

AlphaFold2 on Alpine

RMACC Symposium

22 May 2024

Layla Freeborn



Research Computing
Office of Information and Technology

UNIVERSITY OF COLORADO BOULDER

Slides



https://bit.ly/af_rmacc_repo



Research Computing
Office of Information and Technology

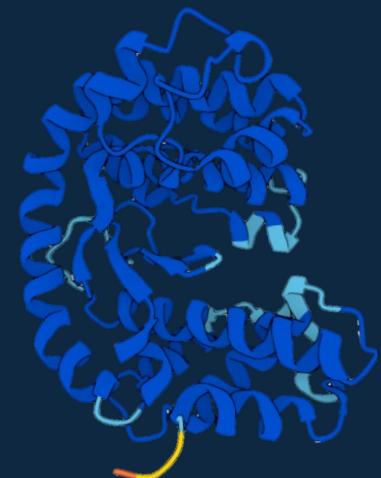
UNIVERSITY OF COLORADO BOULDER

Background: The Protein Folding Problem

- What a protein does (it's biological function) depends on its 3D structure
- Figuring out the what shapes a protein folds into has been a grand challenge in biology for 50+ years

Can we predict a protein's 3D structure based on its 1D aa sequence?

This is computationally **VERY** intensive!



Background

Article | [Open access](#) | Published: 15 July 2021

Highly accurate protein structure prediction with AlphaFold

[John Jumper](#) , [Richard Evans](#), [Alexander Pritzel](#)

[Kathryn Tunyasuvunakool](#), [Russ Bates](#), [Augustin Žídek](#)

[Meyer](#), [Simon A. A. Kohl](#), [Andrew J. Ballard](#), [Andrea](#)

[Nikolov](#), [Rishabh Jain](#), [Jonas Adler](#), [Trevor Back](#), [Stijn](#)

[Zielinski](#), ... [Demis Hassabis](#)  [+ Show authors](#)

[Nature](#) **596**, 583–589 (2021) | [Cite this article](#)

1.56m Accesses | **8815** Citations | **3559** Altmetric | [Metrics](#)

Access & Citations

1.57m

Article Accesses

8815

[Web of Science](#)

1237

[CrossRef](#)

Online attention



5229 tweeters

318 news outlets

17 Video uploaders

104 blogs

7 Redditors

36 Wikipedia page

6 Facebook pages

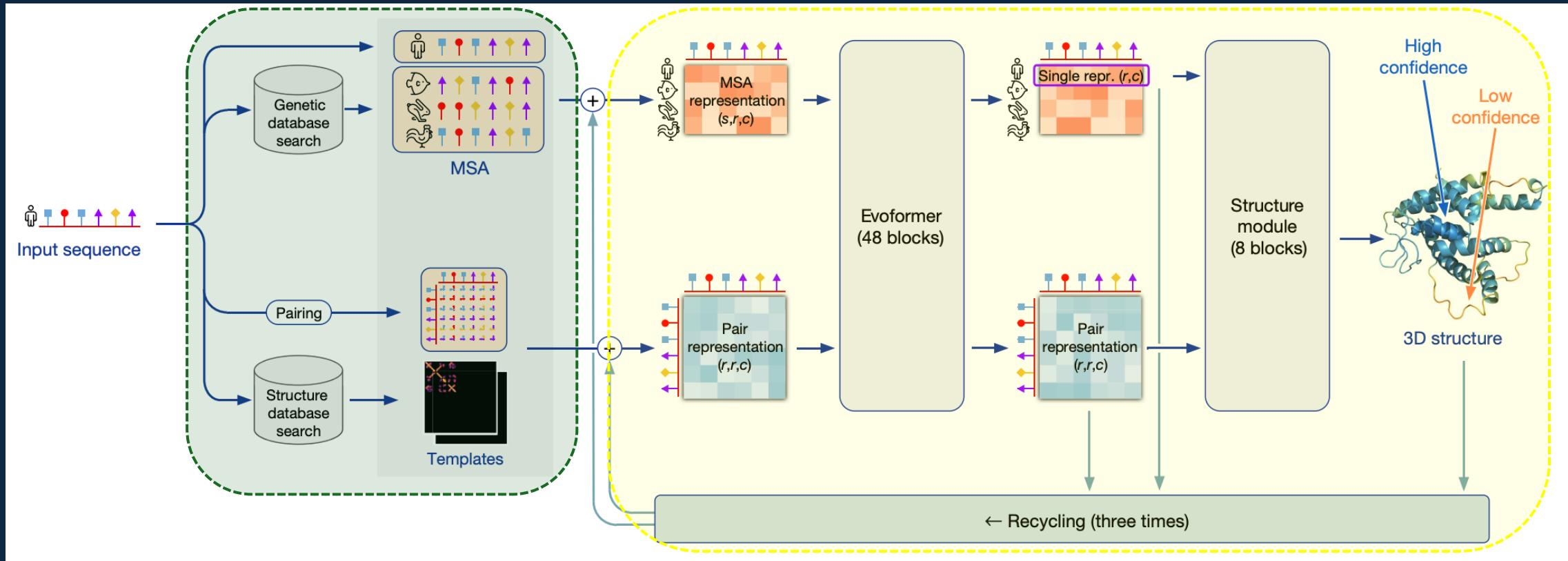
15 F1000

This article is in the 99th percentile (ranked 108th) of the 449,474 tracked articles of a similar age in all journals and the 98th percentile (ranked 15th) of the 927 tracked articles of a similar age in *Nature*

