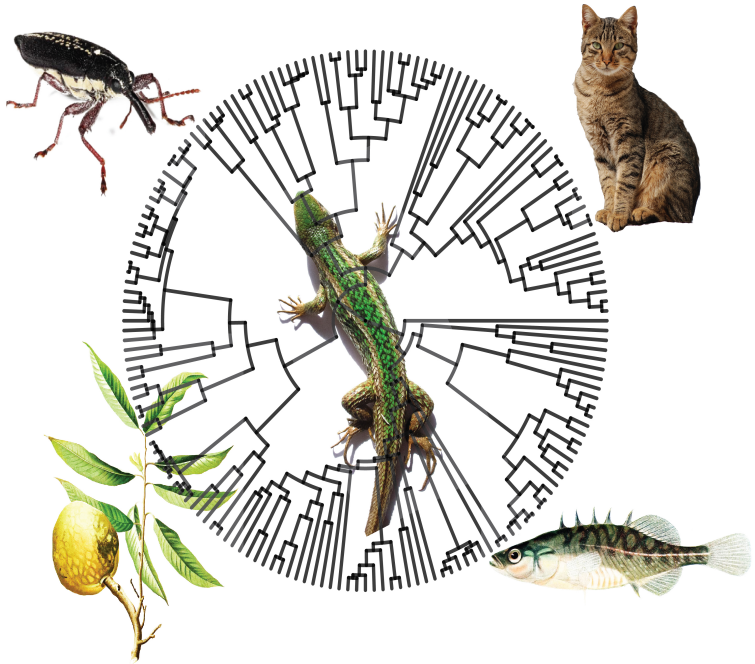
Biology 559R: Introduction to Phylogenetic Comparative Methods

Topics for this week (Jan 27 & 29):

- Statistical estimation of models of sequence evolution
- Phylogenetic inference using maximum likelihood: RAxML and Garli



Phylogenetic Inference using Likelihood: Garli and RAxML

- You can download these software for the following webpages:

Garli (command based): https://code.google.com/p/garli/

Garli GUI: http://www.bio.utexas.edu/faculty/antisense/garli/garli.html

RAxML (command based): https://github.com/stamatak/standard-RAxML

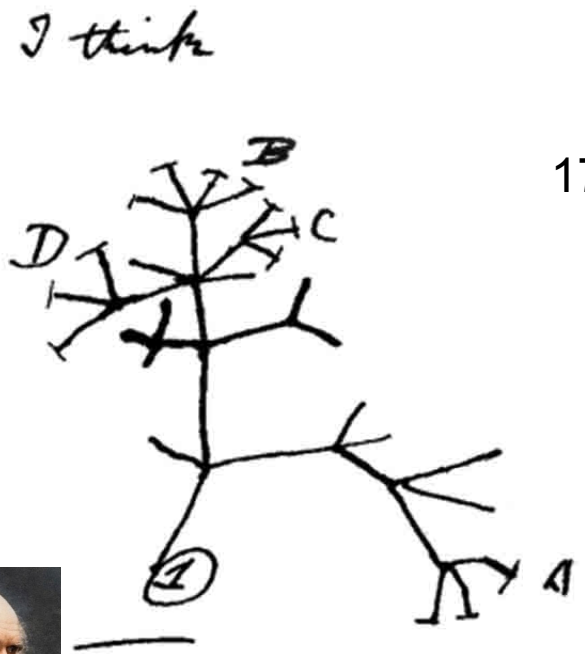
RAxML GUI (userfriendly graphical front-end for phylogenetic analyses using RAxML):
http://sourceforge.net/projects/raxmlgui/

- We also will like to visualize our trees with a user friendly software:

