Unit 12 Introduction

Systematics (ZOOL 6305)

Introduction

Goals

We will explore the major* software packages commonly* used to implement the major* phylogenetic methods.

Format (flexible)

- Wednesday: Discuss Analyses
- Thursday: Implement

Evaluation

- *Final Exam (Dec 2)*– Multi-format test over phylogenetic methods, software, implementation and interpretation.
- *Final Project* (next slide)

Final project

- Full phylogenetic analysis of your own or supplied data.
- Write up - paper style
- We will work on this throughout the rest of the semester
- Please discuss your dataset with me in the next few days and submit a short write-up. (Oct. 28th).

- Due Date: Dec 2.

http://evolution.genetics.washington.edu/phylip/software.html

There are **hundreds** of programs for phylogenetics that range in their usefulness, availability, cost, accuracy, recognition, methods, age, user interface, etc…

It is **YOUR** job to find the best program for **YOUR** project. There is no way (or desire) to discuss each program. 90% of this section is to expose you to the most commonly used programs and give you the tools to pick up the others we do not cover.

The Black Box

Completely random data will still generate a tree. It is your responsibility to know

1. the quality of your input data
2. general methodology of the program
3. how the methodology is implemented
4. what the options/parameters are
5. how to run the program
6. how to judge the results

The Black Box

Completely random data will still generate a tree. It is your responsibility to know

1. Is my data high quality, homologous, in the proper format, in order…?
2. What method does MrBayes use to build a trees?
3. How is this method implemented?
4. What options are most appropriate (nucleotide substitution models, burnin)?
5. What files are needed to run MrBayes and in what format?
6. Is the tree comparable to what I expected? Is it reproducible?

The most important thing is to…

The Black Box

Completely random data will still generate a tree. It is your responsibility to know

1. What method does MrBayes use to build a trees?
2. How is this method implemented?
3. What options are most appropriate (nucleotide substitution models, burnin)?
4. What files are needed to run MrBayes and in what format?
5. Is my data high quality, homologous, in the proper format, in order…?
6. Is the tree comparable to what I expected? Is it reproducible?

The most important thing is to…

be confident in your methodology, implementation and results.