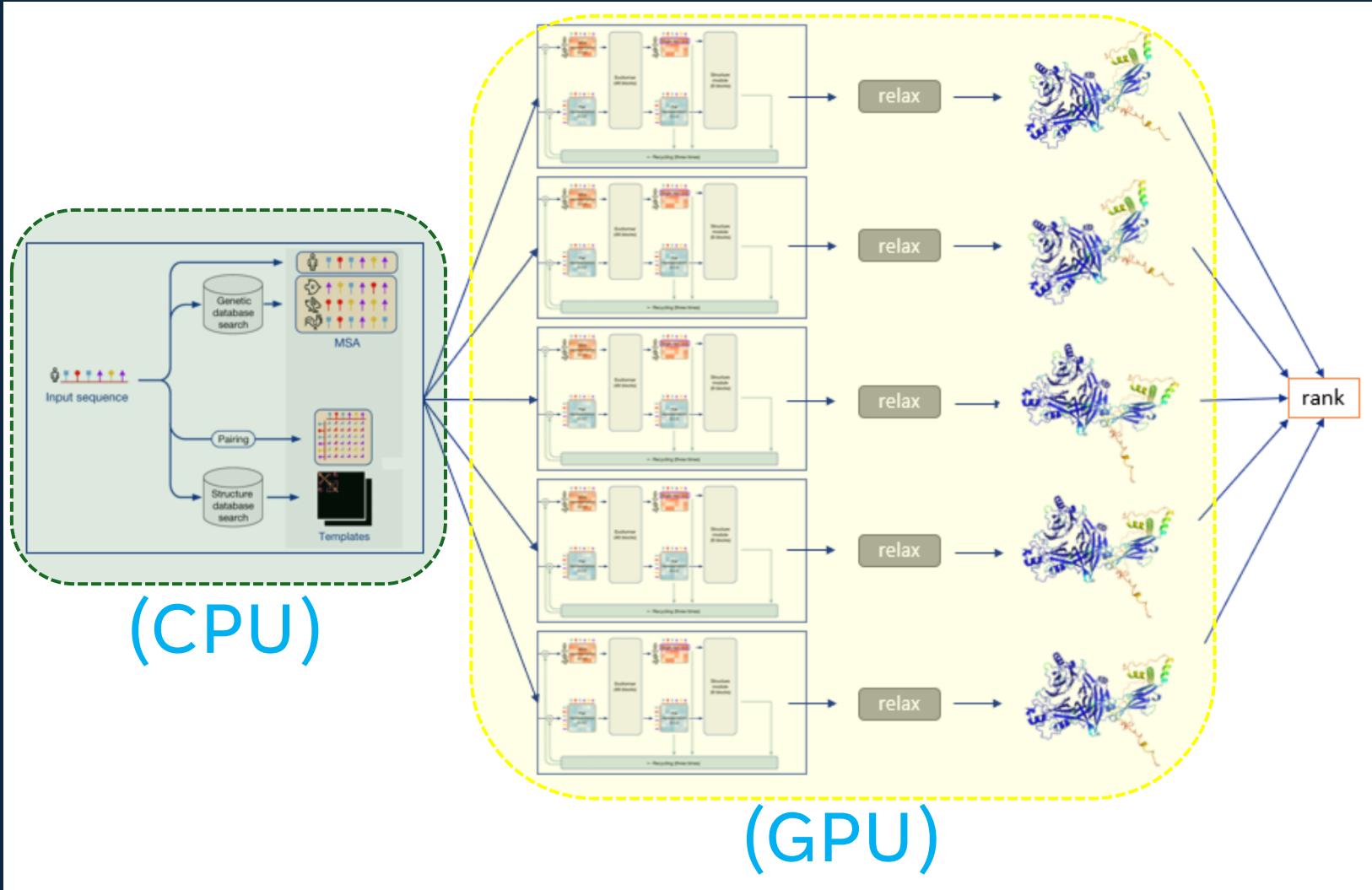
[View more on Altmetric](#)



The Pipeline



The Pipeline



Research Computing
Office of Information and Technology

UNIVERSITY OF COLORADO BOULDER

AlphaFold Inputs

- FASTA file (e.g., `dummy.fasta`, `T1061.fa`)

```
>T1061 T5 phage tail subcomplex, T5 phage, 949 residues|
MKKILDSAKNYLNTDKLKTACLIALELPSSSGSAATYIYLTDYFRDVTYNGILYRSGKVKSISSHKQNRQLSIGSLSFTITGTAEDEV
KLVQNGVSFLDRGITIHQAIIINEEGNILPVDPTDGPLLFFRGRITGGGIKDNDVNTSGIGTSVITWNCSNQFYDFDRVNGRYTDDASHRG
LEVNVNTLQPSNGAKRPEYQEDYGFHSNKSTTILAKYQVKEERYKLQSKKKLFGLRSYSLKYYETVTKEVLDNFNLAAKFIPVVYGV
QKIPGIPIFADTELNNPNIVVVYAFAAEGEIDGFLDFYIGDSPMICFDETDSDTRTCFGRKKIVGDTMHRLAAGTSTSQPSVHGQEYKYN
DGNGDIRIWTFHGKPDQTAAQVLVDIAKKKGFYLNQNQNGNGPEYWDSRYKLLDTAYAIVRFTINENRTEIPEISAEVQGKKVKVYNSDGT
IKADKTSLNGIWQLMDYLTSDRYGADITLDQFPLQKVISEAKILDIDIIDESYQTSWQPYWRYVGWNDPLSENQRQIVQLNTILDSEVFKN
VQGILESFGGAINNLSGEYRITVEKYSTNPLRINFQDTYGDLDLSDTTGRNKFNSVQASLVDPALSWKTSITFYNSKFKEQDKGLDKKL
QLSFANITNYYTARSYADRELKKSRYSRTLSFSVPYKFIGIEPNPIAFTYERYGWKDKFFLVDEVENTRDGKINLVLQEYGEDVFINSE
QVDNSGNDIPDISNNVLPPRDFKYTPPGVVGAIGKNGELSWLPSLTNNVYYSSIAHSGHVNPYIVQQLENPNERMIQEIIIGEPAGLA
IFELRAVDINGRRSSPVTLSVDLNSAKNLSVVSNFRVVNTASGDVTEFVGPDVKLAWDKIPEEEIIPEIYYTLEIYDSQDRMLRSVRIED
VYTYDYLTYNKADFALLNSGALGINRKLFRIRAEGENGEQSVGWATI
```

A FASTA file is the only *required* AlphaFold input.



Alpine AlphaFold2 Module

Loading the AlphaFold module (`ml alphafold`)

- activates a conda environment containing all the necessary packages
- redirects `TMPDIR` to `/scratch/alpine/$USER`
- creates a shortcut to the AlphaFold run script
- sets convenient environment variables

`$CURC_AF_DB`

`$CURC_AF_EXAMPLES`



Alpine AlphaFold2 Module

```
-- This module loads AlphaFold

-- Set local variables
local PACKAGE_PREFIX = "/curc/sw/install/bio/alphafold/2.3.1/2.3.1_env"
local USER = os.getenv("USER")

-- Load dependencies
always_load("anaconda/2022.10")

-- Set paths
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))

-- Set environment variables
setenv("TMPDIR", (os.getenv("SCRATCHDIR")) .. "/" .. (os.getenv("USER")))
setenv("CURC_AF_DBs", "/gpfs/alpine1/datasets/bioinformatics/alphafold")
setenv("CURC_AF_EXAMPLES", "/curc/sw/install/bio/alphafold/examples")

-- Create AlphaFold script shortcut
-- Note this will only work with bash; can add csh using $* (see https://lmod.readthedocs.io/en/latest/015_writing_modules.html#setting-aliases-and-shell-functions)
set_shell_function("run_alphaFold", 'bash /curc/sw/install/bio/alphafold/2.3.1/alphafold-2.3.1/run_alphaFold.sh "$@"')
```

sets environment variables/redirects TMPDIR

creates a shortcut to run_alphaFold.sh



Alpine AlphaFold2 Module

```
-- Activate alphafold conda environment
execute{cmd="conda activate /curc/sw/install/bio/alphafold/2.3.1/2.3.1_en
v",modeA={"load"}}
-- Software info:

help([
AlphaFold:
AlphaFold is an artificial intelligence program developed by DeepMind, a
subsidiary of Alphabet, which performs predictions of protein structure.

For detailed instructions, go to:
https://github.com/deepmind/alphafold
])
whatis("Version: 2.3.1")
whatis("Citing AlphaFold: Jumper, J et al. Highly accurate protein struct
ure prediction with AlphaFold. Nature (2021). Varadi, M et al. AlphaFold
Protein Structure Database: massively expanding the structural coverage o
f protein-sequence space with high-accuracy models. Nucleic Acids Researc
h (2022).")
```

activates the AlphaFold conda
environment



AlphaFold2 Required Parameters

```
(/curc/sw/install/bio/alphafold/2.3.1/2.3.1_env) [lafr9499@c3cpu-a2-u32-3 ~]$ run_alphaFold
```

Please make sure all required parameters are given

Usage: /curc/sw/install/bio/alphafold/2.3.1/alphafold-2.3.1/run_alphaFold.sh <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_paths> Path to FASTA files containing sequences. If a FASTA file contains multiple sequences, then it will be folded as a multimer. To fold more sequences one after another, write the files separated by a comma
- t <max_template_date> Maximum template release date to consider (ISO-8601 format - i.e. YYYY-MM-DD). Important if folding historical test sets

To predict the structure of a protein in the PDB without using its experimental structure as a template, set **max_template_date** (-t) to before the release date of the structure.

TIP: A date like 1900-01-01 will ensure no templates are used.



AlphaFold2 Optional Parameters

Optional Parameters:

```
-g <use_gpu>          Enable NVIDIA runtime to run with GPUs (default: true)
-r <run_relax>         Whether to run the final relaxation step on the predicted models. Turning relax off might result in predictions with distracting stereochemical violations but might help in case you are having issues with the relaxation stage (default: true)
-e <enable_gpu_relax> Run relax on GPU if GPU is enabled (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')
```

Parameter values for --model_preset (-m):

monomer

monomer_casp14

monomer_ptm

multimer



AlphaFold2 Optional Parameters

```
-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 <num_multimer_predictions_per_model> How many predictions (each with a different random seed) will be generated per model. E.g. if this is 2 and there are 5 models then there will be 10 predictions per input. Note: this FLAG only applies if model_preset=multimer (default: 5)
-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')
```

Using precomputed MSAs: AlphaFold looks for the MSAs in the output directory, so they must stay the same if you want to reuse them.



Folding Monomers (1 Polypeptide Chain)

- The default model preset is monomer prediction (`--model_preset=monomer` or `-m "monomer"`)
- Requires a FASTA file with a single sequence

Run Command: `run_alphaFold -d $CURC_AF_DBs -o . -f $CURC_AF_EXAMPLES/dummy.fasta -t 2020-05-14 -m "monomer" -g true`



Folding Multimers (>1 Polypeptide Chain)