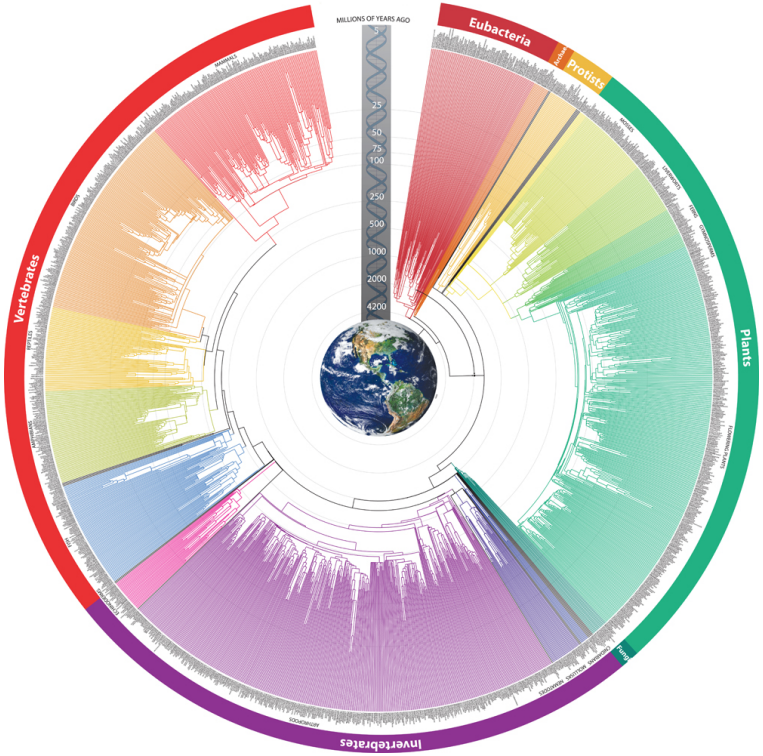
FigTree: http://tree.bio.ed.ac.uk/software/figtree/

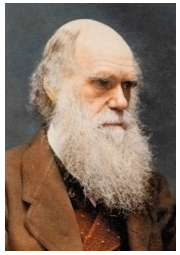
Phylogenetic Inference

• Modern biology has been built from the idea that all living organisms are related to each other by at least one common ancestor.



178 years of this idea





Darwin's notebook B (1837-1838)

Phylogenetic Inference

• The goal of phylogenetics is to reconstruct the most likely (probable) genealogical ties among biological entities (from individuals to species), including their time of divergence from a last common ancestor.

• We can also use such trees to determine the evolutionary changes and process that gave rise to the these entities and phenotypes (i.e., comparative methods).

• Tree estimation has not been a easy task and different methods have been developed to allow us to reconstruct phylogenetic trees.

Phylogenetic Inference

Among those are:

Distance based methods (e.g., Neighbor-joining to find the set of neighboring taxa that minimizes the total length of the phylogenetic tree)

Maximum parsimony (the smallest number of evolutionary changes that explain the differences among taxa under study)

Maximum likelihood (finding the optimal tree that has the highest probability given the data and model of character evolution)

Bayesian inference (finding the optimal tree given an a priori knowledge that we have about our data: alignment, model of molecular evolution, and a set of priors)

• Since the the last half of the XX century, the development of these methods required robust statistical algorithms, lots of variable characters (e.g., immunological, protein and nucleotide sequences) and powerful computational hardware.

Phylogenetic Inference: Why use molecular data?

• Most phylogenetic algorithms are build to use molecular data because:

1) DNA and protein sequences are strictly heritable units

2) The character states are can be determined unambiguously (discrete nucleotide entities) in contrast to other like morphological traits (usually continuous)

3) More amenable to computational algorithms that use discrete characters states such a nucleotide in a given position (however, new methods are trying to combine morphological, molecular, ecological, etc… types of data)

4) Molecular sequences have models of the evolution that are more regular that be inferred using general algorithms (our last class)

Phylogenetic Inference: Why use molecular data?

• Most phylogenetic algorithms are build to use molecular data because:

5) Homology (i.e., a trait derived by descent from a common ancestor) is easier to determine in molecular data than other types of data (e.g., morphology).

