Phylogenetic Analyses on HiPerGator

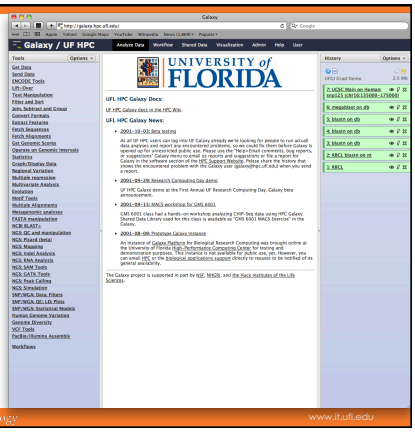
Matt Gitzendanner magitz@ufl.edu

HiPerGator

The University of Florida Supercomputer for Research

- 16,384 cores (total of ~20,000 today)
- Infiniband interconnect
- >3PB fast, high-availability, storage

Galaxy



Galaxy: Data intensive biology *for everyone*

- Accessible, reproducible, transparent computational biology
- galaxy.hpc.ufl.edu
  - Local instance of Galaxy
    - Faster access to storage, easier upload
    - Local compute resources
    - Local control

Galaxy

- RAxML
- Garli
- Beast
  - TreeAnnotator

Phylogenetics

- RaXML – Maximum Likelihood based inference of large phylogenetic trees
- Garli phylogenetic inference using the maximum-likelihood
- Beast Bayesian MCMC analysis of molecular sequences.
- TreeAnnotator BEAST tree annotator.

RaXML (version 1.0.0)

Model Type:
Nucleotide

Substitution Model (-m):
GTRCAT

Random seed used for the parsimony inferences (-p):
1234567890

RAxML options to use:
Required options only

The required minimal settings are the input file and the substitution model. To specify extra options select the 'Full option list'

Sequence File (relaxed PHYLIP format) (-s):
1: dna.phy

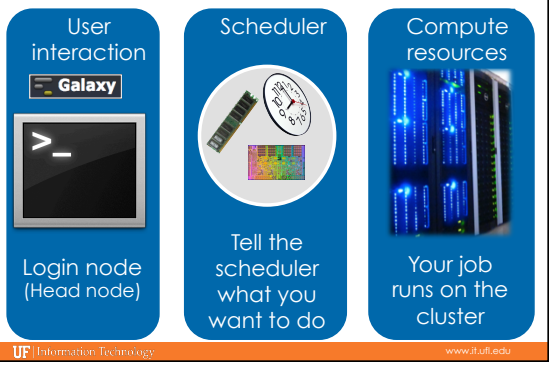
Execute

The full RAxML options list is LONG!

RAxML on Galaxy

- Make sure your dataset doesn't have invariant sites.
- Load dataset into RAxML, it will delete those sites and output a reduced dataset that you can use.

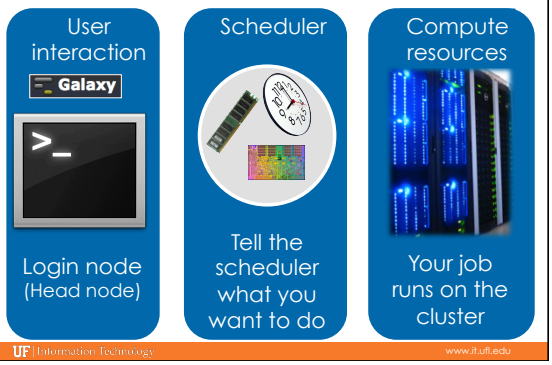
Cluster basics



Galaxy uses the scheduler too

- RAxML and BEAST
  - nodes=1:ppn=8
  - pmem=1gb
  - walltime=166:00:00 (~7days)
- Command line offers more flexibility

Cluster basics



Running RAxML

```
raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -s dna.phy\
  -p 12345 -x 12345 -N 100 -n dna -T 4
```

A standard RAxML command

Running RAxML

```
#!/bin/bash
raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -s dna.phy\
  -p 12345 -x 12345 -N 100 -n dna -T 4
```

Put it in a file and make it a bash script

Running RAxML

```
#!/bin/bash

module load raxml
raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -s dna.phy\
  -p 12345 -x 12345 -N 100 -n dna -T 4
```

Use the HPC's module system to make life easier—don't need to worry about PATHs

Running RAxML

```
#!/bin/bash
#PBS -M magitz@ufl.edu
#PBS -m abe
#PBS -o RAxML.out
#PBS -e RAxML.err
#PBS -l nodes=1:ppn=4
#PBS -l pmem=1gb
#PBS -l walltime=05:00:00

cd $PBS_O_WORKDIR
module load raxml
raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -s dna.phy\
  -p 12345 -x 12345 -N 100 -n dna -T 4
```

Add information for the scheduler
-M email address
-m when to email **a**bort, **b**egin, **e**nd
-o output file name
-e error file name
-l number of nodes and cores to use
-l per processor memory request
-l how long will job take?

