

## Raxml on HPC cluster:

### *What is Raxml?*

Raxml is a phylogenetic tree inference tool which uses maximum-likelihood (ML) optimality criterion. Its search heuristic is based on iteratively performing a series of Subtree Pruning and Regrafting (SPR) moves, which allows to quickly navigate to the best-know ML tree.

RAxML-NG offers improvement in speed, flexibility, and user-friendliness over the previous versions.

The documentation for the Raxml is found on:

<https://github.com/amkozlov/raxml-ng>

<https://github.com/amkozlov/raxml-ng/wiki/Tutorial#preparing-the-alignment>

### Versions Available:

RAxML-8.2.4

RAxML-8.2.12

RAxML-ng

### *How to load a version of Raxml?*

To load a version of Raxml on the HPC on current terminal session, use the following command:

`module avail bio/raxml`

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

```
module load bio/raxml/ng
```

Verify by using this command:

```
module list
```

The loaded software and dependencies will be shown.

### *How to use Raxml on the cluster?*

Here is one example of how to use the Raxml on the cluster. To get a test file to process through Raxml, use the following commands

```
# Make a test directory on home folder
cd ~ && mkdir raxml_test && cd raxml_test

# download test file
curl -L -o test.phy https://adelaide.figshare.com/ndownloader/files/14243024

# Now after you loaded your module and downloaded test files, we can make a script to
run Raxml
```

### The Script:

```
#!/bin/bash
$SBATCH -J Jobname # Jobname
#SBATCH -n 1       # Nodes per task
#SBATCH -mem=5G    # Must need to allocate memory to run raxml - at least 4G
#SBATCH -p main    # Partition
#SBATCH --qos main # Quality of service
#SBATCH -o Raxml_sim_out-%J.txt # STDOUT Out file
#SBATCH -e Raxml_sim_out-%J.txt # Error file
#SBATCH -t 25      # time in minuets
```

```
module load bio/raxml/ng          # load the module

cd $SLURM_SUBMIT_DIR              # Moving to the test directory

#runs raxml command on test.phy
raxml-ng --msa test.phy --model GTR+G --prefix first --seed 2 --threads 3
```

Schedule the job with following sbatch command.

```
sbatch myscript.sh
```

All the processed file will be written in a same directory as the script. The required file will be generated there.

## ***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>

Resources:

<https://github.com/amkozlov/raxml-ng/wiki/Tutorial#preparing-the-alignment>

clear