6) It is easier to “collect” molecular data (e.g., some conserved sequences such as ribosomal genes, genome sequencing) that is known to contain homologous characters and can be used to compare distantly related organisms (e.g., amoebas versus lizards).

7) Molecular data has become easier to collect with minimal effort (e.g., genomes, transcriptomes, proteomes) than morphological and ecological data.

Phylogenetic Inference: Tree Reliability

• A final step is to determine how reliable our estimated phylogeny. This includes:

1) Which parts of the tree that we have reconstructed are reliable?

2) Is our tree significantly better that other trees? Do we need more information to find this optimal tree?

• Usually this is accomplished using resampling methods such as bootstrap and other techniques that compare our tree against other likely solutions.

Phylogenetic Inference: Tree Reliability

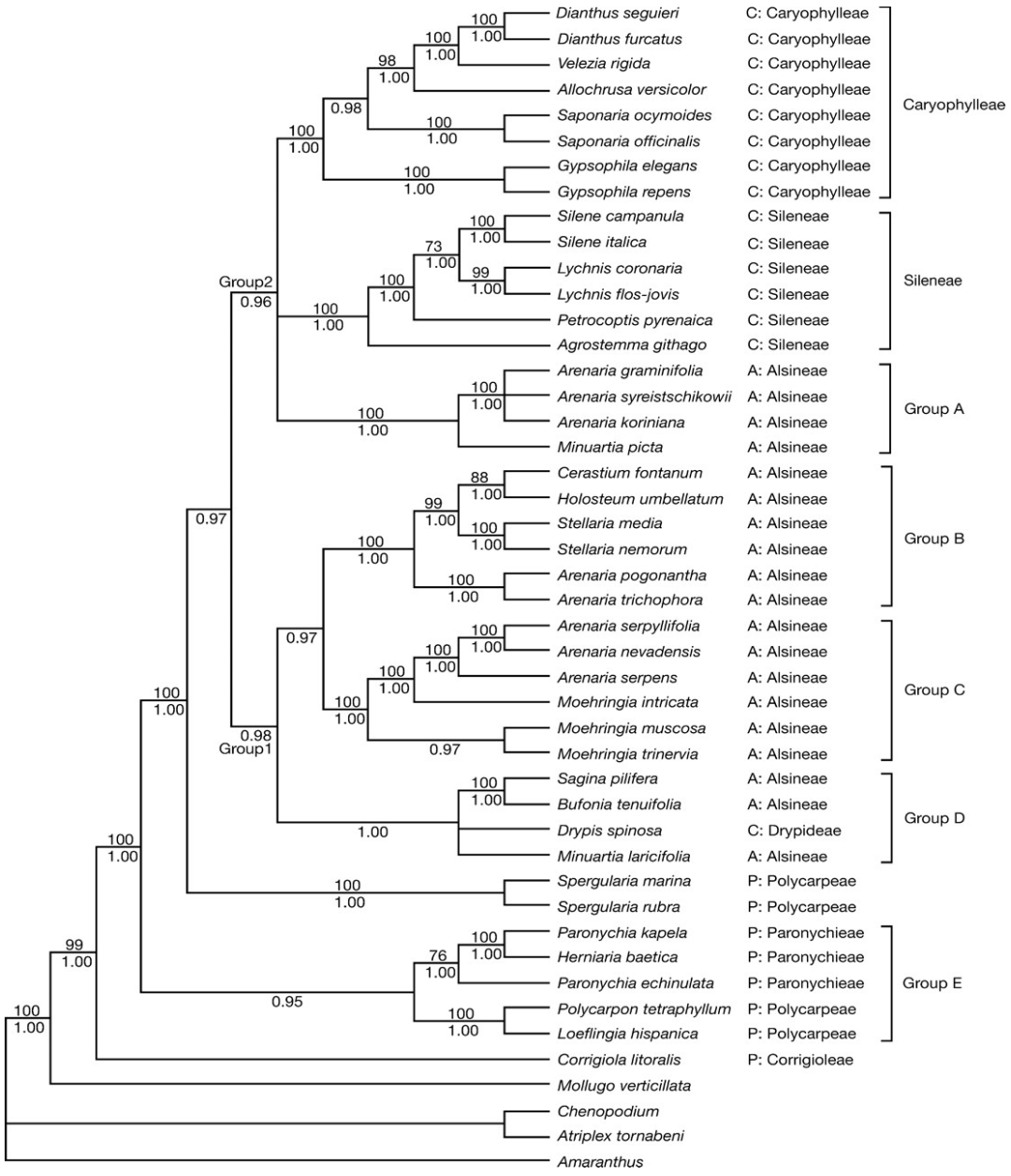
The Bootstrap: This is a technique that allows us to estimate the confidence level of particular parts of our tree (i.e., nodes).

