UT Southwestern Medical Center
Lyda Hill Department of Bioinformatics
BioHPC

Alphafold – Multimer and Astrocyte workflow

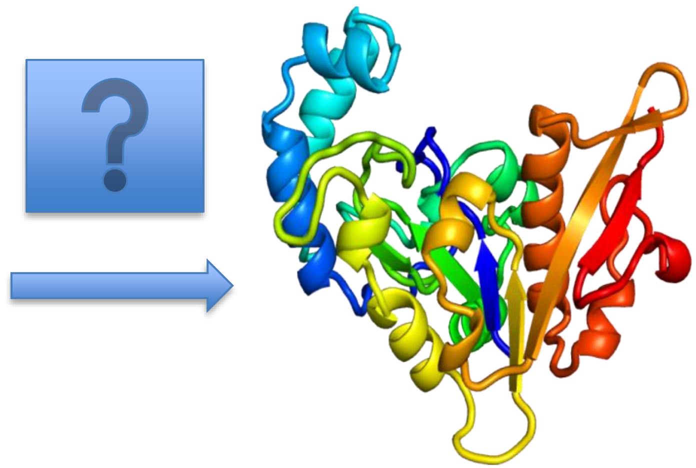
[web] portal.biohpc.swmed.edu
[email] biohpc-help@utsouthwestern.edu

2022-03-23

Outline

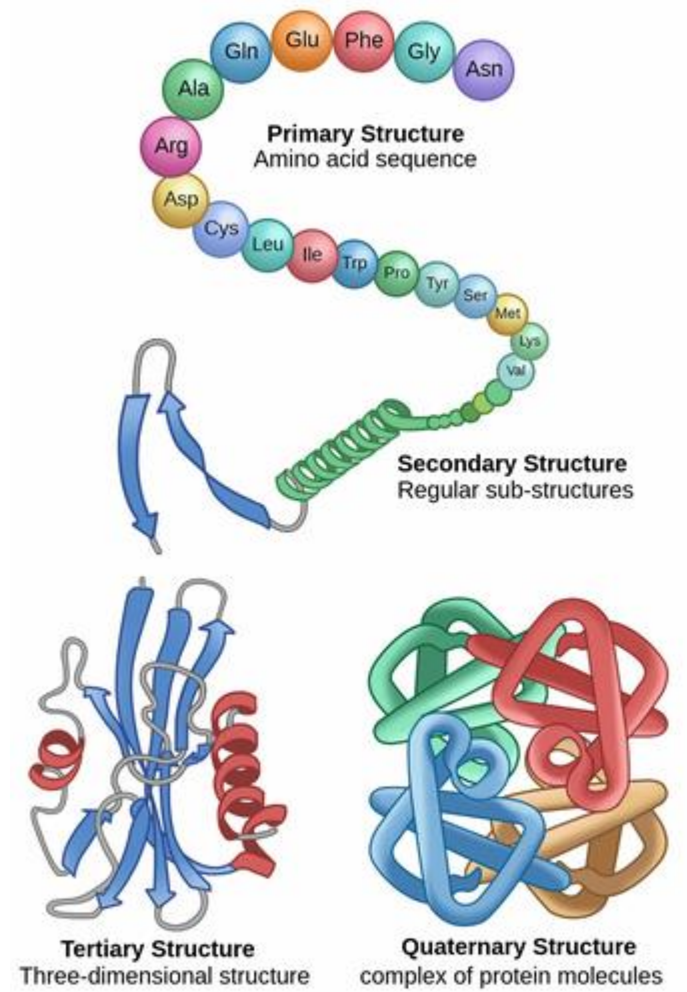
- Protein structure
- CASP, the Critical Assessment of protein Structure Prediction
- Alphafold source code on git hub
- Alphafold database
- Alphafold new feature: Multimer
- Demo: How to run Alphafold using command line on BioHPC
- Demo: How to run Alphafold with Astrocyte GUI on BioHPC