- Requires a FASTA file containing multiple sequences

```
>bNACHT09
MKKIFISYSWDGEHQEWVRKLADSLEEDHAYHVVWDGYDLDPLSDKNFFMERAATNTDFILVVATEKYKEADNRNGGVGIETYLAARHWEMM
TGDKKKTNVITINREPDPSTPNLYNGLHYVDFSKDELFNDSLVELKRLNAEPKFKRPEKQIGVTSLKSYNLTKVSDIIGIGAKNRICLINAEGT
DFSAGKKIKYEFWELRNPQGVITHVLALHNNIISQTLNRAEEIKERFQNISTLLLRPREKKGVESMDTILEKKGSRTVNINEYTYDEYIW
EYCIDSDFKNITPPDAIEFYTSQEVDSDKGKIYSSGVNTLSDELLKTSVDYANLIIGGGIGKTSCLTVNKLIKEKSESFLTILRSEDIRKY
IHADAGINSLYVSIDIYDIYEMQAKYLNHNVNFDRNTFLSILSGKIIIVDVGDELSSIFGDKFDLVFLKSLTKLHDELGETRILLTRENNIS
QELISDLNINVYKLLGFKLDNCIKYLNIRFKHNEYKEMITSKIRSQIEKCSLSGEERIIPFFDVISNIYEDNLYSGSEDEFELSQUESTPYPSL
NEVNDHVIHSIFNREKVRHKYNNISLEMIELFKTLNIELGDKWDINEASVLIGVLYENQQKELTDCILKNPLLNVSDSFTTYKYEFLHSYFNSL
SIMEFLTKNDNVENTINLLSKSSKDTPHEKDIYKFLYNKGDALQILTTPSLNKAIKMSLEYEKQDKIKYERAVAAIENIIYLISMFTNGKDEFTG
IIKKIYNSQNQNDKITGLYIKGDLPPINLTGMAISHSKFKHYKRFLKSDFENSKFTYTFQFINCHNENIKNTSFLAADIDRNTCDIGDLDYTFNMLT
KIQDAAEQVSEELAKFLSSFYKGSSFVENKKIYIKFSNRLPGLKSGAFNKLSSHGLISVKAEKEVDTFYIISENYRKSVRRFLNDGYKDAKIKD
LIEFIKA
>T4_nrdC.3
MKTRKHYIDYFDSSLITKHDYQKGHREVINNIIRDFLDYIGWENHICKDTQNAYSHLSLLEWFKRSRLLSSVIAVNNVKKFMYPYIETNVSN
DNVVTFNIIINDVKRTYLEEWFSKDSKEFASEFSHEFNNNNVNLFKHSRRLFCHGDDRTINVVKDWVTAKFIPSSQNGPFELLIIVCAPHEIYK
NLPYMKPCANEKHKTIRSLTYKLRTLLSKMDVVESFDDNTNYGLSLFETKVVIKLKDPNKFKPTPKPNHGNDTMKEEREYLSTRLIEVEKQIEE
HTKVLKDLTAKANGLRNAIEVLK
```

Run Command: `run_alphaFold -d $CURC_AF_DBs -o . -f $CURC_AF_EXAMPLES/multimer.fa -t 2020-05-14 -m “multimer” -g true`



Folding Multimers (>1 Polypeptide Chain)

```
I0521 14:26:53.124762 140296744294208 run_alphaFold.py:386] Have 25 models: ['model_1_multimer_v3_pred_0', 'model_1_multimer_v3_pred_1', 'model_1_multimer_v3_pred_2', 'model_1_multimer_v3_pred_3', 'model_1_multimer_v3_pred_4', 'model_2_multimer_v3_pred_0', 'model_2_multimer_v3_pred_1', 'model_2_multimer_v3_pred_2', 'model_2_multimer_v3_pred_3', 'model_2_multimer_v3_pred_4', 'model_3_multimer_v3_pred_0', 'model_3_multimer_v3_pred_1', 'model_3_multimer_v3_pred_2', 'model_3_multimer_v3_pred_3', 'model_3_multimer_v3_pred_4', 'model_4_multimer_v3_pred_0', 'model_4_multimer_v3_pred_1', 'model_4_multimer_v3_pred_2', 'model_4_multimer_v3_pred_3', 'model_4_multimer_v3_pred_4', 'model_5_multimer_v3_pred_0', 'model_5_multimer_v3_pred_1', 'model_5_multimer_v3_pred_2', 'model_5_multimer_v3_pred_3', 'model_5_multimer_v3_pred_4']
I0521 14:26:53.124969 140296744294208 run_alphaFold.py:403] Using random seed 21484416166174026
for the data pipeline
I0521 14:26:53.125209 140296744294208 run_alphaFold.py:161] Predicting multimer
I0521 14:26:53.139846 140296744294208 pipeline_multimer.py:210] Running monomer pipeline on chain A: bNACHT09
I0521 14:26:53.141395 140296744294208 jackhmmer.py:133] Launching subprocess "/curc/sw/install/bio/alphaFold/2.3.1/2.3.1_env/bin/jackhmmer -o /dev/null -A /scratch/alpine/lafr9499/tmpfqonmyqd/output.sto --noali --F1 0.0005 --F2 5e-05 --F3 5e-07 --incE 0.0001 -E 0.0001 --cpu 8 -N 1 /scratch/alpine/lafr9499/tmpic840x7q.fasta /gpfs/alpine1/datasets/bioinformatics/alphaFold/uniref90/uniref90.fasta"
```



Example Job Script

- An example job script is located in
`/curc/sw/install/bio/alphafold/2.3.1`
- Copy to any space you have write permissions and make your desired changes:

```
cd /projects/$USER
```

```
cp /curc/sw/install/bio/alphafold/2.3.1/alphafold_alphine.sh .
```



Example Job Script

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --time=06:00:00
#SBATCH --partition=aa100
#SBATCH --qos=normal
#SBATCH --gres=gpu:1
#SBATCH --job-name=monomer_test
#SBATCH --output=monomer_test_%j.out
#SBATCH --ntasks=8
#SBATCH --mail-type=ALL
#SBATCH --mail-user=<your email address>

