▾ AlphaFold Colab

This Colab notebook allows you to easily predict the structure of a protein using a slightly simplified version of AlphaFold v2.2.4.

Differences to AlphaFold v2.2.4

In comparison to AlphaFold v2.2.4, this Colab notebook uses **no templates (homologous structures)** and a selected portion of the BFD database. We have validated these changes on several thousand recent PDB structures. While accuracy will be near-identical to the full AlphaFold system on many targets, a small fraction have a large drop in accuracy due to the smaller MSA and lack of templates. For best reliability, we recommend instead using the full open source AlphaFold, or the AlphaFold Protein Structure Database.

This Colab has a small drop in average accuracy for multimers compared to local AlphaFold installation, for full multimer accuracy it is highly recommended to run AlphaFold locally. Moreover, the AlphaFold-Multimer requires searching for MSA for every unique sequence in the complex, hence it is substantially slower. If your notebook times-out due to slow multimer MSA search, we recommend either using Colab Pro or running AlphaFold locally.

Please note that this Colab notebook is provided as an early-access prototype and is not a finished product. It is provided for theoretical modelling only and caution should be exercised in its use.

The **PAE file format** has been updated to match AFDB. Please see the AFDB FAQ for a description of the new format.

Citing this work

Any publication that discloses findings arising from using this notebook should cite the AlphaFold paper.

Licenses

This Colab uses the AlphaFold model parameters which are subject to the Creative Commons Attribution 4.0 International (CC BY 4.0) license. The Colab itself is provided under the Apache 2.0 license. See the full license statement below.

More information

You can find more information about how AlphaFold works in the following papers:

- [AlphaFold methods paper](#)
- [AlphaFold predictions of the human proteome paper](#)
- [AlphaFold-Multimer paper](#)

FAQ on how to interpret AlphaFold predictions are [here](#).

If you have any questions not covered in the FAQ, please contact the AlphaFold team at alphafold@deepmind.com.

Get in touch

We would love to hear your feedback and understand how AlphaFold has been useful in your research. Share your stories with us at alphafold@deepmind.com.

▾ Setup

Start by running the 2 cells below to set up AlphaFold and all required software.

▸ 1. Install third-party software

Please execute this cell by pressing the *Play* button on the left to download and import third-party software in this Colab notebook. (See the [acknowledgements](#) in our readme.)

Note: This installs the software on the Colab notebook in the cloud and not on your computer.

Show code

100%   100/100 [elapsed: 05:20 remaining: 00:00]

▸ 2. Download AlphaFold

Please execute this cell by pressing the *Play* button on the left.

```
100%                                        100/100 [elapsed: 03:32 remaining: 00:00]
Running with Tesla T4 GPU
```

▾ Making a prediction

Please paste the sequence of your protein in the text box below, then run the remaining cells via *Runtime > Run after*. You can also run the cells individually by pressing the *Play* button on the left.

Note that the search against databases and the actual prediction can take some time, from minutes to hours, depending on the length of the protein and what type of GPU you are allocated by Colab (see FAQ below).

▸ 3. Enter the amino acid sequence(s) to fold ⬇

Enter the amino acid sequence(s) to fold:

- If you enter only a single sequence, the monomer model will be used.
- If you enter multiple sequences, the multimer model will be used.

sequence_1: " AIASEFSSLPSYAAFATAQEAYEQAVANGDSEVVLKKLKKSLNVAKSEFDRDAAMQRKLEKMADQAMTQMYKQARSEDKRAKVTSAN "

sequence_2: " Insert text here   "

sequence_3: " Insert text here   "

sequence_4: " Insert text here   "

sequence_5: " Insert text here   "

sequence_6: " Insert text here "

sequence_7: " Insert text here "

sequence_8: " Insert text here "

Show code

```
Using the single-chain model.
```

▸ 4. Search against genetic databases

Once this cell has been executed, you will see statistics about the multiple sequence alignment (MSA) that will be used by AlphaFold. In particular, you'll see how well each residue is covered by similar sequences in the MSA.

Show code

```
/opt/conda/lib/python3.7/site-packages/haiku/_src/data_structures.py:37: FutureWarning: jax.tree_structure is deprecated, and w
    PyTreeDef = type(jax.tree_structure(None))

Getting MSA for sequence 1
Searching manifv: 100%                                               147/147 [elapsed: 24:13 remaining: 00:00]
```

▸ 5. Run AlphaFold and download prediction

Once this cell has been executed, a zip-archive with the obtained prediction will be automatically downloaded to your computer.

In case you are having issues with the relaxation stage, you can disable it below. Warning: This means that the prediction might have distracting small stereochemical violations.

run_relax: ☑

Relaxation is faster with a GPU, but we have found it to be less stable. You may wish to enable GPU for higher performance, but if it doesn't converge we suggested reverting to using without GPU.