>sp|Q07792|ESTE_VIBMI Arylesterase OS=Vibrio mimicus OX=674 PE=1 SV=1
MIRLLSLVLFFCLSAASQASEKLLVLGDSLSAGYQMPIEK
SWPSLLPDALLEHGQDVTVINGSISGDTTGNGLARLP
QLLDQHTPDLVLIELGANDGLRGFPPKVITSNLSKMISL
IKDSGANVVMMQIRVPPNYGKRYSDMFYDIYPKLAE
HQQVQLMPFFLEHVITKPEWMMDDGLHPKPEAQP
WIAEFVAQELVKHL



Protein structure

- Protein structure
- Protein folding problem
    - Sequence of amino acids do not show how they fold into shape
- Why is protein folding important?
    - Protein structure dictates its function
    - Scientists can develop drugs based on known protein structures
- Methods:
    - X-ray crystallography
    - NMR (Nuclear Magnetic Resonance)
    - CryoEM
    - AI (Artificial Intelligence)

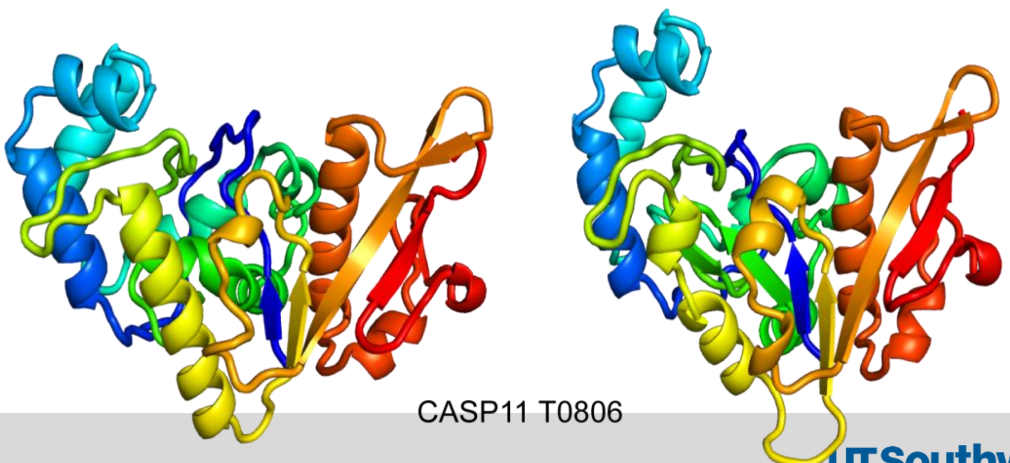


Ref to https://deepmind.com/blog/article/AlphaFold-Using-AI-for-scientific-discovery

CASP

- CASP, Critical Assessment of protein Structure Prediction
  https://predictioncenter.org/
- Community-wide, worldwide experiment for protein structure prediction
- Take place every two years since 1994
- Problem:
  - Target proteins: structures just been solved and hold by Protein Data Bank
  - Neither predictors nor the organizers and assessors know the structures of the target proteins
  - Evaluation: GDT-TS (global Distance Test – Total Score) describing percentage of well-modeled residues in the model with respect to the target