module purge
module load alphafold

#change directory
cd /projects/$USER

#run AlphaFold
run_alphaFold -d $CURC_AF_DBs -o . -f $CURC_AF_EXAMPLES/dummy.fasta -t 2020-05-14 -m "monomer" -g true
```



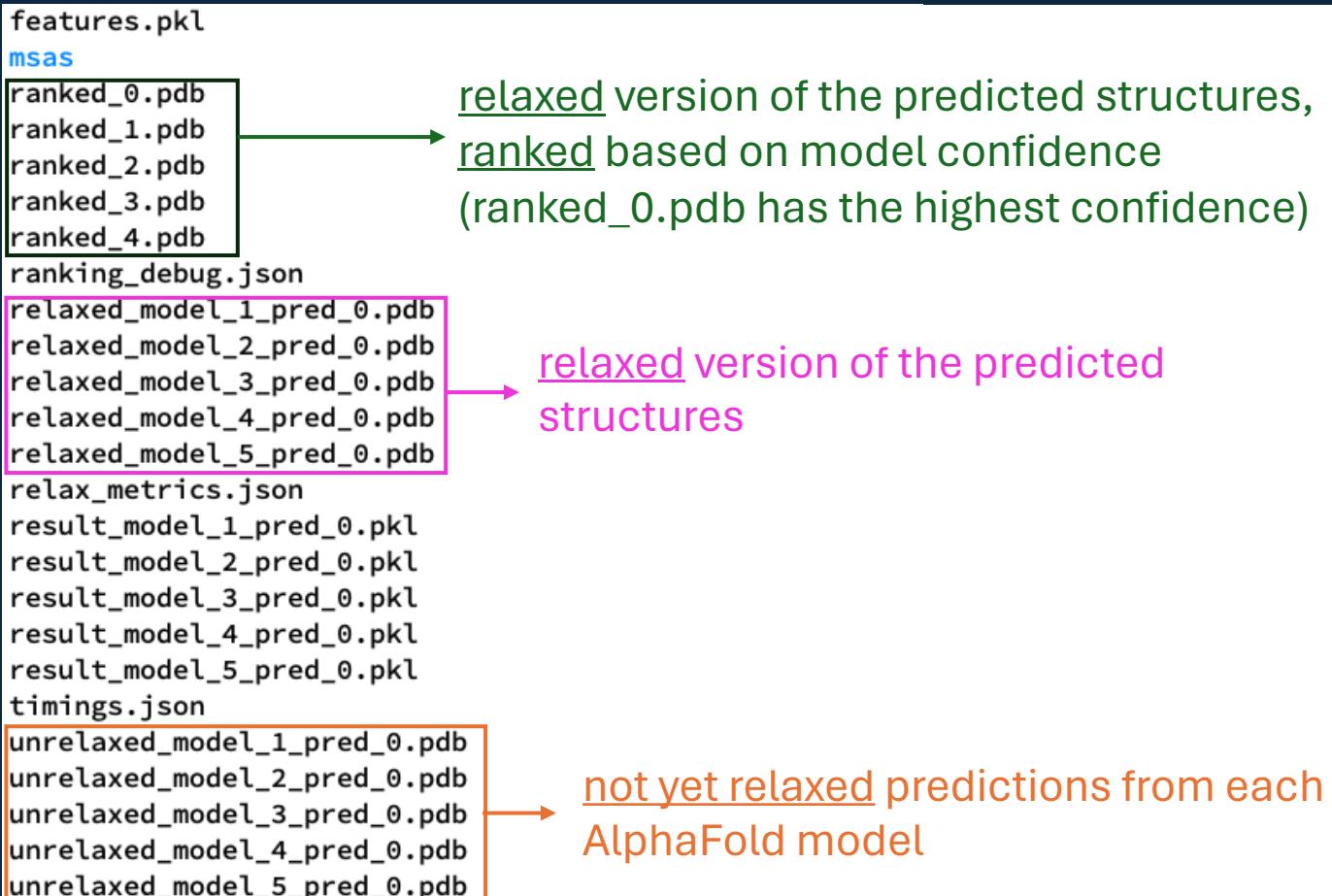
AlphaFold2 Outputs

- Slurm .out file (e.g., `rmacc_dummy_6078420.out`)
 - progress
 - warnings or errors
- a directory with the FASTA name (e.g., `dummy`)
 - computed MSAs
 - unrelaxed, relaxed, and ranked structures
 - raw model outputs
 - prediction metadata
 - timing information



AlphaFold2 Outputs

.pdb
protein
database
format
(human
readable)



AlphaFold2 Outputs

.pkl
pickle data
format
(binary; not
readable)

```
features.pkl
msas
ranked_0.pdb
ranked_1.pdb
ranked_2.pdb
ranked_3.pdb
ranked_4.pdb
ranking_debug.json
relaxed_model_1_pred_0.pdb
relaxed_model_2_pred_0.pdb
relaxed_model_3_pred_0.pdb
relaxed_model_4_pred_0.pdb
relaxed_model_5_pred_0.pdb
relax_metrics.json
result_model_1_pred_0.pkl
result_model_2_pred_0.pkl
result_model_3_pred_0.pkl
result_model_4_pred_0.pkl
result_model_5_pred_0.pkl
timings.json
unrelaxed_model_1_pred_0.pdb
unrelaxed_model_2_pred_0.pdb
unrelaxed_model_3_pred_0.pdb
unrelaxed_model_4_pred_0.pdb
unrelaxed_model_5_pred_0.pdb
```

processed MSA information (NumPy arrays), used as input to produce structures

outputs in .pkl format. includes pLDDT and PTM values



AlphaFold2 Outputs

.json
(readable
data
format)

```
features.pkl
msas
ranked_0.pdb
ranked_1.pdb
ranked_2.pdb
ranked_3.pdb
ranked_4.pdb
ranking_debug.json
relaxed_model_1_pred_0.pdb
relaxed_model_2_pred_0.pdb
relaxed_model_3_pred_0.pdb
relaxed_model_4_pred_0.pdb
relaxed_model_5_pred_0.pdb
relax_metrics.json
result_model_1_pred_0.pkl
result_model_2_pred_0.pkl
result_model_3_pred_0.pkl
result_model_4_pred_0.pkl
result_model_5_pred_0.pkl
timings.json
unrelaxed_model_1_pred_0.pdb
unrelaxed_model_2_pred_0.pdb
unrelaxed_model_3_pred_0.pdb
unrelaxed_model_4_pred_0.pdb
unrelaxed_model_5_pred_0.pdb
```

info on the pLDDT (confidence score) of each model and how they were ranked

relax metrics, including remaining violations

info on how long each section of the pipeline took to run



AlphaFold2 Outputs

Files in the
msas
directory
are in
various file
formats, all
human
readable

