

AlphaFold2 (2021/8/19)

Please check <https://ccportal.ims.ac.jp/en/node/2946> for basic information and installation procedure.

Webpage (codes)

<https://github.com/deepmind/alphafold>

Reference

- Nature paper: <https://doi.org/10.1038/s41586-021-03819-2>
- Notes on first RCCS installation: <https://ccportal.ims.ac.jp/en/node/2946>

Aug 23, 2021 update

- installed under /local/apl/lx/alphafold2 directory.
 - Latest code on 2021/8/19 of alphafold (/local/apl/lx/alphafold2/alphafold-20210819).
 - Sample input/script at /local/apl/lx/alphafold2/samples.
- modified version of [run_alphafold_rccs.sh](#)
 - use the latest databases on 2021/8/20 in default configuration
 - support -p reduced_dbs
 - UniRef30* can be used instead of uniclust30 (automatically determined; if there are multiple of UniRef30* directories, job will fail.)
 - -a option to specify alphafold code root
 - your own version of alphafold (parameters etc.) can be called from the script ([modification of alphafold/common/residue_constants.py](#) may be necessary)

Sample script

```
#!/bin/sh
#PBS -l select=1:ncpus=12:mpiprocs=1:ompthreads=12:jobtype=core
#PBS -l walltime=72:00:00

# at least 8 cpu cores will be requested internally.
# in this sample, we employ 12 cores to get enough amount of memory.
# not sure how much is necessary/required, though.

# note about available memory:
# Available memory amount is proportional to ncpus value.
# If you need more memory, please increase ncpus in the header.

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd "${PBS_O_WORKDIR}"
fi

AF2ROOT=/local/apl/lx/alphafold2
RUNAF2=${AF2ROOT}/run_alphafold_rccs.sh

# load miniconda environment (where necessary binaries reside)
. ${AF2ROOT}/conda_init.sh

# Required:
# -o [output directory]
# -f [sequence file (FASTA)]

# Optional arguments:
# -a [alphafold code root]
#   (default: /local/apl/lx/alphafold2/alphafold-20210819)
#   /local/apl/lx/alphafold2/alphafold-20210720 can be used instead.
# -d [database root]
#   (default: /local/apl/lx/alphafold2/databases-20210819)
# -t [max template date] (default: 2021-08-20)
#   in the original example 2020-05-14 was used (CASP14 target case?)
# -m [model names]
```

```
# (default: model_1,model_2,model_3,model_4,model_5)
# -p [preset] (default: full_dbs)
# speed/quality control. available values are casp14, full_dbs, reduced_dbs
# -Q
# change model names to _ptm version; additional scores will be available
# in output file. pLDDT score will be in B-factor column of output pdbs.
sh ${RUNAF2} \
-o ./dummy_test/ \
-f query.fasta \
-Q
```