FastStructure on HPC cluster:

What is FastStructure?

fastStructure is a fast algorithm for inferring population structure from large SNP genotype data. It is based on a variational Bayesian framework for posterior inference and is written in Python2.x.

The documentation for the FastStructures is found in the following links:

GitHub

Versions Available:

- fastStructures with GSL

How to load a version of FastStructure?

To load a version of FastStructures on the HPC, use the following command:

```
module avail bio/faststructure
```

The version will be listed. To use a version of software, use following command:

```
module load bio/faststructure
```

Verify by using this command:

```
module list
```

The loaded software and dependencies, python and gls in this case, will be shown.

How to use FastStructures on the cluster?

There are two methods to run faststructure on the cluster.

The Interactive Way:

To run the program interactively, follow the steps:

```
#Open a bash session on compute node
srun -p main --qos main -n 1 -c 12 --mem 10G --pty bash

#load the module
module load bio/faststructure

# start your commands here
structure.py --help
#
```

This method is ideal for short runs which produces runtime output and to debug the codes.

The Script:

To run a slurm job, the user must prepare input files. For this example, get input files with,

```
#Copy the input files required to run fast structures
mkdir test && cd test
cp -r /share/apps/faststructure/fastStructure .
#Now all the test files are copied
```

```
# Make a sbatch script
touch script.sbatch
```

Use the following template for the script,

```
#!/bin/bash
#SBATCH -J FastStructures-test
#SBATCH -n 1
#SBATCH –mem=16G
#SBATCH -p main
#SBATCH --qos main
#SBATCH -o FastStructures_sim_out-%J.txt
#SBATCH -e FastStructures_sim_out-%J.txt
#SBATCH -t 30

module load bio/faststructure        # load the module


cd $SLURM_SUBMIT_DIR                    # Moving to the test directory



#Run FastStructures
structure.py -K 3 --input=test/testdata --output=testoutput_simple --full --seed=100
```

Schedule the job with the following sbatch command.

sbatch script.sbatch

All the processed files will be generated in the same directory as the sbatch script.

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