```
features.pkl
msas
ranked_0.pdb
ranked_1.pdb
ranked_2.pdb
ranked_3.pdb
ranked_4.pdb
ranking_debug.json
relaxed_model_1_pred_0.pdb
relaxed_model_2_pred_0.pdb
relaxed_model_3_pred_0.pdb
relaxed_model_4_pred_0.pdb
relaxed_model_5_pred_0.pdb
relax_metrics.json
result_model_1_pred_0.pkl
result_model_2_pred_0.pkl
result_model_3_pred_0.pkl
result_model_4_pred_0.pkl
result_model_5_pred_0.pkl
timings.json
unrelaxed_model_1_pred_0.pdb
unrelaxed_model_2_pred_0.pdb
unrelaxed_model_3_pred_0.pdb
unrelaxed_model_4_pred_0.pdb
unrelaxed_model_5_pred_0.pdb
```

A directory containing the files describing
the various genetic tool hits that were
used to construct the input MSA



Visualizing Outputs

- To date we have left this up to the user
- Some options are
 - AlphaPickle: <https://github.com/mattarnoldbio/alphapickle>
 - PyMol (free for education): <https://pymol.org/>
- AlphaFold Protein Structure Database website provides a **great** resource for learning how to interpret visualizations:
<https://alphafold.ebi.ac.uk/entry/Q9Y223#help>



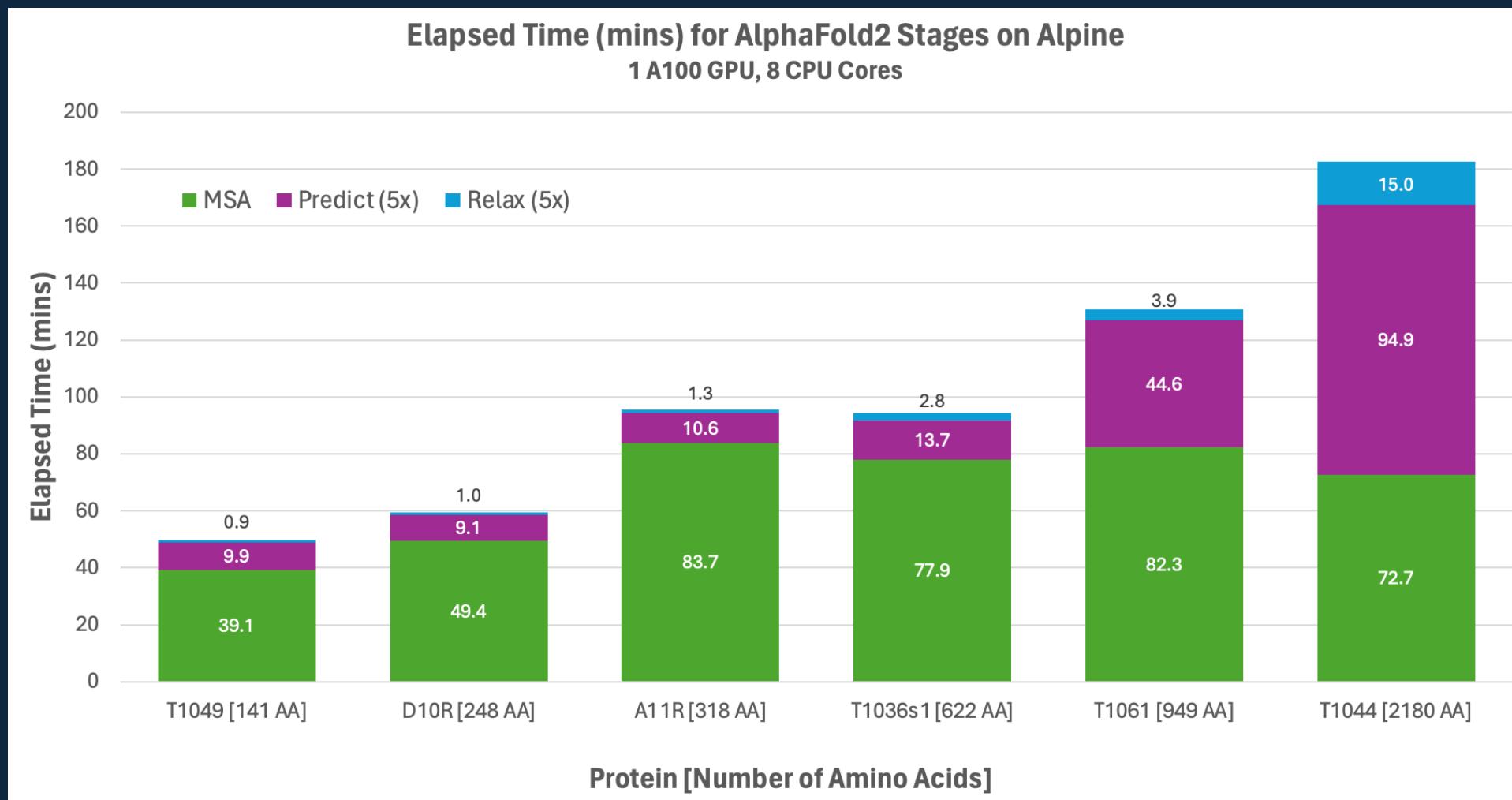
AlphaFold2 Performance Considerations