OSs

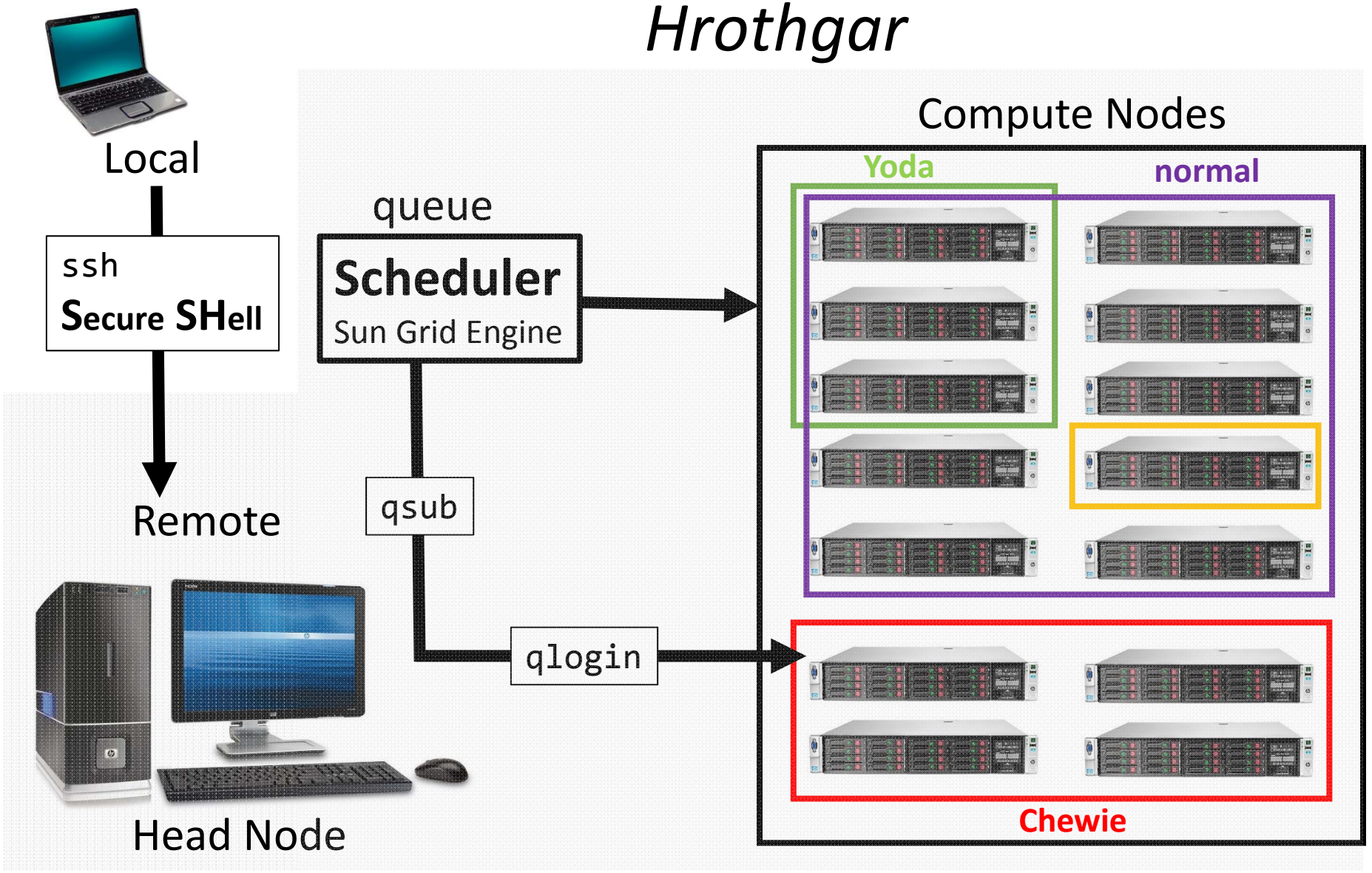
- Windows, Mac, Unix
- Local vs Remote
- Graphical User Interface vs. Command Line Interface
- TTU's HPCC

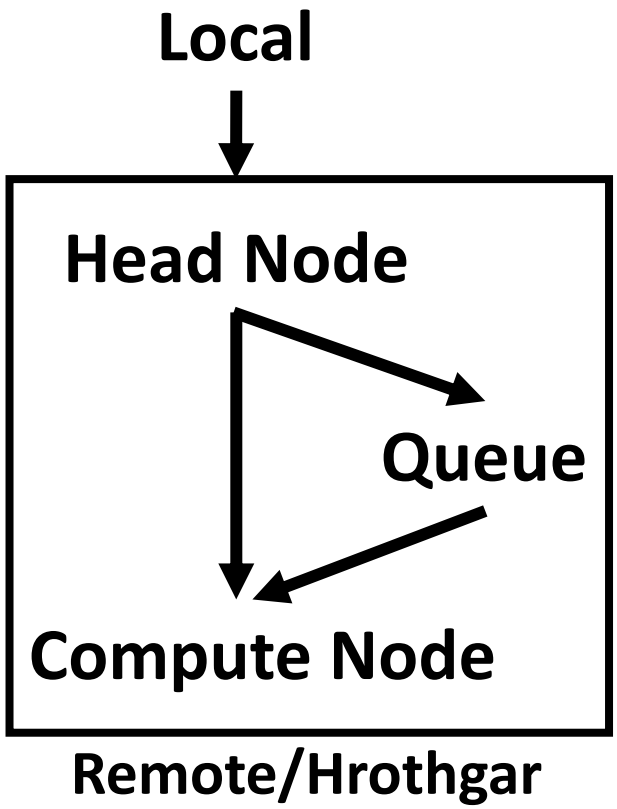
HIGH PERFORMANCE COMPUTING CENTER

HROTHGAR ---- TTU HPCC's NEWEST COMPUTATIONAL RESOURCE

Texas Tech University's High Performance Computing Center (HPCC) was established in 1999 to promote research and teaching on campus through integrating leading-edge high performance computing and visualization for the faculty, staff and students of Texas Tech University.

