

Running Colabfold on the FASRC cluster.

- Colabfold is a port of Alphafold2 that is much faster than the standard Alphafold and does not need large databases protein sequences. It is available as a google colab notebook which uses hardware provided by Google.
- The free version of Colabfold has limits on computing time, RAM, and length of protein sequence to be predicted.
- Localcolabfold is a flavor of colabfold that you can install locally on your own computer or on a high performance super computer such as FASRC's Cannon cluster.

Using the localcolabfold on the cluster has following advantages.

- Localcolabfold runs the calculations on the GPU (provided by the cluster), however we do not need to download large databases.
- All the files are generated on the cluster in your desired folder of choice. No need to download results from a website.
- No need to keep a web browser tab active unlike in case of a job running on the public colabfold server. Job will run in the cluster and the results are available indefinitely.

Some caveats

- A gpu node is essential. However, colabfold uses only one gpu at a time. Multiple gpu usage is not supported.
- It is unclear how much RAM or how many cores are needed for a colabfold job to run successfully. A colabfold test job for BmGr9 (~450 aa) in tetramer mode with 8 CPUs and 4 GB RAM per CPU finished successfully in about 100 minutes.
- It is a good idea to request 12 hours of runtime and 8 CPUs while submitting a colabfold job. The cluster queue is relatively free, and the job starts running within 10 to 30 minutes.

Installation

- Login to Cannon using terminal on Mac, or Powershell on Windows. Use the following command.

```
ssh user-name@login.rc.fas.harvard.edu
```

- Let's open a **tmux** session using the command.

```
tmux
```

- We are going to use conda/mamba to install localcolabfold in a fresh new environment. If you do not have conda/mamba already setup on the RC cluster, you can do so either by using Miniconda (for conda) or Miniforge (for mamba). I recommend using Miniforge instead of Miniconda.
- After you have successfully set up mamba/conda, upload the file **install_localcolabfold.sh** to your home folder on Cannon or copy its content to a new file with the same name on the cluster. Make the file executable running the following command.

```
chmod +x install_localcolabfold.sh
```

- Run the installation script using the following command.

```
./install_localcolabfold.sh | tee install_localcolabfold.sh
```

- After this script has run successfully, we are ready to test our installation.
- Let's start an interactive session to use a GPU node on the cluster.

```
salloc -p gpu_test -t 0-02:00 -c 4 --mem-per-cpu=4G --gres=gpu:1
```

- Now that we are on a GPU node. Let's first activate the `localcolabfold` environment.

```
conda activate localcolabfold
```

- Additionally, we need to also load the GNU C++ compiler.

```
module load gcc
```

- Now we are ready to run colabfold. Let's try predicting the structure of mouse CDH23 EC1-EC2. The sequence is included in the provided FASTA file `CDH23.fasta`.

```
mkdir CDH23-prediction
```

```
colabfold_batch --amber --templates --num-recycle 5 --use-gpu-relax --  
rank iptm --model-type auto CDH23.fasta ./CDH23-prediction
```

- Now we will predict the structure of the extended handshake complex between mouse CDH23 EC1-EC2 and mouse PCDH15 EC1-EC2. But this time we will submit it as a job on the cluster. But first, let's look the file `CDH23-PCDH15.fasta` which contains the sequences.
- Notice how there are two sequences separated by a colon (😊). When you want to predict a multimer, just concatenate the sequences one after another and separate them by colons. But make sure that the FASTA file has only one header.
- Upload the provided script `colabfold-multimer.sh` to the cluster.

- Open the file on the cluster using vim/nano or your editor of choice. Examine the file. Change the values of the variables such as partitions, time etc if needed.
- Submit the job by running the command:

```
sbatch colabfold-multimer.sh CDH23-PCDH15.fasta
```

- When submitting the job using this script, always use the format

```
sbatch colabfold-multimer.sh <name-of-the-fasta-file>
```

- Similarly, you can submit a regular monomer job using the provided `colabfold-monomer.sh` script.