Quast on HPC

What is Quast?

QUAST stands for Quality Assessment Tool. It evaluates genome/metagenome assemblies by computing various metrics. The current QUAST toolkit includes the general QUAST tool for genome assemblies, MetaQUAST, the extension for metagenomic datasets, QUAST-LG, the extension for large genomes (e.g., mammalians), and Icarus, the interactive visualizer for these tools.

More info on QUAST:

GitHub-About

References-Manual

Versions Available:

The following versions are available on the cluster:

- Quast–v5.0
- Quast –v4.4

How to load Quast?

To load Quast on the HPC, use the following commands:

```
module avail bio/quast
#This will show all the available versions of quast
```

Use the following command to load the desired version.

```
module load bio/quast/5.0
```

To verify if the module and dependencies are loaded correctly, use the following command.

```
module list
```

This should list all the software and dependencies that are loaded.

How to use Quast?

There are two ways to use most of the software on the cluster.

To see the usages and options available, use the following command:

```
quast.py
```

The Interactive Way:

To run the quast in an interactive session, jump into a bash terminal of a compute node using the following command:

```
srun -p main --qos main -n 1 -c 12  --mem 16G --pty bash
```

Now, load the quast module and start your job interactively on the terminal.

Note: This is preferred usage in case of short runtime jobs and debugging. After you exit the terminal session, your job will be terminated unlike batch method.

The Script method:

To use this method, the user will need to write a batch script and submit it to a slrum.

To copy the example file, use the following command,

```
mkdir ~/qst-test && cd ~/qst-test
cp -r /share/apps/quast/quast-5.0.0/test_data/ ~/qst-test/
```

All the test data required to run will be copied under test_data directory.

```
#!/bin/bash
#SBATCH --jobname=qst-test  # create a short name for your job
#SBATCH -n 1
#SBATCH -c 8           # number of cores
#SBATCH --mem=8G       # total memory per node
#SBATCH --time=00:15:00     # total run time limit (HH:MM:SS)
#SBATCH -p main
#SBATCH --qos main
#SBATCH -e err.err
#SBATCH -o out.out

module load bio/quast/

# Move to test data directory
cd $SLURM_SUBMIT_DIR/test_data

# Run quast on the test files
quast.py contigs_1. fasta contigs_2.fasta -R reference.fasta.gz \
--features genes.gff -t 4 -o ../results/
```

The output directory is specified by –o flag. All the results will be saved in the results directory.

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp