Current Server IP: 34.85.235.58
Current Username: alphafold_user
Hardware: 8 vCPUs, 100GB boot disk, 3TB attached disk, no GPU now
Price: $196 + $10 + $307.2 = $513 monthly

GPU price each piece on google cloud:

nvidia-tesla-T4:	$255.50
nvidia-tesla-P4:	$438.00
nvidia-tesla-P100:	$1065.80
nvidia-tesla-V100:	$1810.40
nvidia-tesla-K80:	$328.50
nvidia-tesla-A100:	$2141.75

Current performance:

- T1029.fasta 3.5 hours running single prediction (MRIDELVPADPRAVSLYTPYYSQANRRRYLPYALSLYQGSSIEGSRAVEGGAPISFV ATWTVTPLPADMTRCHLQFNNDAELTYEILLPNHEFLEYLIDMLMGYQRMQKTDFP GAFYRRLLGYDS)
- GFP11.fasta: 1:45 hours running single prediction (RDHMVLHEYVNAAGIT)
- GFP11.fasta: 2 hours running three predictions simultaneously
- estimation: with 8 vCPUs, and each prediction would take an average of 110% CPU, so running 7~8 predictions at a time would be the maximum efficiency.
- in last case, 2~3 hours for 7~8 GFP11 level (16 proteins)

Way to login:

- web terminal on google cloud (only for project owner)
- linux ssh command with private key (key uploaded)
  sudo ssh -i alphafold_key alphafold_user@[server ip]
- ssh connection tools like mobaXtern on windows

Initialization after login:

1. sudo mount -o discard,defaults /dev/sdb /mnt/disks/data
2. sudo chmod a+w /mnt/disks/data
3. run df -h to check, seeing 3T disks listed means correctly initialized

Way to run alphafold after initialization:

1. cd alphafold
2. *python3 docker/run_docker.py — fasta_paths=seq_input/[STRUCTURE TO PREDICT].fasta — max_template_date=2020–05–14 — use_gpu=False –data_dir=/mnt/disks/data*
   If want to run at background, use *“nohup [command] &*”, and use “*top*” command to view task progress and other details
3. result will be stored at /tmp/alphafold, use “*cd /tmp/alphafold*” to see details

Ways to get prediction results from server:

1. linux terminal:

scp -r sudo scp -r -i alphafold_key alphafold_user@34.85.235.58:/tmp/alphafold/ ./predictions

Then, all results would be at folder called “predictions”

2. other software tools, haven’t done complete research but must have multiple tools

Python program for generating mutations:

1. cd seq_input
2. Three ways of generation:
    - complete list of mutation: *python3 generate_mutation.py all*
    - single point mutation: *python3 generate_mutation.py [position in integer]*
    - all mutation in interval: *python3 generate_mutation.py [start pos] [end pos]*
3. all generated fasta file will be in the current folder, and to delete them all, use “*rm *.fasta*”

Results description:

features.pkl – A pickle file containing the input feature NumPy arrays used by the models to produce the structures.

unrelaxed_model_*.pdb – A PDB format text file containing the predicted structure, exactly as outputted by the model.

relaxed_model_*.pdb – A PDB format text file containing the predicted structure, after performing an Amber relaxation procedure on the unrelaxed structure prediction (see Jumper et al. 2021, Suppl. Methods 1.8.6 for details).

ranked_*.pdb – A PDB format text file containing the relaxed predicted structures, after reordering by model confidence. Here ranked_0.pdb should contain the prediction with the highest confidence, and ranked_4.pdb the prediction with the lowest confidence. To rank model confidence, we use predicted LDDT (pLDDT) scores (see Jumper et al. 2021, Suppl. Methods 1.9.6 for details).

ranking_debug.json – A JSON format text file containing the pLDDT values used to perform the model ranking, and a mapping back to the original model names.

timings.json – A JSON format text file containing the times taken to run each section of the AlphaFold pipeline.

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

result_model_*.pkl – A pickle file containing a nested dictionary of the various NumPy arrays directly produced by the model. In addition to the output of the structure module, this includes auxiliary outputs such as: