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: " test "

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)

Show code

```
jobname test_17ee1
sequence MRTLNTSAMDGTGLVVERDFSVRILTACFLSLLILSTLLGNTLVCAAVIRFRHLRSKVTNFFVISLAV
length 1249
```

Install dependencies

Show code

```
installing colabfold...
CPU times: user 1.49 ms, sys: 2.48 ms, total: 3.96 ms
Wall time: 50.8 s
```

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: auto

- 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

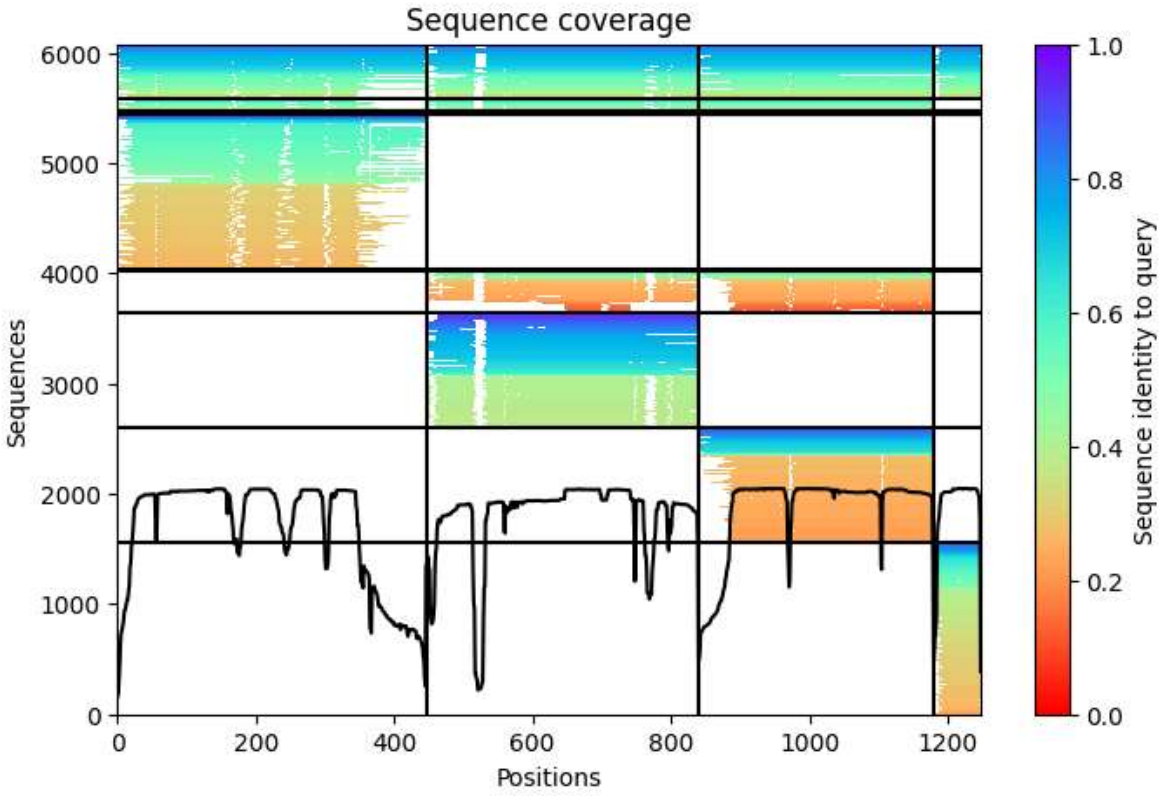
Don't forget to hit `Runtime` -> `Run all` after updating the form.

Run Prediction

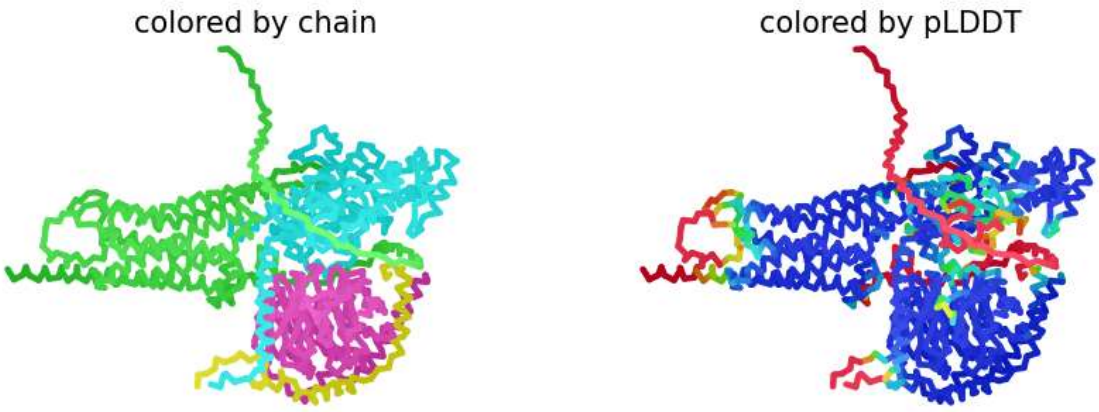
display_images: ☐

Show code

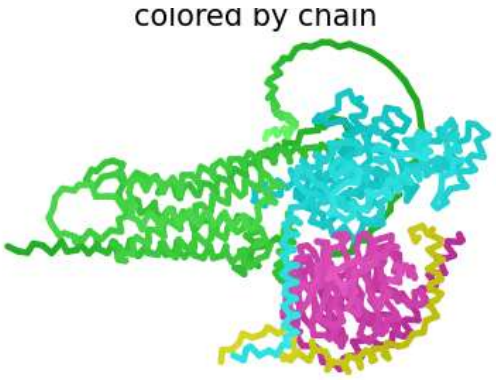
```
Downloading alphafold2_multimer_v3 weights to .: 100%|██████████| 3.82G/3.82G
2025-12-04 13:09:00,108 Running on GPU
2025-12-04 13:09:00,457 Found 5 citations for tools or databases
2025-12-04 13:09:00,457 Query 1/1: test_17ee1 (length 1249)
COMPLETE: 100%|██████████| 600/600 [elapsed: 00:02 remaining: 00:00]
COMPLETE: 100%|██████████| 600/600 [elapsed: 00:00 remaining: 00:00]
```

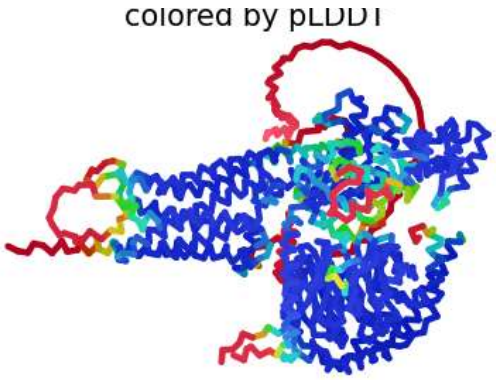



```
2025-12-04 13:09:12,910 Setting max_seq=508, max_extra_seq=2048
/content/alphafold/model/modules_multimer.py:114: UserWarning: Explicitly req
  iota = jax.lax.broadcasted_iota(jnp.int64, logits.shape, axis)
2025-12-04 13:16:51,424 alphafold2_multimer_v3_model_1_seed_000 recycle=0 pLD
2025-12-04 13:23:42,495 alphafold2_multimer_v3_model_1_seed_000 recycle=1 pLD
2025-12-04 13:29:57,380 alphafold2_multimer_v3_model_1_seed_000 recycle=2 pLD
2025-12-04 13:36:12,761 alphafold2_multimer_v3_model_1_seed_000 recycle=3 pLD
2025-12-04 13:36:12,782 alphafold2_multimer_v3_model_1_seed_000 took 1612.6s
```