**relax_use_gpu:** ☐

Show code

```
/opt/conda/lib/python3.7/site-packages/haiku/_src/data_structures.py:144: FutureWarning: jax.tree_flatten is deprecated, and
    leaves, treedef = jax.tree_flatten(tree)
/opt/conda/lib/python3.7/site-packages/haiku/_src/data_structures.py:145: FutureWarning: jax.tree_unflatten is deprecated, an
    return jax.tree_unflatten(treedef, leaves)
/opt/conda/lib/python3.7/site-packages/haiku/_src/stateful.py:771: FutureWarning: jax.tree_leaves is deprecated, and will be
    if not jax.tree_leaves(in_axes):
/opt/conda/lib/python3.7/site-packages/alphafold/model/mapping.py:50: FutureWarning: jax.tree_flatten is deprecated, and will
    values_tree_def = jax.tree_flatten(values)[1]
/opt/conda/lib/python3.7/site-packages/alphafold/model/mapping.py:54: FutureWarning: jax.tree_unflatten is deprecated, and wi
    return jax.tree_unflatten(values_tree_def, flat_axes)
/opt/conda/lib/python3.7/site-packages/alphafold/model/mapping.py:129: FutureWarning: jax.tree_flatten is deprecated, and wil
    flat_sizes = jax.tree_flatten(in_sizes)[0]
/opt/conda/lib/python3.7/site-packages/haiku/_src/stateful.py:575: FutureWarning: jax.tree_leaves is deprecated, and will be
    length = jax.tree_leaves(xs)[0].shape[0]
```

Model Confidence

- Very low (pLDDT < 50)
- Low (70 > pLDDT > 50)
- Confident (90 > pLDDT > 70)
- Very high (pLDDT > 90)
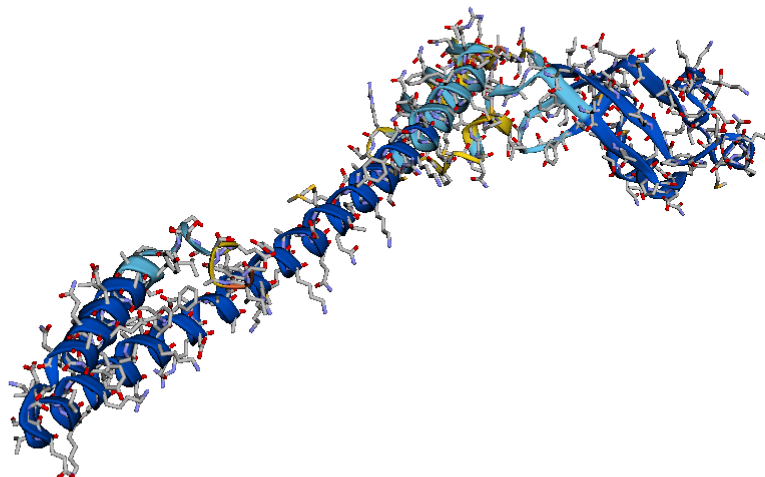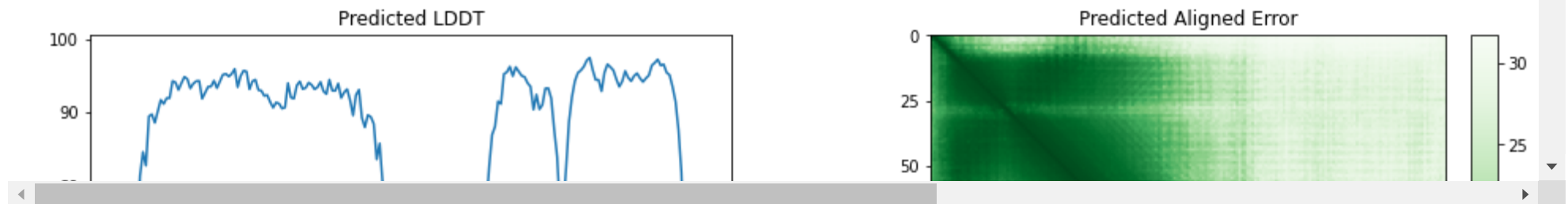
Interpreting the prediction

In general predicted LDDT (pLDDT) is best used for intra-domain confidence, whereas Predicted Aligned Error (PAE) is best used for determining between domain or between chain confidence.

Please see the AlphaFold methods paper, the AlphaFold predictions of the human proteome paper, and the AlphaFold-Multimer paper as well as our FAQ on how to interpret AlphaFold predictions.

