIF formatted): (1) change the `template_mode` to "custom" in the execute cell and (2) wait for an upload box to appear at the end of the "Input Protein" box. Select and upload your templates (multiple choices are possible).

- Templates must follow the four letter PDB naming with lower case letters.
- Templates in mmCIF format must contain `_entity_poly_seq`. An error is thrown if this field is not present. The field `_pdbx_audit_revision_history.revision_date` is automatically generated if it is not present.
- Templates in PDB format are automatically converted to the mmCIF format. `_entity_poly_seq` and `_pdbx_audit_revision_history.revision_date` are automatically generated.

If you encounter problems, please report them to this issue.

Comparison to the full AlphaFold2 and AlphaFold2 Colab

This notebook replaces the homology detection and MSA pairing of AlphaFold2 with MMseqs2. For a comparison against the AlphaFold2 Colab and the full AlphaFold2 system read our paper.

Troubleshooting

- Check that the runtime type is set to GPU at "Runtime" -> "Change runtime type".
- Try to restart the session "Runtime" -> "Factory reset runtime".
- Check your input sequence.

Known issues

- Google Colab assigns different types of GPUs with varying amount of memory. Some might not have enough memory to predict the structure for a long sequence.
- Your browser can block the pop-up for downloading the result file. You can choose the `save_to_google_drive` option to upload to Google Drive instead or manually download the result file: Click on the little folder icon to the left, navigate to file: `jobname.result.zip`, right-click and select "Download" (see screenshot).

Limitations

- Computing resources: Our MMseqs2 API can handle ~20-50k requests per day.
- MSAs: MMseqs2 is very precise and sensitive but might find less hits compared to HHblits/HMMer searched against BFD or MGnify.
- We recommend to additionally use the full AlphaFold2 pipeline.

Description of the plots

- **Number of sequences per position** - We want to see at least 30 sequences per position, for best performance, ideally 100 sequences.
- **Predicted lDDT per position** - model confidence (out of 100) at each position. The higher the better.
- **Predicted Alignment Error** - For homooligomers, this could be a useful metric to assess how confident the model is about the interface. The lower the better.

Bugs

- If you encounter any bugs, please report the issue to https://github.com/sokrypton/ColabFold/issues

License

The source code of ColabFold is licensed under MIT. Additionally, this notebook uses the AlphaFold2 source code and its parameters licensed under Apache 2.0 and CC BY 4.0 respectively. Read more about the AlphaFold license here.

Acknowledgments

- We thank the AlphaFold team for developing an excellent model and open sourcing the software.
- KOBIC and Söding Lab for providing the computational resources for the MMseqs2 MSA server.
- Richard Evans for helping to benchmark the ColabFold's Alphafold-multimer support.
- David Koes for his awesome py3Dmol plugin, without whom these notebooks would be quite boring!
- Do-Yoon Kim for creating the ColabFold logo.
- A colab by Sergey Ovchinnikov (@sokrypton), Milot Mirdita (@milot_mirdita) and Martin Steinegger (@thesteinegger).