Hrothgar


Local
ssh
Secure SHell
Remote
Head Node
queue
Scheduler
Sun Grid Engine
qsub
qlogin
Compute Nodes
Yoda
normal
Chewie


Local
Head Node
Queue
Compute Node
Remote/Hrothgar

Hrothgar vs your laptop

- Hrothgar has 11,616 available processing units.
- Your laptop has …?

- Hrothgar is always on*
- Your laptop is…?

- Hrothgar has unlimited storage.
- Your laptop has…?

*except when its not

Using Hrothgar

- Creating an account
- Logging on
- Interacting (interactive)
- Submitting
- Interacting with the queue

Logging on to HPCC

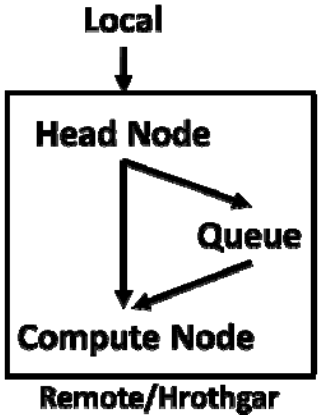
- Loggin on via the CLI

```
ssh <address>
ssh <username>@<address>
ssh <username>@<address>:/path/to/directory
```

- Example

```
ssh neal@hrothgar.hpcc.ttu.edu:/lustre/work/bats
```

or you can use a GUI

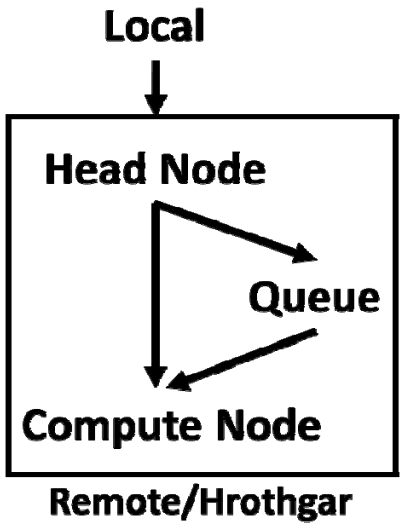


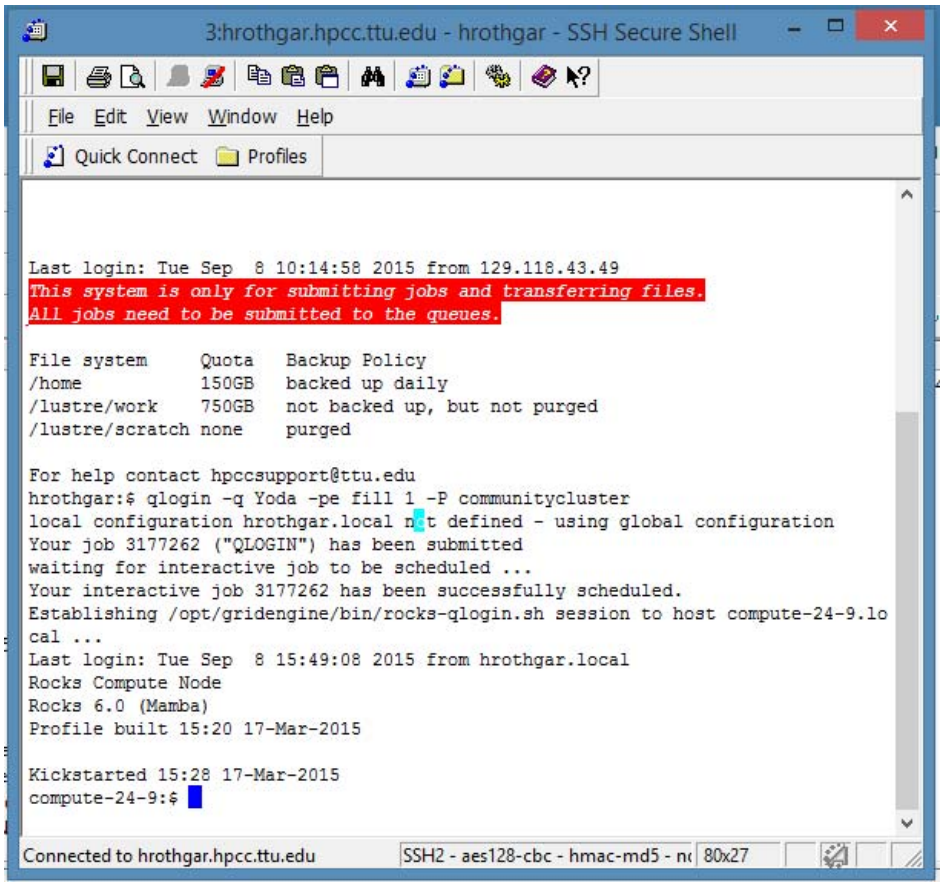
When you first log on you are on the ____?

`qlogin` (Interactive session)

- Head vs. compute node
- `qlogin -q <queueName> -pe fill <thread#> -P communitycluster`
- `qlogin -q Yoda -pe fill 1 -P communitycluster`

Notice the change that occurs at the prompt





qsub (Submitting)

`#$ -V`	use current settings
`#$ -cwd`	use current working directory
`#$ -S /bin/bash`	use bash shell
`#$ -N test`	Job name
`#$ -o $JOB_NAME.o$JOB_ID`	stdout file name
`#$ -e $JOB_NAME.e$JOB_ID`	stderr file name
`#$ -q phylo`	**queue name**
`#$ -pe fill 12`	env & core request
`#$ -P hrothgar`	user group

`x=0`
`while [ $x -le 100]`
`do`
`  echo $x`
`  x=$(($x+1)`
`  sleep 1`
`done`
	commands

Manipulating the queue

- `qsub` – submits job to the queue via submission script
- `qstat` – checks status of your jobs
- `qdel` – deletes job
- `qhold` – places a hold on a job
- `qrls` - releases a held job
- `qalter` – alter parameters of a job
- `qstat -g c` – check status of cluster group

```
hrothgar:$ qall
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
3177196 2.26058 QLOGIN     samangum     r     09/08/2015 13:17:02 Chewie@compute-24-11.local         5
3177229 0.25056 QLOGIN     roplatt      r     09/08/2015 15:49:57 Chewie@compute-24-11.local         3
3177221 0.25808 sort1      daray        r     09/08/2015 15:09:26 Chewie@compute-24-12.local        30
3170873 0.26365 assem3     daray        r     09/07/2015 17:21:21 Chewie@compute-24-15.local        50
3171090 0.25251 sort2      daray        r     09/08/2015 08:26:58 R2D2@compute-24-16.local          10
3171155 2.25947 QLOGIN     lblancob     r     09/08/2015 11:29:38 R2D2@compute-24-16.local           1
3168522 0.25251 QLOGIN     roplatt      r     09/07/2015 10:35:57 Yoda@compute-24-9.local           10
3171111 0.25000 QLOGIN     daray        r     09/08/2015 09:55:59 Yoda@compute-24-9.local            1
3119479 0.53777 doliftshar davichen     Eqw   08/12/2015 11:17:13                                    1
3159659 0.25000 Prat.py    pkottapa     qw    09/01/2015 10:26:09                                    1
hrothgar:$
```

Basic Unix Commands

- `ls` – list files in directory
- `cd` – change directories
- `pwd` – print working directory
- `mkdir` – make directory
- `rm` – remove
- `cp` - copy
- `mv` – move file/directory
- `cat` – concatenate/display
- `head` – show the first 10 lines of a file
- `tail` – show the last 10 lines of a file
- `less` – scroll through a file
- `nano` – basic text editor

The commands we use will get more complex as we go along, but you should be able to accomplish most tasks with these.

The Unix Philosophy

“Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface”

-Doug McIlroy

| (shift+\) – The pipe

Hrothgar file structure

- /home/<userName>
  - Use sparingly, 150 Gb, backed up daily
- /lustre/work/<userName>
  - Use more frequently, 750 Gb, never backed up, never purged
- /lustre/scratch/<userName>
  - Use often, unlimited storage, never backed up, purged (but notified in advance).

Data Formats

The Big three

- Phylip
- Fasta
- Nexus*

Data Formats (The Big 3)

phylip format

- standard format for phylip group of programs
- common extensions .phy or .ph
- Variants: sequential vs. interleaved, relaxed vs strict naming
- first line designates the number of sequences and their length

Interleaved

```
 3 100
seq1    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
seq2    TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
seq3    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
 3 100
seq1    AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
seq2    TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT
seq3    CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC

        AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
        TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT TTTTTTTTTT
        CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC CCCCCCCCCC
```

Sequential

```
 3 100
seq1    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
seq2    TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
        TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
seq3    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
 3 100
seq1    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
seq2    TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
seq3    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Data Formats (The Big 3)

fasta format

- originally used for the FastA/Fast-All
- common extensions: .fa .fas .fasta .mfa
- The basic format used in bioinformatics

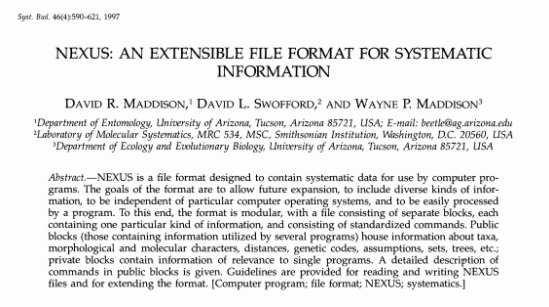
```
>seq1
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
>seq2
TTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTT
>seq3
CCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCC
```

```
>seq1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
>seq2
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
>seq3
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Data Formats (The Big 3)

Nexus format

- initially implemented in PAUP*
- common extensions: .nex, .nexus
- more of a scripting format than a sequence format (contains information to communicate with the program)
- complex
- information stored in "blocks" (`TAXA`, `DATA`, `TREES`)
- each block begins with `BEGIN <block name>;`. This information is read by the program and can be skipped if the block isn't recognized



Syst. Biol. 46(4):590–621, 1997

NEXUS: AN EXTENSIBLE FILE FORMAT FOR SYSTEMATIC INFORMATION

DAVID R. MADDISON,[1] DAVID L. SWOFFORD,[2] AND WAYNE P. MADDISON[3]

[1]*Department of Entomology, University of Arizona, Tucson, Arizona 85721, USA; E-mail: beetle@ag.arizona.edu*
[2]*Laboratory of Molecular Systematics, MRC 534, MSC, Smithsonian Institution, Washington, D.C. 20560, USA*
[3]*Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, Arizona 85721, USA*

Abstract.—NEXUS is a file format designed to contain systematic data for use by computer programs. The goals of the format are to allow future expansion, to include diverse kinds of information, to be independent of particular computer operating systems, and to be easily processed by a program. To this end, the format is modular, with a file consisting of separate blocks, each containing one particular kind of information, and consisting of standardized commands. Public blocks (those containing information utilized by several programs) house information about taxa, morphological and molecular characters, distances, genetic codes, assumptions, sets, trees, etc.; private blocks contain information of relevance to single programs. A detailed description of commands in public blocks is given. Guidelines are provided for reading and writing NEXUS files and for extending the format. [Computer program; file format; NEXUS; systematics.]