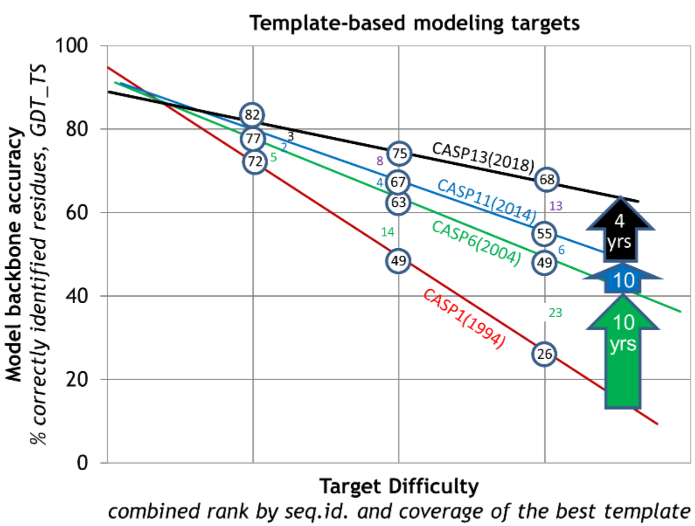
Prediction | Crystal structure

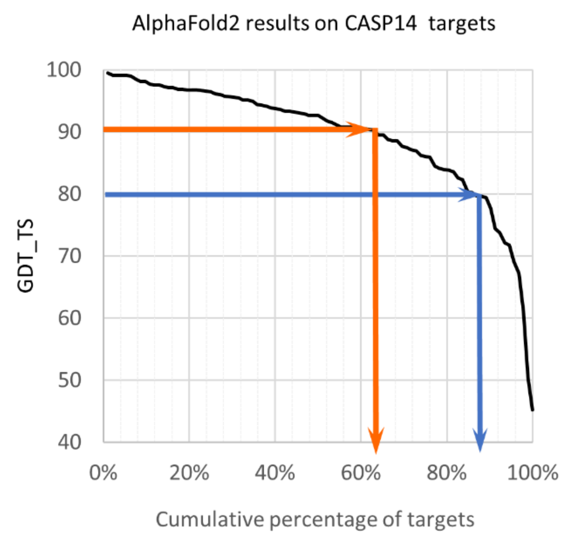


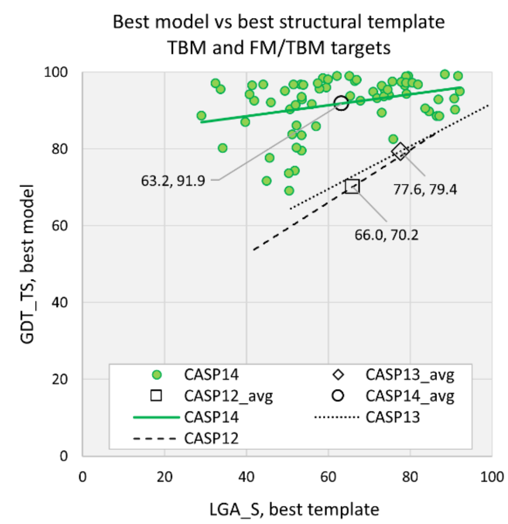
CASP11 T0806

Alphafold in CASP

- Over the course of CASP, template-based modeling get enormous improvement
- The 2014-2018 model accuracy improvement doubled that of 2004-2014
- CASP14 marked an extraordinary increase in the accuracy of the computed three-dimensional protein structures with the emergence of the advanced deep learning method AlphaFold2
- AlphaFold2 proved to be competitive with the experimental accuracy
- The accuracy of CASP14 models is significantly higher than the corresponding average of previous two CASPs







Alphafold Paper

Article

Highly accurate protein structure prediction with AlphaFold

https://doi.org/10.1038/s41586-021-03819-2

Received: 11 May 2021

Accepted: 12 July 2021

Published online: 15 July 2021

Open access

Check for updates

John Jumper[1,4✉], Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], Rishub Jain[1,4], Jonas Adler[1], Trevor Back[1], Stig Petersen[1], David Reiman[1], Ellen Clancy[1], Michal Zielinski[1], Martin Steinegger[2,3], Michalina Pacholska[1], Tamas Berghammer[1], Sebastian Bodenstein[1], David Silver[1], Oriol Vinyals[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Pushmeet Kohli[1] & Demis Hassabis[1,4✉]

