#!/bin/bash

### HMMsearch for HPC ###

# Location: M3

# Inputs: protein sequences (eg. by prodigal)

# Writes and executes array job

# chmod +x this script and then run without sbatch

# ./resfams\_core\_hmmsearch\_HPC.sh resresfams\_core\_hmmsearch\_HPC.shfams\_core\_hmmsearch\_HPC.sh

inp\_dir="../../prodigal-M/Mackay\_MAGs/"

ext=".faa"

op\_dir="./Mackay\_MAGs/"

# Ensure versions are the same by using db version from antismash

hmm\_db="/monfs00/scratch/lwoo0007/WoodsL/miniconda/conda/envs/antismash\_env/lib/python3.8/site-packages/antismash/databases/resfam/Resfams.hmm"

hmm\_name=$(basename ${hmm\_db} ".hmm")

script\_name=${hmm\_name}\_hmmsearch\_array\_job.sh

script\_name\_helper=${hmm\_name}\_hmmsearch\_array\_helper.sh

conda\_base="/monfs00/scratch/lwoo0007/WoodsL/miniconda/bin/activate"

#Something with hmmer3

conda\_env="/monfs00/scratch/lwoo0007/WoodsL/miniconda/conda/envs/hmmer\_env"

# A few failsafes

# Make an output directory if it doesn't exist--depends a bit on the program

# as an existing output directory is sometimes required, sometimes isn't, and

# in some cases, if it exists, will derail the program by default

if [ ! -d ${op\_dir} ]

then

mkdir ${op\_dir}

fi

if [ "${ext::1}" != "." ];

then

echo ".${ext}" > $ext

fi

#Clear old scripts

if [ -f $script\_name ]

then

rm $script\_name

fi

if [ -f $script\_name\_helper ]

then

rm $script\_name\_helper

fi

cat << EOF >> $script\_name

#!/bin/bash

#SBATCH --mail-type=END,FAIL

#SBATCH --mail-user=laura.woods1@monash.edu

#SBATCH -c 4

#SBATCH --time=1:00:00

#SBATCH --partition=short

#SBATCH --job-name $script\_name

source ${conda\_base} base

conda activate ${conda\_env}

#Inherit variables

inp\_dir="${inp\_dir}"

op\_dir="${op\_dir}"

ext="${ext}"

hmm\_db="${hmm\_db}"

hmm\_name="${hmm\_name}"

EOF

cat << 'EOF' >> $script\_name

echo "This job in the array has:"

echo "- SLURM\_JOB\_ID=${SLURM\_JOB\_ID}"

echo "- SLURM\_ARRAY\_TASK\_ID=${SLURM\_ARRAY\_TASK\_ID}"

# grab our filename from a directory listing

FILES=($(ls -1 ${inp\_dir}/\*${ext}))

#### For testing ###

#FILES=$(ls -1 ${inp\_dir}/\*${ext} | head -n 5)

echo $FILES

FILENAME=${FILES[$SLURM\_ARRAY\_TASK\_ID]}

echo "My input file is ${FILENAME}"

sample\_name=$(basename ${FILENAME} ${ext})

# Not sure whether or not to include alignment

echo "COMMAND hmmsearch -o ${sample\_name}\_${hmm\_name}\_hmmsearch.out --tblout ${sample\_name}\_${hmm\_name}\_hmmsearch.tsv --cut\_ga --cpu $SLURM\_CPUS\_ON\_NODE $hmm\_db $FILENAME"

hmmsearch -o ${sample\_name}\_${hmm\_name}\_hmmsearch.out --tblout ${sample\_name}\_${hmm\_name}\_hmmsearch.tsv --cut\_ga --cpu $SLURM\_CPUS\_ON\_NODE $hmm\_db $FILENAME

EOF

cat << EOF >> $script\_name\_helper

#!/bin/bash

inp\_dir="${inp\_dir}"

op\_dir="${op\_dir}"

ext="${ext}"

script\_name="${script\_name}"

EOF

cat << 'EOF' >> $script\_name\_helper

NUMFILES=$(ls -1 ${inp\_dir}/\*${ext}\* | wc -l)

echo "There are a total of ${NUMFILES} input files"

ZBNUMFILES=$(($NUMFILES - 1))

if [ $ZBNUMFILES != 0 ]; then

echo "COMMAND: sbatch --array=0-$ZBNUMFILES $script\_name"

sbatch --array=0-$ZBNUMFILES $script\_name

else

echo "No jobs to submit--no input files in this directory."

fi

EOF

chmod +x $script\_name\_helper

./${script\_name\_helper}