# AlphaPullDown Tutorial

Sophia Aredas

22 August, 2024

## Contents



## My Example Code

This code can be used as a guide to help you see how I was running things.

There will be a second portion called template code where you will see what needs to be added to make your code run! It is a separate Rmarkdown file.

## Step 0

Acquire proteins of interest from wherever. They must be protein sequences not nucleotide!

```
#in your workdir create a directory/folder where you will run alphapulldown
cd /local/workdir/USER
mkdir alphapulldown

cd /local/workdir/USER/alphapulldown
```

```
mkdir data

cd /local/workdir/USER/alphapulldown/data
#add your files by uploading them to the data folder
```

## Step 1: generate msa with colabfold (much faster than the alphafold2 based alignment pipeline)

```
#put all fasta protein sequences into file
#try not to have overly complicated names the simpler the name the better if you can help it:)
cd /local/workdir/USER/alphapulldown/data

#here i have 3 files that I want to test if there are protein interactions in the data directory/folder

#working in my data directory/folder I want to send all of my .fasta files into one fasta file. You mus

#the cat *.fasta >JD_test.fasta specifically looks for any file ending in a .fasta and will take its in
cat *.fasta > JD_test.fasta


#now we will be following the biohpc commands that they have listed for running alphapulldown on the se

#example code
singularity run --bind /local/local_data/colabfold_cache:/cache --bind $PWD --pwd $PWD /programs/colabf
```

## Step 2: run create_individual_features.py on cbsugpu05-07

```
#here we are still copying and pasting the code from the biohpc

#this takes in protein sequences as input and finds structural templates for each protein
singularity exec --bind /local/local_data/alphafold-2.3.2:/db --bind $PWD --pwd $PWD /programs/alphapull
create_individual_features.py \
  --fasta_paths=JD_test.fasta \
  --data_dir=/db \
  --output_dir=msas1 \
  --use_precomputed_msas=True \
  --max_template_date=2024-01-01 \
  --use_mmseqs2=True \
  --skip_existing=False
```

## Step 3: run run_multimer_jobs.pyon GPU (instructions of using two GPU units are at the end of the page)

```
#create two text file bait.txt,candidates.txt. The bait.txt has one line: bait protein name; the candid

  # in this case the bait.txt is the RsiG.fasta file
  #candidates are the whiG_clade_1 and whiG_clade_2.fasta files

#to create bait.txt file
cat RsiG.fasta > bait.txt

#to create candidates
cat whiG_clade_1.fasta whiG_clade_2.fasta > candidates.txt

#run the command from the current directory, which contains: bait.txt,candidates.txt, msas directory, a
cd /local/workdir/sna49/alphapulldown/data
mkdir outputdir

#this part is straight up from the biohpc so this is copy and paste for the most part
#set environment variable to enable GPU to use system memory, so that large proteins can be processed
export TF_FORCE_UNIFIED_MEMORY=1
export XLA_PYTHON_CLIENT_MEM_FRACTION=4.0

singularity exec --nv --bind /local/local_data/alphafold-2.3.2:/db --bind $PWD --pwd $PWD /programs/alph
run_multimer_jobs.py --mode=pulldown \
--num_cycle=3 \
--num_predictions_per_model=1 \
--output_path=./outputdir \
--data_dir=/db \
--protein_lists=bait.txt,candidates.txt \
--monomer_objects_dir=msas
```

## Step 4: generate a summary table

this step can be run on any server. you just need the outputdir from the previous step run the command from the current directory, which has the outputdir under it

Output is a csv file that you can open in Excel: predictions_with_good_interpae.csv

```
#make sure that during this step you are working inside your output directory folder. Mine is called ou
cd outputdir/
singularity exec --bind $PWD --pwd $PWD /programs/alphapulldown-1.0.4/alpha-analysis_jax_0.4.sif run_ge
```

## Step 5: create a jupyter notebook

```
#again make sure that you are in the correct directory
cd outputdir/
singularity exec --bind $PWD --pwd $PWD /programs/alphapulldown-1.0.4/alphapulldown.sif create_notebook
```

```
#after this step, you should see a file output.ipynb in the directory.

#start jupyter in the outputdir

singularity exec --bind $PWD --pwd $PWD /programs/alphapulldown-1.0.4/alphapulldown.sif jupyter lab --ip

#you should see a URL http://cbsuxxxxx.biohpc.cornell.edu:8017/labtoken=xxxxxxxxxxxxxxxxxxxxx, copy pas
```

## Optional step 6: run with alphapickle

It seems that alphapickle is more informative for displaying clearer publication quality charts to evaluate protein-protein interactions

```
#Run command, output files (png and csv will be in the alphafold2 output directory)
#the output directory needs the ABSOLUTE path not a relative path (absolute starts from the root wherea
python /programs/alphapickle/run_AlphaPickle.py -od /local/workdir/sna49/alphapulldown/data/outputdir/Rs

python /programs/alphapickle/run_AlphaPickle.py -od /local/workdir/sna49/alphapulldown/data/outputdir/Rs

#to extract ptm and iptm scores only
/programs/alphafold-2.3.2/alphafoldPickle.py alphafold2_output_directory
```