#!/bin/bash

### Diamond for HPC ###

# Location: MonARCH

# Inputs: protein sequences (eg. by prodigal)

# Writes and initiates array job

# Usage: ./Greening\_metabolic\_marker\_genes\_diamond\_HPC.sh

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

ext=".faa"

op\_dir="./Mackay\_MAGs"

db\_name="Greening\_metabolic\_marker\_genes"

# Uses a specific database organised by identity threshold indicated by \_\_\_\_

diamond\_db\_folder="/monfs00/scratch/lwoo0007/WoodsL/databases/Greening\_2021\_metabolic\_marker\_genes\_databases/metabolic\_markers\_by\_DIAMOND\_ID\_cutoff/"

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

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

# User-defined thresholds

evalue=0.000001

query\_cov=0.8

max\_target\_seqs=1

script\_name=${db\_name}\_diamond\_array\_job.sh

script\_name\_helper=${db\_name}\_diamond\_array\_helper.sh

#conda\_base="/scratch/pa12/lwoo0007/miniconda/bin/activate"

#Something with diamond :)

#env="diamond\_env"

# A few fail-safes

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

then

mkdir ${op\_dir}

fi

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

then

echo ".${ext}" > $ext

fi

#Input directory fail-safe HERE!

#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\_source base

conda activate $conda\_env

#Inherit variables

folder\_suffix="${db\_name}"

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

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

ext="${ext}"

diamond\_db\_folder="${diamond\_db\_folder}"

evalue="${evalue}"

query\_cov="${query\_cov}"

max\_target\_seqs="${max\_target\_seqs}"

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}))

echo $FILES

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

echo "My input file is ${FILENAME}"

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

echo "Working on ${sample\_name}"

mkdir ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/

for cutoff in 50 60 70 75 80

do

cutoff\_str=$( printf $cutoff )

db\_name="metabolic\_markers\_DIAMOND\_ID\_cutoff\_${cutoff\_str}\_diamondDB"

echo "COMMAND diamond blastp --db ${diamond\_db\_folder}/${db\_name}.dmnd --query ${FILENAME} \

--out ${op\_dir}/${sample\_name}\_${db\_name}\_result.tsv --evalue ${evalue} --max-target-seqs ${max\_target\_seqs} \

--outfmt 6 qseqid full\_qseq sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore \

--query-cover ${query\_cov}"

diamond blastp --db ${diamond\_db\_folder}/${db\_name}.dmnd --query ${FILENAME} \

--out ${op\_dir}/${sample\_name}\_${db\_name}\_result.tsv --evalue ${evalue} --max-target-seqs ${max\_target\_seqs} \

--outfmt 6 qseqid full\_qseq sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore \

--query-cover ${query\_cov}

diamond blastp --db ${diamond\_db\_folder}/${db\_name}.dmnd --query ${FILENAME} \

--out ${op\_dir}/${sample\_name}\_${db\_name}\_result.xml --outfmt 5

awk -v var="$cutoff" '$4>var' ${op\_dir}/${sample\_name}\_${db\_name}\_result.tsv > ${op\_dir}/${sample\_name}\_${db\_name}\_result\_pident\_${cutoff\_str}.tsv

filtered\_results="${op\_dir}/${sample\_name}\_${db\_name}\_result\_pident\_${cutoff\_str}.tsv"

cat $filtered\_results >> "${sample\_name}\_diamond\_${folder\_suffix}\_combined\_pident\_cutoffs.tsv"

while read -r line

do

query\_name=$(echo "${line}" | cut -f 1)

query\_seq=$(echo "${line}" | cut -f 2)

subject=$(echo "${line}" | cut -f 3)

perc\_id=$(echo "${line}" | cut -f 4)

header=$(echo ">${query\_name}\_matched\_to\_${subject}\_w\_perc\_id\_${perc\_id}")

echo "${header}"

echo "${query\_seq}"

done < $filtered\_results > ${op\_dir}/${sample\_name}\_${db\_name}\_result"${ext}"

#while read -r line

#do

# query\_name=$(echo "${line}" | cut -f 1)

# subject=$(echo "${line}" | cut -f 3)

# perc\_id=$(echo "${line}" | cut -f 4)

# header=$(echo "${query\_name}\_matched\_to\_${subject}\_w\_perc\_id\_${perc\_id}")

# sstart=$(echo "${line}" | cut -f 10)

# ssend=$(echo "${line}" | cut -f 11)

# echo -e "${header}\t${sstart}\t${ssend}"

# done < $filtered\_results > ${op\_dir}/${sample\_name}\_${db\_name}\_filtered\_op\_for\_gff.tsv

# mv ${op\_dir}/${sample\_name}\_${db\_name}\_filtered\_op\_for\_gff.tsv ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/

mv ${op\_dir}/${sample\_name}\_${db\_name}\_result\* ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/

done

#cat ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/\*filtered\_op\_for\_gff.tsv > ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/${sample\_name}\_diamond\_${folder\_suffix}\_combined\_for\_gff.tsv

cat ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/\*\_result"${ext}" > ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/${sample\_name}\_diamond\_${folder\_suffix}\_combined.faa

mv ${sample\_name}\_diamond\_${folder\_suffix}\_combined\_pident\_cutoffs.tsv ${op\_dir}/${sample\_name}\_diamond\_${folder\_suffix}/

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}

~