Pronghorn Homework Module

All personal directories are located: /data/gpfs/assoc/biol\_bids-1/

Setting up

Submission script is located:

/data/gpfs/assoc/biol\_bids-1/atram.slurm.sh

Copy this file to your personal directory.

In your personal directory you will need to create two directories:

mkdir temp

mkdir atram\_out

Below is an example of the submission script and the bold sections within square brackets should be changed accordingly.

*atram.slurm.sh*

#!/usr/bin/bash -l

#SBATCH --job-name=**[job name]**

#SBATCH --account=cpu-s5-biol\_bids-1

#SBATCH --partition=cpu-core-0

#SBATCH --reservation=cpu-s5-biol\_bids-1\_51

#SBATCH --nodes=1

#SBATCH --ntasks=1

#SBATCH --cpus-per-task=1

#SBATCH --time=14-00:00

#SBATCH --mem-per-cpu=3500M

#SBATCH --output=hostname\_%j.out

#SBATCH --error=hostname\_%j.err

#SBATCH --mail-type=ALL

#SBATCH --mail-user=**[your email]**

singularity exec /apps/atram/aTRAM.sif python3 /aTRAM/atram.py -b /data/gpfs/assoc/biol\_bids-1/Course\_content/atram\_files/AlspHabad/AlspHabad -t /data/gpfs/assoc/biol\_bids-1/**[netID]**/temp -Q /data/gpfs/assoc/biol\_bids-1/Course\_content/atram\_files/query\_files/**[Number of genes]**\_proteins.fasta -a trinity -o /data/gpfs/assoc/biol\_bids-1/**[netID]**/atram\_out/

Target files

All target files are found here: /data/gpfs/assoc/biol\_bids-1/Course\_content/atram\_files/query\_files/

Each file begins with the number of genes it contains. In class you will assemble 2 genes. For your homework you will assemble 20 genes.

Processors

You will need to select the number of processors you want to use per job. You can change the number of tasks and CPUs per task in the SBATCH options in the submission script. Pronghorn has 32 CPUs per node. Pronghorn will allocate the resources you request to aTRAM as it runs. You can start with a single task and CPU per task, measure the gene assembly rate, and adapt the computational resources from there.

Submitting

Submit the job using:

sbatch atram.slurm.sh

Queue

To check on the status of your job, use squeue in the command line. This will pull up a list of all of the jobs currently running or waiting in the queue. You will see your job by the job name and your user name. If your job is waiting in the queue, the time will say 0:00. Once it starts running the timer will start. You can check on your jobs only with squeue -u <USERNAME>

**For your homework you will assemble 20 genes.**

**Answer the following questions:**

1. **What is the purpose of High Performance Computers (HPCs) and when might you use them?**

High Performance Computers uses parallel data processing to improve computing performance and increasing efficiency and reliability when processing data. You need this type of technology when using AI, machine learning, or other processes which require massive amounts of data and high-performance computing.

**2. What is a slurm script? Paste the path of your slurm script here.**

A slurm script is a script which gives information for how a job is to be run on a HPC such a pronghorn. Specifies cpus, time, outputs, and where the request is coming from.

#!/usr/bin/bash -l

#SBATCH --job-name=**mcvatram2**

#SBATCH --account=cpu-s5-biol\_bids-1

#SBATCH --partition=cpu-core-0

#SBATCH --reservation=cpu-s5-biol\_bids-1\_51

#SBATCH --nodes=1

#SBATCH --ntasks=1

#SBATCH --cpus-per-task=5

#SBATCH --time=1-00:00

#SBATCH --mem-per-cpu=3500M

#SBATCH --output=hostname\_%j.out

#SBATCH --error=hostname\_%j.err

#SBATCH --mail-type=ALL

#SBATCH --mail-user=**mmcvicar**

singularity exec /apps/atram/aTRAM.sif python3 /aTRAM/atram.py -b /data/gpfs/assoc/biol\_bids-1/Course\_content/atram\_files/AlspHabad/AlspHabad -t /data/gpfs/assoc/biol\_bids-1/**mmcvicar**/temp -Q /data/gpfs/assoc/biol\_bids-1/Course\_content/atram\_files/query\_files/**20**\_proteins.fasta -a trinity -o /data/gpfs/assoc/biol\_bids-1/**mmcvicar/** atram\_out/

**3. Paste the path to your gene assemblies on pronghorn.**

/data/gpfs/assoc/biol\_bids-1/mmcvicar/atram\_out

**4. How many CPUs did you ask for?**

I asked for 5 CPUs.

**5. How many genes assembled?**

I assembled 20 genes.

**6. Did all genes assemble? Why or why not?**

Yes, it seems like they all did (there were no errors reported). I had enough CPUs and time to run properly.

**7. How long did it take for them to assemble? What processes might have slowed this down?**

It took 51 minutes for the genes to assemble. There were more genes and if I had done more CPUs it would have run faster.