workflow

July 12, 2025

1 Preparint the raw reads

[ ]:
```
# Preparing the samples: concatenating them

for i in *.fq.gz
do
    name=$(echo $i | cut -f1 -d"_")

    if [[ $i == *'_1.fq.gz' ]]
    then
        newname="${name}_R1.fq.gz"
        zcat  $i | gzip >> $newname

    elif [[ $i == *'_2.fq.gz' ]]
    then
        newname="${name}_R2.fq.gz"
         zcat  $i | gzip >> $newname
    fi
done
```

[ ]:
```
# Removing R
for i in *; do name="${i//R/}"; mv $i $name;done

# Adding R

for i in *
do
    prefix="${i%_[1,2].fq.gz}" # % for removing the pattern

    if [[ $i == *'_1.fq.gz' ]] # for string pattern matchin wee use [[]]
    then
        newname="${prefix}_R1.fq.gz"
    elif [[ $i == *'_2.fq.gz' ]]
    then
```

```
        newname="${prefix}_R2.fq.gz"
    fi

    mv $i "$newname"
done
```

2 For qc, assembly, binning, taxonomic classification and functional annotation, a ready-to-use pipeline was used, ATLAS: https://github.com/metagenome-atlas/atlas

[ ]:
```
conda activate atlas

# we added human genome to the database as the host contamination: https://www.
 ↪seqanswers.com/forum/bioinformatics/bioinformatics-aa/
 ↪37175-introducing-removehuman-human-contaminant-removal



#creating a cluster profile
cookiecutter --output-dir ~/.config/snakemake https://github.com/
 ↪metagenome-atlas/clusterprofile.git
#cookiecutter --output-dir ~/.config/snakemake https://github.com/
 ↪Snakemake-Profiles/pbs-torque.git  # first choose cluster, then choos pbs



TMPDIR=/home/projects/cu_10168/people/farpan/data/keneth/output2/tmp/$PBS_JOBID
export TMPDIR
mkdir -p $TMPDIR


atlas init --db-dir database samples -w output2 --threads 10
```

2.1 The memory must be defiend clearly in the config file through mem, in the cluster profile through mem_mb, and in the queue.tsv file. Here I am using 800GB memory.

2.2 NOTE: You may the ruby from conda env since there was a coredump error due to conflicts between ruby versions in atlas and in the base environment. This is important for the DAS tool

conda activate /home/projects/data/database2/conda_envs/75a0578aa4e28da2ac52374be6cb1540_

2.2.1 conda remove ruby –force

[ ]:
```
##Running atlas on a remote cluster with PBS jub manager system

#!/bin/bash

#PBS -W group_list=cu_10168  -A cu_10168
#PBS -e error.err
#PBS -o logs.log
#PBS -l nodes=1:ppn=40
#PBS -l mem=150gb
#PBS -l walltime=360:00:00

echo Working directory is $PBS_O_WORKDIR
cd $PBS_O_WORKDIR

TMPDIR=/home/projects/data/output/tmp/$PBS_JOBID
export TMPDIR
mkdir -p $TMPDIR


source activate atlas

#--max-mem controls the amount of memory used by atlas, this avoids error␣
 ↪casued by memory drainage for Java.
# To setup the profile, inside the ~/.config/snakemake/cluster/cluster_config.
 ↪yaml, change the queue to batch. We can see the queue name by typing qstat␣
 ↪-q.

__default__:
  queue: batch
  account: cu_10168
 # nodes: 1
 # mem_mb: 409600 #in megabyte
  threads: 40

# And

# only parameters defined in key_mapping (see below) are passed to the command␣
 ↪in the order specified.
# system: "pbs" #check if system is defined below

# pbs:
#   command: "qsub -l mem=60gb -v TMPDIR=/home/projects/cu_10168/people/farpan/
 ↪data/keneth/output2/tmp"
#   key_mapping:
#     name: "-N {}"
```

```
#     account: "-A {}"
#     queue: "-q {}"
#     threads: "-l nodes=1:ppn={}" # always use 1 node
#   # mem_mb: "-l mem={}mb"
#     time_min: "-l walltime={}00" #min= seconds x 100



## The queues.tsv file shoudl look like this
# queue   priority        threads mem_mb  time_min
# batch   1       40      61440   5000
# small   1       40      382000  4320
# large   2       1040    382000  4320
# hugemem 3       160     1534000 4320
# longrun 4       40      382000  20160
# hugemem_longrun 6       40      1534000 5760

#Head of config file

######################################################################
####                  _______   _                    _____       ####
####         /\      |__   __| | |          /\      / ____|     ####
####        /  \        | |    | |         /  \    | (___       ####
####       / /\ \       | |    | |        / /\ \    \___ \      ####
####      / ____ \      | |    | |____   / ____ \   ____) |     ####
####     /_/    \_\     |_|    |______| /_/    \_\ |_____/      ####
####                                                            ####
######################################################################

#  For more details about the config values see:
#  https://metagenome-atlas.rtfd.io


########################
# Execution parameters
########################
# threads and memory (GB) for most jobs especially from BBtools, which are␣
 ↪memory demanding
threads: 40
mem: 192

# threads and memory for jobs needing high amount of memory. e.g GTDB-tk,checkm␣
 ↪or assembly
java_mem: 150
large_mem: 192
large_threads: 40
assembly_threads: 40
```

```
assembly_memory: 192
simplejob_mem: 20
simplejob_threads: 10




atlas run all --profile cluster --jobs 40  -w ./output2 -c ./output2/config.
 ↪yaml --keep-going  #-max-mem 500 -n #--latency-wait 60
#--resources mem=400

#--report atlas_report.html

#rm -rf $TMPDIR


#use qstat -r to check the job
#qdel <jobid> to delete a job
# In this run we did not activate ~/.config/snakemake/cluster/queue.tsv.example␣
 ↪as it would return an error for the submissions
```

2.3 Expected outputs from ATLAS

taxonomy_file = "gtdb_taxonomy.tsv"

tree_file = "gtdbtk.bac120.nwk"

tree_arch = "gtdbtk.ar53.nwk"

quality_file = "genome_quality.tsv"

counts_file = "counts_genomes.parquet"

abundance_file = "median_coverage_genomes.parquet"

readstats_file = "read_counts.tsv"

keggmodules_file = "kegg_modules.tsv"

dram= "dram_annotations.tsv"

dram_xlsx = "metabolism_summary.xlsx"

gene2genomes= "gene2genome.parquet"

bin2genome = "allbins2genome.tsv"

3 Gene catalog

coverage_stats = "Genecatalog/counts/gene_coverage_stats.parquet"

coverage = "Genecatalog/counts/median_coverage.h5"

counts = “Genecatalog/counts/Nmapped_reads.h5”

sample_stats = “Genecatalog/counts/sample_coverage_stats.tsv”

geneinfo = “Genecatalog/clustering/orf_info.parquet”

eggnog = “Genecatalog/annotations/eggNOG.parquet”

kegg = “Genecatalog/annotations/dram/kegg.parquet”

cazy = “Genecatalog/annotations/dram/cazy.parquet”

pfam = “Genecatalog/annotations/dram/pfam.parquet”

3.1 Gene annotation by prokka

[ ]:
```
#!/bin/bash


base_folder="chunk"
num_chunks=10

# Loop over each chunk folder
for i in $(seq 4 $num_chunks); do
    chunk_folder="${base_folder}${i}"


    pbs_script="${chunk_folder}_prokka.pbs"

    # Write the PBS submission script
    cat << EOF > $pbs_script
#!/bin/bash
#PBS -W group_list=cu_10168  -A cu_10168
#PBS -e prok_${chunk_folder}_error.err
#PBS -o prok_${chunk_folder}_logs.log
#PBS -l nodes=1:ppn=40
#PBS -l mem=40gb
#PBS -l walltime=8:00:00

echo Working directory is \$PBS_O_WORKDIR
cd \$PBS_O_WORKDIR

TMPDIR=/home/projects/cu_10168/people/farpan/data/keneth/results/
 ↪tmp_prokka_${chunk_folder}/
export TMPDIR
mkdir -p \$TMPDIR

echo "This is tempdir: \${TMPDIR}" > dir_${chunk_folder}.log

for fasta_file in ${chunk_folder}/*.fasta; do
```

```
    name=\$(basename \$fasta_file | cut -f1 -d'_')
    dest="./prok_out/\${name}_prokka"
    prokka --outdir "\$dest" --usegenus --metagenome --prefix "\$name"␣
 ↪"\$fasta_file" --cpus 40 > "\${name}_log.txt" 2>&1
done
EOF

    # Submit the generated PBS script
    qsub $pbs_script
done
```