Data Formats (The Big 3)

Nexus format

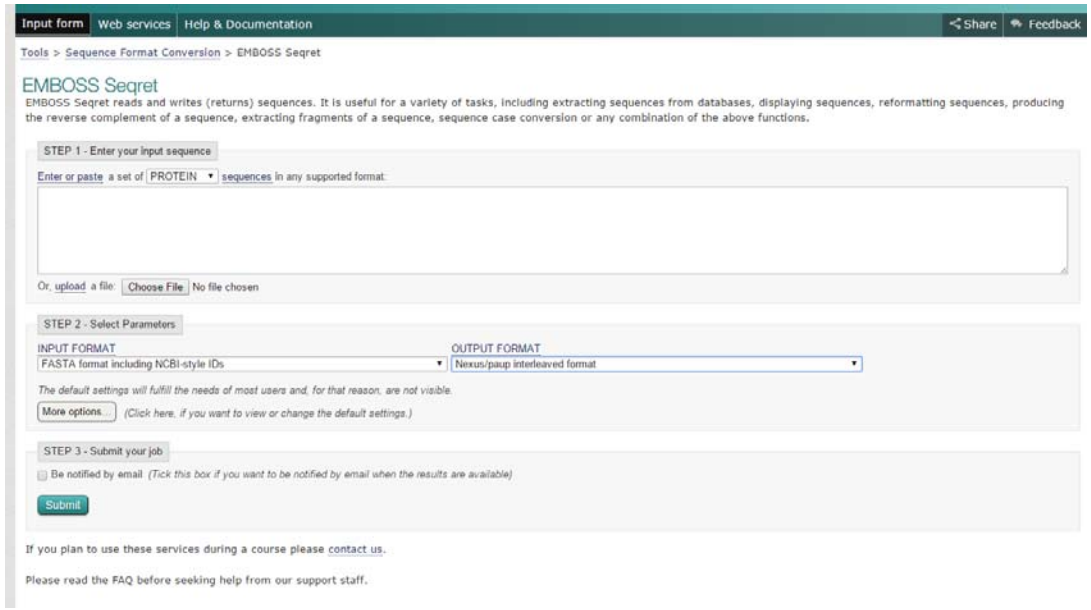
```
#NEXUS

BEGIN TAXA;
        TAXALABELS seq seq2 seq3
END;

BEGIN DATA;
        DIMENSIONS ntaxa=3 nchar=100;
        FORMAT DATATYPE=dna missing=? gap =-;
        MATRIX
        seq1    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        seq2    TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
        seq3    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        ;
END;
```

```
BEGIN TREES;
        [tree one is from miller and engstrom 2008]
        TREE tree1=(seq1,(seq2,seq3));
        [tree two is a polytomy]
        TREE tree2=(seq2,seq3,seq1);
        [tree trhee is our hyothesis]
        TREE tree3=(seq1,(seq2,seq3));
END:
```

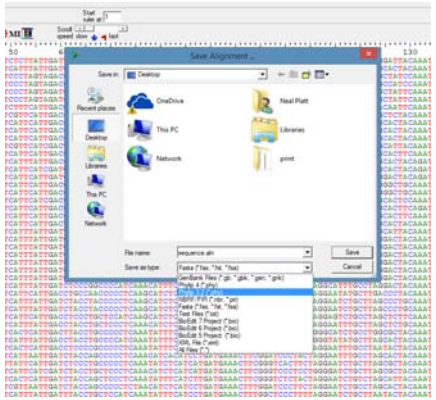
Notice the comments here.

Format Conversion

Seqret: http://www.ebi.ac.uk/Tools/sfc/emboss_seqret/



Some programs will convert formats



Making sure your data is in the right format/variant will be one of the more frustrating aspects of this class.

Other important formats

Newick

- common format used to display trees
- ((seq1,seq2),seq3)
- ((seq1:10,seq2:4):1,seq3:7)
- file extensions: nwk, newick, tree, tre

Scaleable vector graphic SVG

- image format that provides the highest resolution possible for graphics.
- Used frequently to display trees or other images in a publication quality format. (ASCII based)
- Think of zooming in on text with your iphone

Top 10 Tips for Troubleshooting

1. `qlogin`
2. Reduce your data
3. Read the error messages
4. Read the manual/README
5. Ask questions appropriately
6. Document your steps and use detailed file names
7. Google is your friend
8. `chmod a-w <importantFile>`
9. GitHub (Udacity Course)
10. Dos2Unix