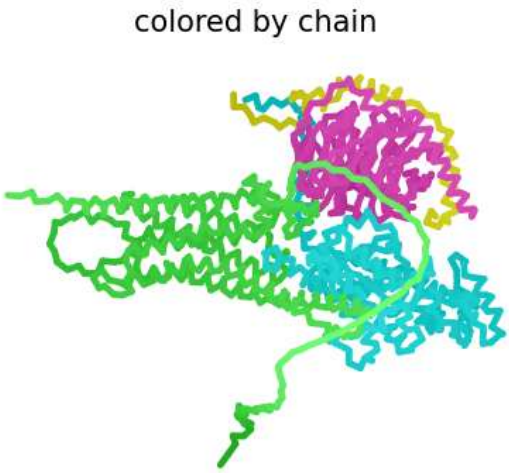


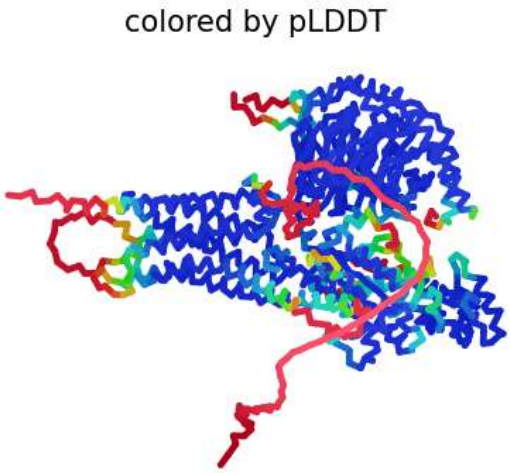

```
2025-12-04 13:42:27,881 alphafold2_multimer_v3_model_2_seed_000 recycle=0 pLD
2025-12-04 13:48:42,366 alphafold2_multimer_v3_model_2_seed_000 recycle=1 pLD
2025-12-04 13:54:57,176 alphafold2_multimer_v3_model_2_seed_000 recycle=2 pLD
2025-12-04 14:01:12,559 alphafold2_multimer_v3_model_2_seed_000 recycle=3 pLD
2025-12-04 14:01:12,577 alphafold2_multimer_v3_model_2_seed_000 took 1498.2s
```



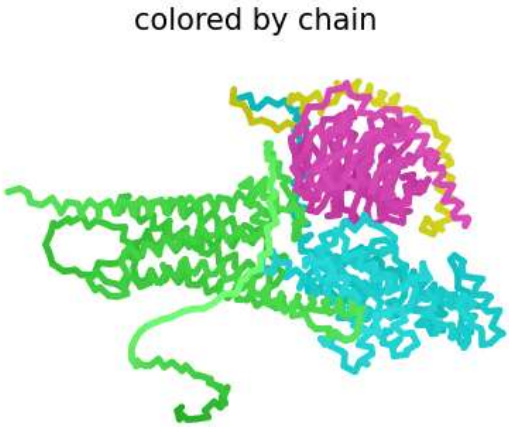


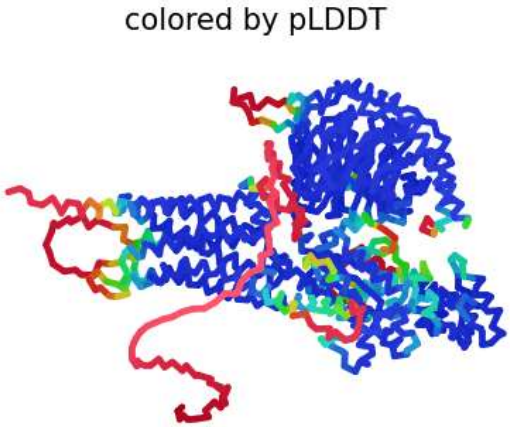

```
2025-12-04 14:07:28,565 alphafold2_multimer_v3_model_3_seed_000 recycle=0 pLD
2025-12-04 14:13:44,008 alphafold2_multimer_v3_model_3_seed_000 recycle=1 pLD
2025-12-04 14:19:58,653 alphafold2_multimer_v3_model_3_seed_000 recycle=2 pLD
2025-12-04 14:26:13,388 alphafold2_multimer_v3_model_3_seed_000 recycle=3 pLD
2025-12-04 14:26:13,401 alphafold2_multimer_v3_model_3_seed_000 took 1499.2s
```






```
2025-12-04 14:32:29,555 alphafold2_multimer_v3_model_4_seed_000 recycle=0 pLD
2025-12-04 14:38:44,324 alphafold2_multimer_v3_model_4_seed_000 recycle=1 pLD
2025-12-04 14:44:58,859 alphafold2_multimer_v3_model_4_seed_000 recycle=2 pLD
2025-12-04 14:51:14,183 alphafold2_multimer_v3_model_4_seed_000 recycle=3 pLD
2025-12-04 14:51:14,196 alphafold2_multimer_v3_model_4_seed_000 took 1499.2s
```