Change to the directory from which the job was submitted (not $HOME)

Running RAxML

```
#!/bin/bash
#PBS -M magitz@ufl.edu
#PBS -m abe
#PBS -o RAxML.$PBS_JOBID.out
#PBS -e RAxML.$PBS_JOBID.err
#PBS -l nodes=1:ppn=4
#PBS -l pmem=1gb
#PBS -l walltime=05:00:00

cd $PBS_O_WORKDIR
module load raxml
raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -s dna.phy\
  -p $RANDOM -x $RANDOM -N 100 -n dna –T $PBS_NUM_PPN
```

Make use of some other PBS and environment variables

Scheduling a job

- Need to tell scheduler what you want to do
  - **How many CPUs** you want and how you want them grouped
  - **How much RAM** your job will use
  - **How long** your job will run
  - The commands that will be run

Nodes and processors

Single processor apps:
#PBS –l nodes=1:ppn=1

Threaded (& MPI) apps:
#PBS –l nodes=1:ppn=4

MPI apps:
#PBS –l nodes=2:ppn=32

RAM

#PBS –l pmem=900mb

- Lots to consider, but do your best at estimating RAM needed for job
- Over about 4GB of RAM, "costs" toward CPU allocation

Wasted RAM leads to idle CPUs and low job throughput

Processor Equivalents

- Accounts for large RAM requests
- Average ~4GB RAM/core

1 core, 10GB RAM: ~2.5 PEs
1 core, 60GB RAM: ~15 PEs

- Non-investor's limit: 8 PEs
- Investor limits are based on PEs

- `pbs_info –f <job_file>`

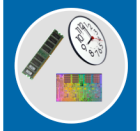
Walltime

#PBS –l walltime=00:50:00

- Fairly straight forward
- As with all resource requests, accuracy helps ensure ***your*** jobs and all other jobs will run sooner

	Maximum	Short	Long
Investor	31 days	<12 hrs	7 days
Other	7 days	<12 hrs	3 days

Scheduler



Tell the scheduler what you want to do

RAxML 8.0.0

- raxmlHPC-SSE3 Let us know if you need this.
  - Single threaded
- raxmlHPC-PTHREADS-SSE3
  - Multi-threaded, all on one node
  - E.g.: nodes=1:ppn=8
- ramlHPC-MPI-SSE3
  - MPI
  - E.g.: nodes=2:ppn=10
- raxmlHPC-HYBRID-SSE3
  - MPI and multi-threaded, span multiple nodes
  - E.g: nodes=4:ppn=8

RAxML parallelism

- Course-grained
  - Can go across multiple nodes (MPI or HYBRID)
  - Parallelizes bootstrap replicates
  - Independent tree searches
- Fine-grained
  - All CPUs on same node (PTHREADS or HYBRID)
  - Parallelizes single tree search

RAxML MPI

```
#PBS -l nodes=1:ppn=10
#PBS -l pmem=900mb
#PBS -l walltime=24:00:00
#
cd $PBS_O_WORKDIR

# Load the raxml environment
module load intel openmpi raxml

mpiexec -bynode raxmlHPC-MPI-SSE3 -f d \
  -m GTRCAT -s your_data.phy -n output_name \
  -p $RANDOM -b $RANDOM -N 500
```

RAxML Hybrid

```
#PBS -l nodes=1:ppn=20
#PBS -l pmem=900mb
#PBS -l walltime=24:00:00
#
cd $PBS_O_WORKDIR

# Load the raxml environment
module load intel openmpi raxml

mpiexec -bynode -np 5 raxmlHPC-MPI-SSE3 -f d \
  -m GTRCAT -s your_data.phy -n output_name \
  -T 4 -p $RANDOM -b $RANDOM -N 500
```


ExaML

- Exascale Maximum Likelihood code for phylogenetic inference using MPI

```
…
#PBS –l nodes=2:ppn=12
#PBS –l pmem=1gb
#PBS –l walltime=24:00:00
…
module load intel openmpi examl
mpiexec examl –s data.part.binary –n data –m PSR \
  -t starting.tree –r checkpoint –S -Q
```

Can all be on one node: nodes=1:ppn=12

- Need to convert .phy into binary file:

```
parser –s data.phy –m DNA –q partition.txt\
 -n data.part
```

- Need a starting tree--RAxML:
  - `raxmlHPC-PTHREADS-SSE3 –y –m GTRCAT –s data.phy –p 1234 –n starting.tree`

MrBayes 3.2.1

- mrbayes
  - mb –single threaded
  - E.g.: nodes=1:ppn=1
- intel openmpi mrbayes
  - mb –MPI version,
  - Can span multiple nodes
    - But doesn't need to: **nodes=1:ppn=8 is much preferred** to nodes=8:ppn=1
      - Faster for your job, fewer points of failure, doesn't partially occupy lots of nodes
    - To run: `mpiexec mb test.nex`

GARLI

- For single ML search
  - Single threaded
  - Multi-threaded, probably not worth it
- For bootstrap
  - MPI, splits each replicate onto a processor

Others (see wiki)

- BayesRate (speciation/extinction rates)
- BEAST (Bayesian dating)
- ETE2 (python module for tree manipulation)
- HyPhy (phylogenetics, molecular evolution, machine learning)
- PartitionFinder (Best fit partitioning scheme)
- PhyML (ML)
- RADICAL (Phylogenomic)
- SATé (Simultaneous alignment & inference)
- SPIMAP (Bayesian gene tree)
- TreeFix (Gene tree/species tree)

UF Research Computing

- Help and Support (Continued)
  - http://wiki.hpc.ufl.edu
    - Documents on hardware and software resources
    - Various user guides
    - Many sample submission scripts
  - http://hpc.ufl.edu/support
    - Frequently Asked Questions
    - Account set up and maintenance