Proteins are essential to life, and understanding their structure can facilitate a mechanistic understanding of their function. Through an enormous experimental effort[1–4], the structures of around 100,000 unique proteins have been determined[5], but this represents a small fraction of the billions of known protein sequences[6,7]. Structural coverage is bottlenecked by the months to years of painstaking effort required to determine a single protein structure. Accurate computational approaches are needed to address this gap and to enable large-scale structural bioinformatics. Predicting the three-dimensional structure that a protein will adopt based solely on its amino acid sequence—the structure prediction component of the 'protein folding problem'[8]—has been an important open research problem for more than 50 years[9]. Despite recent progress[10–14], existing methods fall far short of atomic accuracy, especially when no homologous structure is available. Here we provide the first computational method that can regularly predict protein structures with atomic accuracy even in cases in which no similar structure is known. We validated an entirely redesigned version of our neural network-based model, AlphaFold, in the challenging 14th Critical Assessment of protein Structure Prediction (CASP14)[15], demonstrating accuracy competitive with experimental structures in a majority of cases and greatly outperforming other methods. Underpinning the latest version of AlphaFold is a novel machine learning approach that incorporates physical and biological knowledge about protein structure, leveraging multi-sequence alignments, into the design of the deep learning algorithm.

Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." *Nature* 596.7873 (2021): 583-589.

Alphafold Multimer Paper

DeepMind

2022-03-10

Protein complex prediction with AlphaFold-Multimer

Richard Evans[1*], Michael O'Neill[1*], Alexander Pritzel[1*], Natasha Antropova[1*], Andrew Senior[1], Tim Green[1], Augustin Žídek[1], Russ Bates[1], Sam Blackwell[1], Jason Yim[1], Olaf Ronneberger[1], Sebastian Bodenstein[1], Michal Zielinski[1], Alex Bridgland[1], Anna Potapenko[1], Andrew Cowie[1], Kathryn Tunyasuvunakool[1], Rishub Jain[1], Ellen Clancy[1], Pushmeet Kohli[1], John Jumper[1*] and Demis Hassabis[1*]

[1]DeepMind, London, UK, [*]These authors contributed equally

While the vast majority of well-structured single protein chains can now be predicted to high accuracy due to the recent AlphaFold [1] model, the prediction of multi-chain protein complexes remains a challenge in many cases. In this work, we demonstrate that an AlphaFold model trained specifically for multimeric inputs of known stoichiometry, which we call AlphaFold-Multimer, significantly increases accuracy of predicted multimeric interfaces over input-adapted single-chain AlphaFold while maintaining high intra-chain accuracy. On a benchmark dataset of 17 heterodimer proteins without templates (introduced in [2]) we achieve at least medium accuracy (DockQ [3] ≥ 0.49) on 13 targets and high accuracy (DockQ ≥ 0.8) on 7 targets, compared to 9 targets of at least medium accuracy and 4 of high accuracy for the previous state of the art system (an AlphaFold-based system from [2]). We also predict structures for a large dataset of 4,446 recent protein complexes, from which we score all non-redundant interfaces with low template identity. For heteromeric interfaces we successfully predict the interface (DockQ ≥ 0.23) in 70% of cases, and produce high accuracy predictions (DockQ ≥ 0.8) in 26% of cases, an improvement of +27 and +14 percentage points over the flexible linker modification of AlphaFold [4] respectively. For homomeric interfaces we successfully predict the interface in 72% of cases, and produce high accuracy predictions in 36% of cases, an improvement of +8 and +7 percentage points respectively.