```
ice set to true (defaults to false) to enable this.  
I0521 13:41:11.550761 140716613289792 run_alphafold.py:386] Have 5 models: ['model_1_pred_0', 'model_2_pred_0', 'mode  
l_3_pred_0', 'model_4_pred_0', 'model_5_pred_0']  
I0521 13:41:11.550974 140716613289792 run_alphafold.py:403] Using random seed 262200831417486581 for the data pipelin  
e  
I0521 13:41:11.551202 140716613289792 run_alphafold.py:161] Predicting dummy  
I0521 13:41:11.567026 140716613289792 jackhmmer.py:133] Launching subprocess "/curc/sw/install/bio/alphafold/2.3.1/2.  
3.1_env/bin/jackhmmer -o /dev/null -A /scratch/alpine/lafr9499/tmpuiv6sw6x/output.sto --noali --F1 0.0005 --F2 5e-05  
--F3 5e-07 --incE 0.0001 -E 0.0001 --cpu 8 -N 1 /curc/sw/install/bio/alphafold/examples/dummy.fasta /gpfs/alpine1/dat  
assets/bioinformatics/alphafold/uniref90/uniref90.fasta"  
I0521 13:41:11.584823 140716613289792 utils.py:36] Started Jackhmmer (uniref90.fasta) query
```

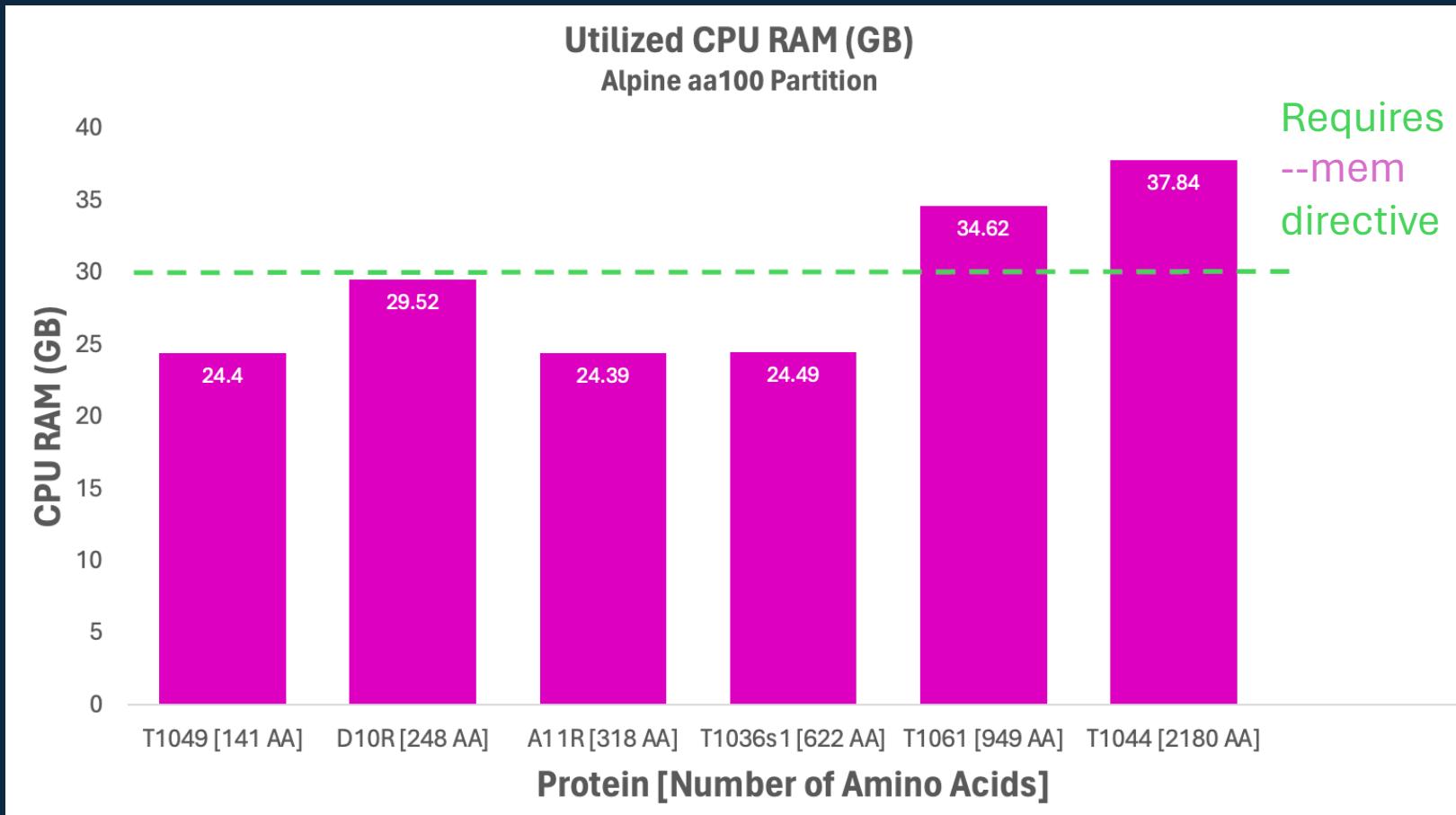
The number of cores/threads used during MSA steps is hardcoded and so AF cannot use more than 8 CPU cores. Requesting more cores will **not** improve performance and you may end up waiting longer for your job to start.



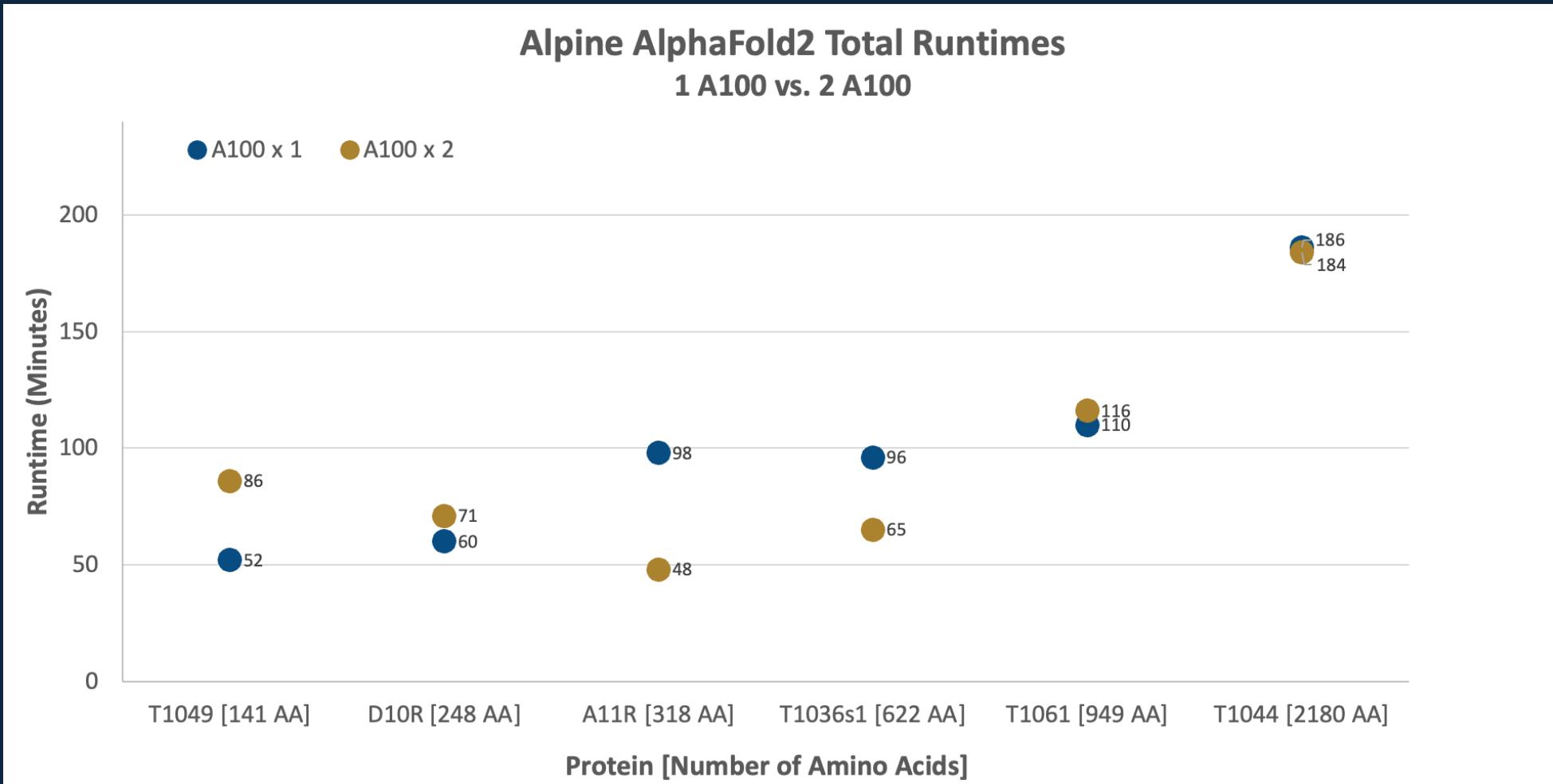
AlphaFold2 Performance Considerations



AlphaFold2 Performance Considerations



AlphaFold2 Performance Considerations



AlphaFold Spin-Offs: ParaFold (ParallelFold)

- Divide CPU-intensive sections from the GPU-intensive sections
- PF isn't currently available as a module, but example scripts are available in `/curc/sw/install/bio/parafold/2.0`
 - `parafold_cpu.sh`
 - `parafold_gpu.sh`



AlphaFold Spin-Offs: ParaFold (ParallelFold)

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --time=10:00:00
#SBATCH --partition=amilan
#SBATCH --qos=normal
#SBATCH --job-name=parafold_cpu
#SBATCH --output=parafold_cpu_%j.out
#SBATCH --ntasks=8
#SBATCH --ntasks-per-node=8
#SBATCH --mem=60G
#SBATCH --mail-type=ALL
#SBATCH --mail-user=<your_email_address>

module purge
module load anaconda

conda activate /curc/sw/install/bio/parafold/2.0/2.0_env

export TMPDIR=/scratch/alpine/$USER

bash /curc/sw/install/bio/parafold/2.0/ParallelFold/run_alphaFold.sh -d /gpfs/alpine1/datasets/bioinformatics/alphafold
-o /projects/$USER/pf_runs -p multimer -i /curc/sw/install/bio/alphaFold/examples/multimer.fa -f -g false
```