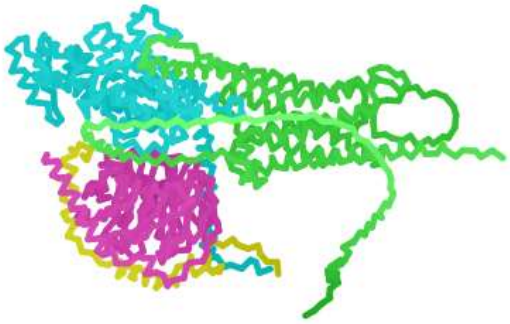




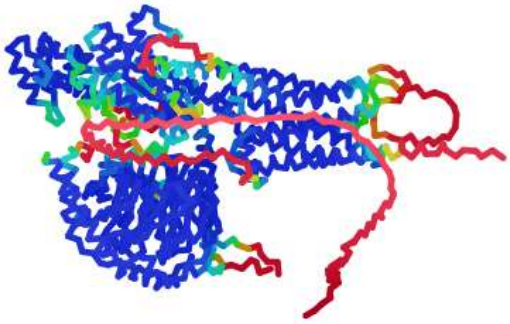

```
2025-12-04 14:57:30,020 alphafold2_multimer_v3_model_5_seed_000 recycle=0 pLD
```

```
2025-12-04 15:03:45,392 alphafold2_multimer_v3_model_5_seed_000 recycle=1 pLD
2025-12-04 15:10:00,480 alphafold2_multimer_v3_model_5_seed_000 recycle=2 pLD
2025-12-04 15:16:15,505 alphafold2_multimer_v3_model_5_seed_000 recycle=3 pLD
2025-12-04 15:16:15,512 alphafold2_multimer_v3_model_5_seed_000 took 1499.3s
```

colored by chain



colored by pLDDT



```
2025-12-04 15:16:17,175 reranking models by 'multimer' metric
2025-12-04 15:16:17,176 rank_001_alphafold2_multimer_v3_model_4_seed_000 pLDD
2025-12-04 15:16:17,176 rank_002_alphafold2_multimer_v3_model_5_seed_000 pLDD
2025-12-04 15:16:17,176 rank_003_alphafold2_multimer_v3_model_1_seed_000 pLDD
2025-12-04 15:16:17,176 rank_004_alphafold2_multimer_v3_model_3_seed_000 pLDD
2025-12-04 15:16:17,176 rank_005_alphafold2_multimer_v3_model_2_seed_000 pLDD
2025-12-04 15:16:21,831 Done
0
```

Display 3D structure

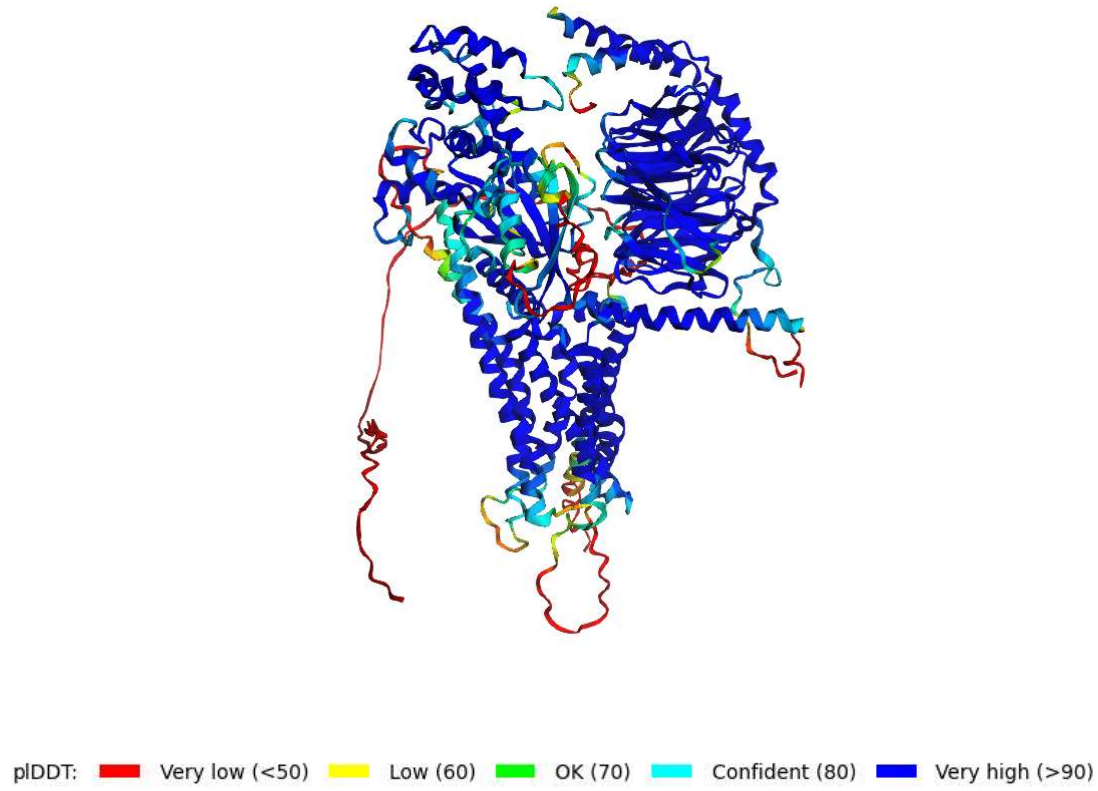
rank_num: 1

color: lDDT

show_sidechains:

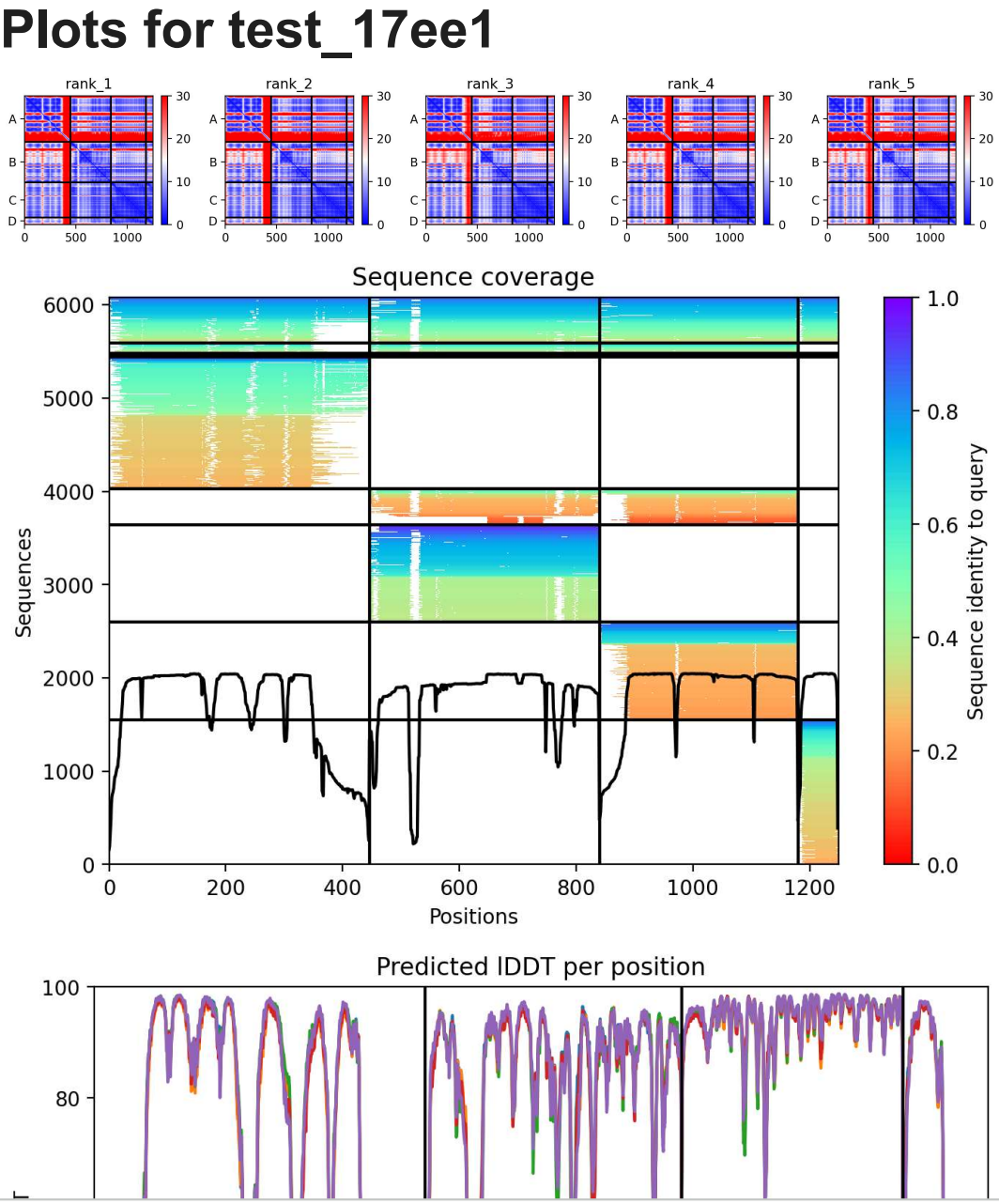
show_mainchains:

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.