Protein complex prediction with AlphaFold-Multimer
Richard Evans, et al. BioRxiv (2021)

Alphafold code on github

A) Deepmind https://github.com/deepmind/alphafold

- Docker:
  - Use "root" (administrative account) to run docker
  - On HPC, users are not allowed to use "root" to run docker
- Databases:
  - full database (2.2TB) /project/apps_database
  - reduced database (415GB)
- Running Alphafold: run_docker.py

```
python3 docker/run_docker.py --fasta_paths=T1050.fasta --max_template_date=2020-05-14
```

B) Alphafold non docker https://github.com/kalininalab/alphafold_non_docker

- Conda environment instead of docker
- Download Alphafold git repo
- Databases: full database and reduced database, same as above
- Install as a BioHPC module
- Running Alphafold: run_alphafold.sh calls run_alphafold.py (Deepmind)

```
run_alphafold.sh -d /project/apps_database/alphafold/database_full -o
/your/path/to/dummy_test/ -m model_1 -f /your/path/to/query.fasta -t 2020-05-14
```

Alphafold database

- https://alphafold.ebi.ac.uk/
- Collaboration between Google Deepmind and EMBL-EBI
- Human proteome (98.5%): 58% of residues with a confident prediction, of which a subset (36% of all residues) have very high confidence
- Cover over 100 million proteins in UniRef90

Article

Highly accurate protein structure prediction for the human proteome

https://doi.org/10.1038/s41586-021-03828-1
Received: 11 May 2021
Accepted: 16 July 2021
Published online: 22 July 2021
Open access
Check for updates

Kathryn Tunyasuvunakool[1]✉, Jonas Adler[1], Zachary Wu[1], Tim Green[1], Michal Zielinski[1], Augustin Žídek[1], Alex Bridgland[1], Andrew Cowie[1], Clemens Meyer[1], Agata Laydon[1], Sameer Velankar[2], Gerard J. Kleywegt[2], Alex Bateman[2], Richard Evans[1], Alexander Pritzel[1], Michael Figurnov[1], Olaf Ronneberger[1], Russ Bates[1], Simon A. A. Kohl[1], Anna Potapenko[1], Andrew J. Ballard[1], Bernardino Romera-Paredes[1], Stanislav Nikolov[1], Rishub Jain[1], Ellen Clancy[1], David Reiman[1], Stig Petersen[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Ewan Birney[2], Pushmeet Kohli[1], John Jumper[1,3]✉ & Demis Hassabis[1,3]✉

Protein structures can provide invaluable information, both for reasoning about biological processes and for enabling interventions such as structure-based drug development or targeted mutagenesis. After decades of effort, 17% of the total residues in human protein sequences are covered by an experimentally determined structure[1]. Here we markedly expand the structural coverage of the proteome by applying the state-of-the-art machine learning method, AlphaFold[2], at a scale that covers almost the entire human proteome (98.5% of human proteins). The resulting

Tunyasuvunakool, Kathryn, et al. "Highly accurate protein structure prediction for the human proteome." Nature 596.7873 (2021): 590-596.

How to run Alphafold using command line?

GPU nodes on BioHPC

GPU Partition	Number of CPU/Node	Memory Per Node	Number of GPU/Node	GPU Memory	Number of nodes
GPU	32	256GB	1 K20/K40	6GB/12GB	8
GPUp4	72	384GB	1 P4	8GB	16
GPUp40	72	384GB	1 P40	24GB	16
GPUp100	56	256GB	2 P100	16GB	12
GPUv100s	72	384GB	1 V100S	32GB	32
GPU4v100	72	384GB	4 V100S	32GB	12
GPUA100	72	1.5TB	1 A100	40GB	16

Check node availability

```
[s179389@Nucleus005 ~]$ sinfo -p GPUp4
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
GPUp4        up   infinite      5  alloc NucleusC[002,012-013,016-017]
GPUp4        up   infinite     11   idle NucleusC[003-011,014-015]
```

Run alphafold on BioHPC

- WebGPU session: https://portal.biohpc.swmed.edu/terminal/webgui/
  20 hour time limit, graphical interface, check results using Pymol
  Better for short time job, such as one monomer
- Submit a Slurm job. Better for long time job, such as multimer

For best performance we recommend TurboVNC for webGUI and webGPU sessions, and the NICE DCV client for webWinDCV sessions.