-f tells PF to stop after generating the
feature.pkl file



AlphaFold Spin-Offs: ParaFold (ParallelFold)

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --time=24:00:00
#SBATCH --partition=aa100
#SBATCH --gres=gpu:1
#SBATCH --constraint=gpu80
#SBATCH --qos=normal
#SBATCH --job-name=parafold_gpu
#SBATCH --output=parafold_gpu_%j.out
#SBATCH --ntasks=8
#SBATCH --ntasks-per-node=8
#SBATCH --mail-type=ALL
#SBATCH --mail-user=<your_email_address>

module purge
module load cuda
module load anaconda

conda activate /curc/sw/install/bio/parafold/2.0/2.0_env

export TF_FORCE_UNIFIED_MEMORY=1
export XLA_PYTHON_CLIENT_MEM_FRACTION=4.0
export TMPDIR=/scratch/alpine/$USER

bash /curc/sw/install/bio/parafold/2.0/ParallelFold/run_alphaFold.sh -d /gpfs/alpine1/datasets/bioinformatics/alphafold
-o /projects/$USER/pf_runs -p multimer -i /curc/sw/install/bio/alphafold/examples/multimer.fa -g true -m model_1_multimer,model_2_multimer,model_3_multimer,model_4_multimer,model_5_multimer
```

Not needed for the multimer.fa example,
but note that it is an option.

May need to add --mem directive for
some predictions (relaxation step?)



Explore AlphaFold2 on Alpine

Get a training account at
https://bit.ly/af_rmacc24

These accounts will be locked after the session!



Slides/Self-guided material:
https://bit.ly/af_rmacc_repo



Research Computing
Office of Information and Technology

UNIVERSITY OF COLORADO BOULDER

Thank you!



Research Computing
Office of Information and Technology

UNIVERSITY OF COLORADO **BOULDER**