FAQ & Troubleshooting

- How do I get a predicted protein structure for my protein?
    - Click on the *Connect* button on the top right to get started.
    - Paste the amino acid sequence of your protein (without any headers) into the "Enter the amino acid sequence to fold".
    - Run all cells in the Colab, either by running them individually (with the play button on the left side) or via *Runtime > Run all.* Make sure you run all 5 cells in order.

- The predicted protein structure will be downloaded once all cells have been executed. Note: This can take minutes to hours - see below.

- How long will this take?

  - Downloading the AlphaFold source code can take up to a few minutes.
  - Downloading and installing the third-party software can take up to a few minutes.
  - The search against genetic databases can take minutes to hours.
  - Running AlphaFold and generating the prediction can take minutes to hours, depending on the length of your protein and on which GPU-type Colab has assigned you.

- My Colab no longer seems to be doing anything, what should I do?

  - Some steps may take minutes to hours to complete.
  - If nothing happens or if you receive an error message, try restarting your Colab runtime via *Runtime > Restart runtime*.
  - If this doesn't help, try resetting your Colab runtime via *Runtime > Factory reset runtime*.

- How does this compare to the open-source version of AlphaFold?

  - This Colab version of AlphaFold searches a selected portion of the BFD dataset and currently doesn't use templates, so its accuracy is reduced in comparison to the full version of AlphaFold that is described in the [AlphaFold paper](#) and [Github repo](#) (the full version is available via the inference script).

- What is a Colab?

  - See the [Colab FAQ](#).

- I received a warning "Notebook requires high RAM", what do I do?

  - The resources allocated to your Colab vary. See the [Colab FAQ](#) for more details.
  - You can execute the Colab nonetheless.

- I received an error "Colab CPU runtime not supported" or "No GPU/TPU found", what do I do?

  - Colab CPU runtime is not supported. Try changing your runtime via *Runtime > Change runtime type > Hardware accelerator > GPU*.
  - The type of GPU allocated to your Colab varies. See the [Colab FAQ](#) for more details.

- If you receive "Cannot connect to GPU backend", you can try again later to see if Colab allocates you a GPU.
  - [Colab Pro](#) offers priority access to GPUs.
- I received an error "ModuleNotFoundError: No module named ...", even though I ran the cell that imports it, what do I do?
  - Colab notebooks on the free tier time out after a certain amount of time. See the [Colab FAQ](#). Try rerunning the whole notebook from the beginning.
- Does this tool install anything on my computer?
  - No, everything happens in the cloud on Google Colab.
  - At the end of the Colab execution a zip-archive with the obtained prediction will be automatically downloaded to your computer.
- How should I share feedback and bug reports?
  - Please share any feedback and bug reports as an [issue](#) on Github.

Related work

Take a look at these Colab notebooks provided by the community (please note that these notebooks may vary from our validated AlphaFold system and we cannot guarantee their accuracy):

- The [ColabFold AlphaFold2 notebook](#) by Sergey Ovchinnikov, Milot Mirdita and Martin Steinegger, which uses an API hosted at the Södinglab based on the MMseqs2 server ([Mirdita et al. 2019, Bioinformatics](#)) for the multiple sequence alignment creation.

License and Disclaimer

This is not an officially-supported Google product.

This Colab notebook and other information provided is for theoretical modelling only, caution should be exercised in its use. It is provided 'as-is' without any warranty of any kind, whether expressed or implied. Information is not intended to be a substitute for professional medical advice, diagnosis, or treatment, and does not constitute medical or other professional advice.

AlphaFold Code License

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at https://www.apache.org/licenses/LICENSE-2.0.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Model Parameters License

The AlphaFold parameters are made available under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) license. You can find details at: https://creativecommons.org/licenses/by/4.0/legalcode

Third-party software

Use of the third-party software, libraries or code referred to in the Acknowledgements section in the AlphaFold README may be governed by separate terms and conditions or license provisions. Your use of the third-party software, libraries or code is subject to any such terms and you should check that you can comply with any applicable restrictions or terms and conditions before use.

Mirrored Databases

The following databases have been mirrored by DeepMind, and are available with reference to the following:

- UniProt: v2021_03 (unmodified), by The UniProt Consortium, available under a Creative Commons Attribution-NoDerivatives 4.0 International License.
- UniRef90: v2021_03 (unmodified), by The UniProt Consortium, available under a Creative Commons Attribution-NoDerivatives 4.0 International License.
- MGnify: v2019_05 (unmodified), by Mitchell AL et al., available free of all copyright restrictions and made fully and freely available for both non-commercial and commercial use under CC0 1.0 Universal (CC0 1.0) Public Domain Dedication.

- BFD: (modified), by Steinegger M. and Söding J., modified by DeepMind, available under a [Creative Commons Attribution-ShareAlike 4.0 International License](#). See the Methods section of the [AlphaFold proteome paper](#) for details.

✓ 22m 42s   completed at 13:56