• This methodology is based on resampling with replacement our original dataset repeatedly and re-estimate the tree. At the end, we can determine the frequency that each node (set of phylogenetic relationships) that occurs across all sampling events (e.g., 100 or more resampling events).

• Many people argue about what are a significant support and usually for ML (maximum likelihood) values, we consider:

>75% (moderate support) and >95% (excellent support)

• For Bayesian methods, this is not based on bootstrap but in posterior probabilities >0.95


Dianthus seguieri
Dianthus furcatus
Velezia rigida
Allochrusa versicolor
Saponaria ocymoides
Saponaria officinalis
Gypsophila elegans
Gypsophila repens
Silene campanula
Silene italica
Lychnis coronaria
Lychnis flos-jovis
Petrocoptis pyrenaica
Agrostemma githago
Arenaria graminifolia
Arenaria syreistschikowii
Arenaria koriniana
Minuartia picta
Cerastium fontanum
Holosteum umbellatum
Stellaria media
Stellaria nemorum
Arenaria pogonantha
Arenaria trichophora
Arenaria serpyllifolia
Arenaria nevadensis
Arenaria serpens
Moehringia intricata
Moehringia muscosa
Moehringia trinervia
Sagina pilifera
Bufonia tenuifolia
Drypis spinosa
Minuartia laricifolia
Spergularia marina
Spergularia rubra
Paronychia kapela
Herniaria baetica
Paronychia echinulata
Polycarpon tetraphyllum
Loeflingia hispanica
Corrigiola litoralis
Mollugo verticillata
Chenopodium
Atriplex tornabeni
Amaranthus
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Caryophylleae
C: Sileneae
C: Sileneae
C: Sileneae
C: Sileneae
C: Sileneae
C: Sileneae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
A: Alsineae
C: Drypideae
A: Alsineae
P: Polycarpeae
P: Polycarpeae
P: Paronychieae
P: Paronychieae
P: Paronychieae
P: Polycarpeae
P: Polycarpeae
P: Corrigioleae
Caryophylleae
Sileneae
Group A
Group B
Group C
Group D
Group E
Group1
Group2

Phylogenetic Inference: Garli GUI

• Genetic Algorithm for Rapid Likelihood Inference (GARLI) was developed by Derrick Zwickl (2006):

Zwickl, D. J., 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Ph.D. dissertation, The University of Texas at Austin.

http://www.bio.utexas.edu/faculty/antisense/garli/garli.html

From the author:

GARLI performs phylogenetic searches on aligned sequence datasets using the maximum-likelihood criterion.

It includes many new features, including the ability to perform tree inference using amino acid and codon-based models, in addition to the standard nucleotide-based models available in previous versions.

On a practical level, the program is able to perform maximum-likelihood tree searches on large datasets in a number of hours.

Phylogenetic Inference: Garli GUI

- In order to run garli in the command line we need to create and edit the config: *.conf
- Garli GUI will create a series of intermediate files, so we better create a folder our matrix:

CYTB_garli_folder

- Copy in a text editor the corresponding .nex files in the course website:

CYTB_nucleotide.nex

Phylogenetic Inference: Garli GUI

• This is file some peculiarities and it is call a simplified nexus file. Open the .nex file in a text editor a take a look at it.

