AlphaFold 2.3.1

(python 環境の準備部分と実行用 wrapper スクリプトのみ)

python 環境構築メモ

miniforge を導入済とする

```
(base) conda install -y -c conda-forge openmm=7.5.1 cudatoolkit==11.1.1 cudnn pdbfixer pip python=3.8
(base) conda install -y -c bioconda hmmer==3.3.2 hhsuite==3.3.0 kalign2==2.04
(base) pip install absl-py==1.0.0 biopython==1.79 chex==0.0.7 dm-haiku==0.0.9 dm-tree==0.1.6 immutabledict==2.0.0 jax==0.3.25 ml-collections==0.1.0 numpy==1.21.6 pandas==1.3.4 protobuf==3.20.1 scipy==1.7.0 tensorflow-cpu==2.9.0
(base) pip3 install --upgrade jax==0.3.25 jaxlib==0.3.25+cuda11.cudnn805 -f https://storage.googleapis.com/jax-releases/jax_cuda_releases.html
(base) cd miniforge2/lib/python3.8/site-packages/
(base) patch -p0 < ../../../../2.3.1/docker/openmm.patch
```

wrapper スクリプト

```
#!/bin/bash
# Description: AlphaFold non-docker version
# Author: Sanjay Kumar Srikakulam
#
#
# RCCS notes:
# This script was customized for RCCS by M. Kamiya (IMS).
# original: https://github.com/kalininalab/alphafold_non_docker

# This script is for AlphaFold 2.3.x!
# Former AlphaFold versions may not be compatible with this script!

# RCCS default value
af2root="/apl/alphafold/2.3.1"
data_dir="/apl/alphafold/databases/20230130"

max_template_date="2023-01-30"
benchmark=false
db_preset="full_dbs"
model_preset="monomer"
use_gpu=false
MYOPTS="" # variable for misc options

usage() {
        echo ""
        echo "Usage: $0 <OPTIONS>"
        echo "Required Parameters:"
        echo "-o <output_dir>   Path to a directory that will store the results."
        echo "-f <fasta_path>   Path to a FASTA file containing one sequence"
        echo ""
        echo "Optional Parameters:"
        echo "-a <alphafolddir> Path to alphafold code"
        echo "-d <data_dir>     Path to directory of supporting data"
        echo "-t <max_template_date> Maximum template release date to consider (ISO-8601 format - i.e. YYYY-MM-DD). Important if folding historical test sets (default: 2021-11-05)"
        echo "-Q                show also pTM score etc. (alias of -m monomer_ptm)"
        echo "-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')"
        echo "-g                Enable NVIDIA runtime to run with GPUs"
        echo "-a <gpu_devices>  Comma separated list of devices to pass to 'CUDA_VISIBLE_DEVICES' (default: '')"
        echo "-S                Skip relaxation of predicted structures"
        echo "-R                Skip running MSA tools and use precomputed one. NOTE: this will not check if sequence/db/conf have changed."
        echo "-s <seeds per model>  Number of seeds per model for multimer system. (Number of models (usually 5)) * (number of seeds; this param) predictions will be performed. (default: 5)"
        echo "-p <db_preset>    Choose db preset - no ensembling (full_dbs), reduced version of dbs (reduced_dbs) (default: 'full_dbs')"
        echo "-m <model_preset> Choose model preset - monomer model (monomer), monomer with extra ensembling (monomer_casp14), monomer
```

```
model with pTM head (monomer_ptm), or multimer model (multimer) (default: 'monomer')"
        echo ""
        exit 1
}

while getopts ":a:d:o:f:t:a:p:s:m:bgQRS" i; do
        case "${i}" in
        a)
                echo "INFO: set AF2 root to $OPTARG"
                af2root=$OPTARG
        ;;
        d)
                echo "INFO: set database root to $OPTARG"
                data_dir=$OPTARG
        ;;
        o)
                output_dir=$OPTARG
        ;;
        f)
                fasta_path=$OPTARG
        ;;
        t)
                max_template_date=$OPTARG
        ;;
        b)
                benchmark=true
        ;;
        g)
                use_gpu=true
        ;;
        Q)
                echo "INFO: set model_preset=monomer_ptm"
                model_preset="monomer_ptm"
        ;;
        a)
                gpu_devices=$OPTARG
        ;;
        p)
                db_preset=$OPTARG
        ;;
        m)
                model_preset=$OPTARG
        ;;
        s)
                MYOPTS="$MYOPTS --num_multimer_predictions_per_model=$OPTARG"
        ;;
        R)
                MYOPTS="$MYOPTS --use_precomputed_msas=True"
        ;;
        S)
                MYOPTS="$MYOPTS --run_relax=False"
        ;;
        esac
done

# Parse input and set defaults
if [[ "$data_dir" == "" || "$output_dir" == "" || "$fasta_path" == "" ]] ; then
    usage
fi

if [[ "$db_preset" != "full_dbs" && "$db_preset" != "reduced_dbs" ]] ; then
    echo "Unknown db_preset! Using default ('full_dbs')"
    db_preset="full_dbs"
fi
```

```
if [[ "$model_preset" != "monomer" && "$model_preset" != "monomer_casp14" && "$model_preset" != "monomer_ptm" && "$model_preset" != "multimer" ]]; then
    echo "Unknown model_preset! Using default ('monomer')"
    model_preset="monomer"
fi

alphafold_script="$af2root/run_alphafold.py"
if [ ! -f "$alphafold_script" ]; then
    echo "Alphafold python script $alphafold_script does not exist."
    exit 1
fi

if "$use_gpu" ; then
  MYOPTS="$MYOPTS --use_gpu_relax=True"
else
  MYOPTS="$MYOPTS --use_gpu_relax=False"
fi

if [[ "$gpu_devices" ]] ; then
    export CUDA_VISIBLE_DEVICES=$gpu_devices
fi

export TF_FORCE_UNIFIED_MEMORY='1'
export XLA_PYTHON_CLIENT_MEM_FRACTION='4.0'

# Binary path (change me if required)
hhblits_binary_path=$(which hhblits)
hhsearch_binary_path=$(which hhsearch)
jackhmmer_binary_path=$(which jackhmmer)
kalign_binary_path=$(which kalign)

MYOPTS="$MYOPTS --hhblits_binary_path=$hhblits_binary_path"
MYOPTS="$MYOPTS --hhsearch_binary_path=$hhsearch_binary_path"
MYOPTS="$MYOPTS --jackhmmer_binary_path=$jackhmmer_binary_path"
MYOPTS="$MYOPTS --kalign_binary_path=$kalign_binary_path"

# uniref30 path
uniref_new=$(find $data_dir -maxdepth 1 -name 'UniRef*')
if [ ! -z "$uniref_new" ]; then
  uniref_name=$(basename $uniref_new)
  uniref30_database_path="$data_dir/$uniref_name/$uniref_name"
elif [ -d "$data_dir/uniref30" ]; then
  uniref30_database_path="$data_dir/uniref30/UniRef30_2021_03"
fi

# bfd path
if [[ "$db_preset" == "reduced_dbs" ]] ; then
  small_bfd_database_path="$data_dir/small_bfd/bfd-first_non_consensus_sequences.fasta"
  MYOPTS="$MYOPTS --small_bfd_database_path=$small_bfd_database_path"
  # uniref30 not necessary
else
  bfd_database_path="$data_dir/bfd/bfd_metaclust_clu_complete_id30_c90_final_seq.sorted_opt"
  MYOPTS="$MYOPTS --bfd_database_path=$bfd_database_path"
  # uniref30 required
  MYOPTS="$MYOPTS --uniref30_database_path=$uniref30_database_path"
fi

# Path and user config (change me if required)
if [ -f $data_dir/mgnify/mgy_clusters_2022_05.fa ]; then
  mgnify_database_path="$data_dir/mgnify/mgy_clusters_2022_05.fa"
else
  mgnify_database_path="$data_dir/mgnify/mgy_clusters.fa"
fi
template_mmcif_dir="$data_dir/pdb_mmcif/mmcif_files"
obsolete_pdbs_path="$data_dir/pdb_mmcif/obsolete.dat"
```

```
uniref90_database_path="$data_dir/uniref90/uniref90.fasta"

MYOPTS="$MYOPTS --mgnify_database_path=$mgnify_database_path"
MYOPTS="$MYOPTS --template_mmcif_dir=$template_mmcif_dir"
MYOPTS="$MYOPTS --obsolete_pdbs_path=$obsolete_pdbs_path"
MYOPTS="$MYOPTS --uniref90_database_path=$uniref90_database_path"

# for multimer (pdb70 must not be specified this case)
if [[ "$model_preset" == "multimer" ]]; then
  echo "INFO: appending database paths for multimer model..."
  uniprot_database_path="$data_dir/uniprot/uniprot.fasta"
  MYOPTS="$MYOPTS --uniprot_database_path=$uniprot_database_path"
  pdb_seqres_database_path="$data_dir/pdb_seqres/pdb_seqres.txt"
  MYOPTS="$MYOPTS --pdb_seqres_database_path=$pdb_seqres_database_path"
else
  pdb70_database_path="$data_dir/pdb70/pdb70"
  MYOPTS="$MYOPTS --pdb70_database_path=$pdb70_database_path"
fi

#echo $MYOPTS

# Run AlphaFold with required parameters
$(python $alphafold_script --data_dir=$data_dir --output_dir=$output_dir --fasta_paths=$fasta_path --max_template_date=$max_template_date --
db_preset=$db_preset --model_preset=$model_preset --benchmark=$benchmark --logtostderr $MYOPTS)
```

サンプルジョブスクリプト(monomer)

```
#!/bin/sh
#PBS -l select=1:ncpus=24:mpiprocs=1:ompthreads=12
#PBS -l walltime=72:00:00

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

AF2ROOT=/apl/alphafold
RUNAF2=${AF2ROOT}/run-af-23x.sh

# pass "-a $AF2DIR" to $RUNAF2 if you want to change alphafold version
#AF2DIR=/apl/alphafold/2.3.1

# load miniconda environment (where necessary binaries reside)
. ${AF2ROOT}/mini2_init.sh

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

${RUNAF2} \
  -o ./monomer_test/ \
  -f monomer.fasta \
  -Q
```