TurboVNC Client Download: [Windows] [Mac OSX] [Linux 64-bit .deb] [Linux 64-bit .rpm] (Version 2.0.91)

NICE DCV Client Download: [Windows] [Mac OSX] [Linux .tar.gz]

Launch a new interactive / visualization job

Note that a session may take time to start if there are no nodes currently free in the cluster. Jobs run for a maximum of 20 hours.

Job type*

WebGPUv100s - Tesla V100 GUI + single GPU, high performance

Your session will start immediately, nodes are available.

Launch Job

Run alphafold

- Example command

```
module load alphafold/2.1.1
run_alphafold -d /project/apps_database/alphafold_2.1.1/database_full \
-o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
-m multimer \
-f /project/biohpcadmin/s179389/alphafold_test/example/multimer.fasta \
-t 2020-05-14
```

Please note:

- Use same version of alphafold module and alphafold database
- Use latest version (2.1.1) in most cases
- Only use alphafold/2.0 when need to reproduce previous results in alphafold/2.0

Usage: /cm/shared/apps/alphafold/2.1.1/alphafold/run_alphafold <OPTIONS>

Required parameters	
-d <data_dir>	Path to directory of supporting data
-o <output_dir>	Path to a directory that will store the results
-f <fasta_path>	Path to a FASTA file containing sequence. If a FASTA file contains multiple sequences, then it will be folded as a multimer
-t <max_template_date>	Maximum template release date to consider (ISO-8601 format - i.e. YYYY-MM-DD). Important if folding historical test sets
Optional parameters	
-g <use_gpu>	Enable NVIDIA runtime to run with GPUs (default: true)
-n <openmm_threads>	OpenMM threads (default: all available cores)
-a <gpu_devices>	Comma separated list of devices to pass to 'CUDA_VISIBLE_DEVICES' (default: 0)
-m <model_preset>	Choose preset model configuration - the monomer model, the monomer model with extra ensembling, monomer model with pTM head, or multimer model (default: 'monomer)
-c <db_preset>	Choose preset MSA database configuration - smaller genetic database config (**reduced_dbs**) or full genetic database config (full_dbs) (default: **'full_dbs**')
-p <use_precomputed_msas>	Whether to read MSAs that have been written to disk. WARNING: This will not check if the sequence, database or configuration have changed (default: 'false')
-l <is_prokaryote>	Optional for multimer system, not used by the single chain system. A boolean specifying true where the target complex is from a prokaryote, and false where it is not, or where the origin is unknown. This value determine the pairing method for the MSA (default: 'None')
-b <benchmark>	Run multiple JAX model evaluations to obtain a timing that excludes the compilation time, which should be more indicative of the time required for inferencing many proteins (default: 'false’)

Usage: /cm/shared/apps/alphafold/2.0/alphafold/run_alphafold <OPTIONS>	
Required parameters	
-d <data_dir>	Path to directory of supporting data
-o <output_dir>	Path to a directory that will store the results
-m <model_names>	Names of models to use (a comma separated list) i.e. model_1,model_2, model3
-f <fasta_path>	Path to a FASTA file containing one sequence
-t <max_template_date>	Maximum template release date to consider (ISO-8601 format - i.e. YYYY-MM-DD). Important if folding historical test sets
Optional parameters	
-n <openmm_threads>	OpenMM threads (default: all available cores)
-b <benchmark>	Run multiple JAX model evaluations to obtain a timing that excludes the compilation time, which should be more indicative of the time required for inferencing many proteins (default: 'False')
-g <use_gpu>	Enable NVIDIA runtime to run with GPUs (default: True)
-a <gpu_devices>	Comma separated list of devices to pass to 'CUDA_VISIBLE_DEVICES'. (default: 0)
-p <preset>	Choose preset model configuration - no ensembling and smaller genetic database config (**reduced_dbs**), no ensembling and full genetic database config (**full_dbs**) or full genetic database config and 8 model ensemblings (casp14)