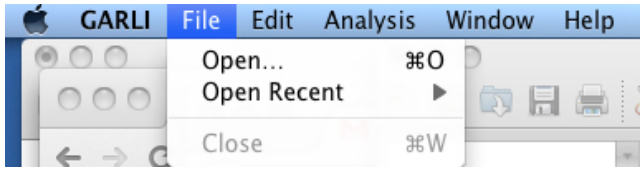
```
#NEXUS¬
¬
BEGIN◊DATA;¬
Δ   DIMENSIONS◊◊NTAX=20◊NCHAR=1141;¬
Δ   FORMAT◊DATATYPE◊=◊DNA◊GAP◊=◊-◊MISSING◊=◊?;¬
Δ   MATRIX¬
Δ   DQ316873◊◊---AC-AACATACGAAAAATCCACCCAATCCTAAAAATTATCAACAACTCATTTATTGACC
Δ   EU129908◊◊---ACAAACCTACGAAAAACTCACCCAATCCTAAAAATTATCAATAACTCATTCATCGACC
Δ   EU116519◊◊-----------ACGAAAATCCCACCAACTCCTTAAAATTATCAACAACTCTTTTATTGACC
Δ   KC853797◊◊---------------AAATCCCACCCAATCATAAAAATTGTAAACAACTCATTCATTGACC
Δ   AF020227◊◊-----------------------------------------------------ATTCATCGACC
Δ   FJ535913◊◊--------------------------ATCCTAAAAATCATTAACAACTCATTCATTGACC
Δ   EF653285◊◊-------------------------------------------------------------
Δ   HQ332321◊◊AC----AACATACGAAAAACCCACCCTGTCCTAAAAATTGTAAACAACTCATTTATCGACC
Δ   EU443141◊◊AC----AACCTACGAAAAACACATCCGATCCTAAAAATCATTAACAACTCATTCATTGATC
Δ   KJ505798◊◊AC----AACCCACGAAAAACCCACCCCATCCTAAAAATTGTAAACAACTCATTCATCGACC
Δ   GQ272812◊◊ATGACAATCACACGAAAATCCCACCCAATAATCAAAATTGTTAACAACTCATTCATCGACC
Δ   HQ141260◊◊-------------------------------------------------------------
Δ   AJ278511◊◊ATGACAGTCATACGAAAATCCCACCCAATCCTAAAAATCTTCAACAACTCATTCATCGACC
Δ   AB218883◊◊ATGACAATCCTACGAAAATCCCACCCAATCCTTAAAATAATCAACTCTTCATTCATCGACC
Δ   AB218960◊◊ATGACAATTACACGCAAATCCCACCCAATTTTCAAAATTATTAACGACTCCTTTATTGATC
Δ   AB218884◊◊ATGACAATCACACGAAAATCCCACCCGATCATCAAAATCGTAAACAACTCATTTATTGACC
Δ   AF020230◊◊-----------------------------------------------------ATTCATCGACC
Δ   KJ504993◊◊---AC-AACCTACGAAAAACCCACCCTATCCTAAAAATCGTAAACAGCTCATTCATCGACC
Δ   JF718358◊◊ATGACAAACCTACGAAAAACCCACCCACTCATAAAAATTGTCAACGACGCCTTCATTGACT
Δ   FJ535926◊◊--------------------------ATCCTAAAAATCATTAACAACTCACTCATTGACC
¬
;¬
¬
¬
END;¬
¬
```

```
----------------------------------------------------------------------------¬
CACTCTTTATTTCGCACTTTTCCTCATTTTATTACCAACCATCGCCACCTTAGAAAACAAACTCCTAAAATGATAA-¬
AATCCTCTATTTC-------------------------------------------------------------¬
AACAATATACTTCATCTTATTCCTAGTAGTTATACCAATCACAGCCACTATAGAAAATAAACTTCTCAAGTGATAA-¬
----------------------------------------------------------------------------¬
CGCCTCCTACTTCCTGCTATTCCTAATAATTATACCTATCACAGCCCTACTTGAAAACAAACTCCTTAACTGATAA-¬
AATCTTATACTTCGCACTATTTTTAATCATCATACCAATAACCTCAATACTAGAAAACAAACTGCTCAAATGATAA-¬
----------------------------------------------------------------------------¬
CACCCTCTACTTCACCATTTTCCTGATTCTTATACCCATCACAGCAATCATAGAAAACAAA----------------¬
AACCACATACTTCATTATTTTCCTAATTTTAATACCAATAACAGCAAAAATAGAAAATACACTA--------------¬
AATCATCTATTTTATATTATTCCTGATCCTAATACCAACCATATCACTACTAGAAAACAAAATACTAAAATGATAA-¬
CACCCTCTACTTCCTACTATT-----------------------------------------------------¬
----------------------------------------------------------------------------¬
CACCCTGTACTTTCTTCTATTTACCATTCTACTACCTACCACAGCAATCCTAGAAAACAAACTACTCAAGTGGTAA-¬
TGTCCTCTACTTCCTATTATTTATTATTTTTATTCCAACCACTTCAATTCTAGAAAACAAACTA--------------¬
TATCCTCTACTTCCTTTTATTTACCTTACTAATACCAGCTACCGCTATTATAGAAAATAAACT--------------¬
----------------------------------------------------------------------------¬
AACC----------------------------------------------------------------------¬
CA------------------------------------------------------------------------¬
CGCCTCCTACTTCCTGCTATTCCTAATAATTATACCTATCACAGCCATACTCGAAAACAAACTCCTTAACTGATAA-¬
```