Run Alphafold – input fasta file

Input FASTA files

- Monomer

>sequence_1
<SEQUENCE>

- Multimer: Homomer

>sequence_1
<SEQUENCE>
>sequence_2
<SEQUENCE>
>sequence_3
<SEQUENCE>

- Multimer: Heteromer

>sequence_1
<SEQUENCE A>
>sequence_2
<SEQUENCE A>
>sequence_3
<SEQUENCE B>
>sequence_4
<SEQUENCE B>
>sequence_5
<SEQUENCE B>

Run Alphafold – Databases

Database Version 2.1.1 (latest):
/project/apps_database/alphafold_2.1.1

Module Alphafold/2.1.1 ⟷ Database version 2.1.1

Database Version 2.0:
/project/apps_database/alphafold

Module Alphafold/2.0 ⟷ Database version 2.0

Database Version 2.X.Y (in the future):
/project/apps_database/alphafold_2.X.Y

Module Alphafold/2.X.Y ⟷ Database version 2.X.Y

Full database ~2.2 TB
Reduced database ~ 615 GB

Run Alphafold - monomer

- Monomer with full database

```
module load alphafold/2.1.1
run_alphafold -d /project/apps_database/alphafold_2.1.1/database_full \
-o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
-m monomer \
-f /project/biohpcadmin/s179389/alphafold_test/example/query_2.fasta \
-t 2020-05-14
```

- Monomer with reduced database

```
module load alphafold/2.1.1
run_alphafold -d /project/apps_database/alphafold_2.1.1/database_full \
-o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
-m monomer \
-f /project/biohpcadmin/s179389/alphafold_test/example/query_3.fasta \
-t 2020-05-14 \
-c reduced_dbs
```

Run Alphafold - multimer

- Multimer with full database

module load alphafold/2.1.1
run_alphafold -d /project/apps_database/alphafold_2.1.1/database_full \
-o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
-m **multimer** \
-f /project/biohpcadmin/s179389/alphafold_test/example/multimer.fasta \
-t 2020-05-14

- Multimer with reduced database

module load alphafold/2.1.1
run_alphafold -d /project/apps_database/alphafold_2.1.1/database_full \
-o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
-m **multimer** \
-f /project/biohpcadmin/s179389/alphafold_test/example/multimer.fasta \
-t 2020-05-14
-c reduced_dbs

Run alphaFold on BioHPC_SLURM Job

submit a SLURM job through: https://portal.biohpc.swmed.edu/sbatch/#/script

```
module load alphafold/2.0

# COMMAND GROUP 1


run_alphafold -d /project/apps_database/alphafold/database_full \
        -o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
        -m model_1,model_2 \
        -f /project/biohpcadmin/s179389/alphafold_test/example/query.fasta \
        -t 2020-05-14

# END OF SCRIPT
```

Run alphaFold on BioHPC_SLURM Job_Multiple sequence prediction

submit a SLURM job through: https://portal.biohpc.swmed.edu/sbatch/#/script
try to limit to 4 jobs per node, as intensive I/O request of the script

```
module load alphafold/2.0

# COMMAND GROUP 1
run_alphafold -d /project/apps_database/alphafold/database_full \
        -o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
        -m model_1,model_2 \
        -f /project/biohpcadmin/s179389/alphafold_test/example/query_2.fasta \
        -t 2020-05-14 &

run_alphafold -d /project/apps_database/alphafold/database_full \
        -o /project/biohpcadmin/s179389/alphafold_test/dummy_test_slrum/ \
        -m model_1,model_2 \
        -f /project/biohpcadmin/s179389/alphafold_test/example/query.fasta \
        -t 2020-05-14 &
wait
# END OF SCRIPT
```

Demo: Astrocyte workflow for Alphafold

Talk to BioHPC

- Actively develop new workflow of Alphafold on BioHPC
- Welcome feedbacks, comments, and thoughts on what feature you want to have of Alphafold on BioHPC
- Contact us biohpc-help@utsouthwestern.edu