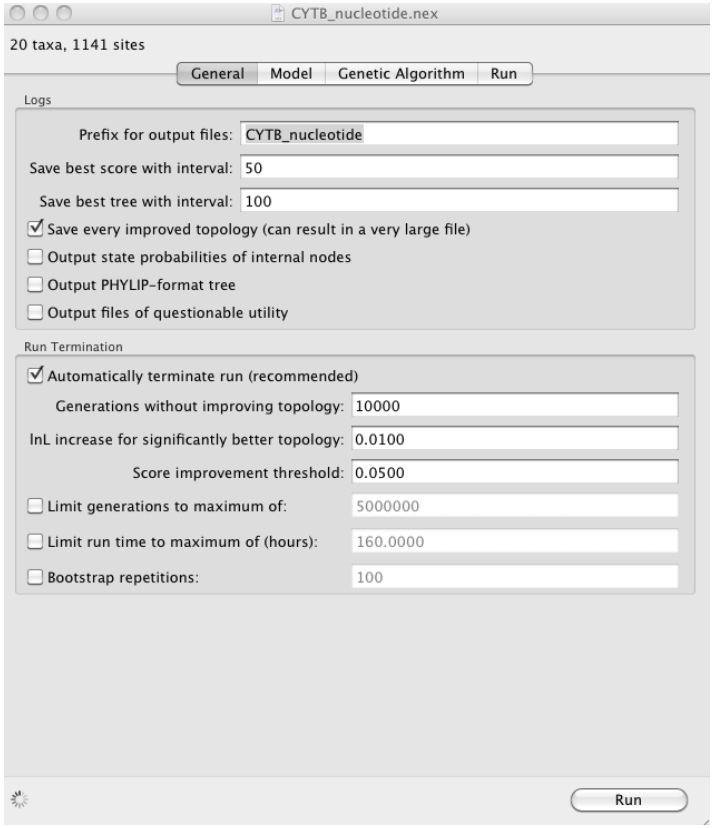
• It has no extra information other than the those above. You can save it using mesquite and should have linux line-breaks.

Phylogenetic Inference: Garli GUI

- Let's open the Garli GUI

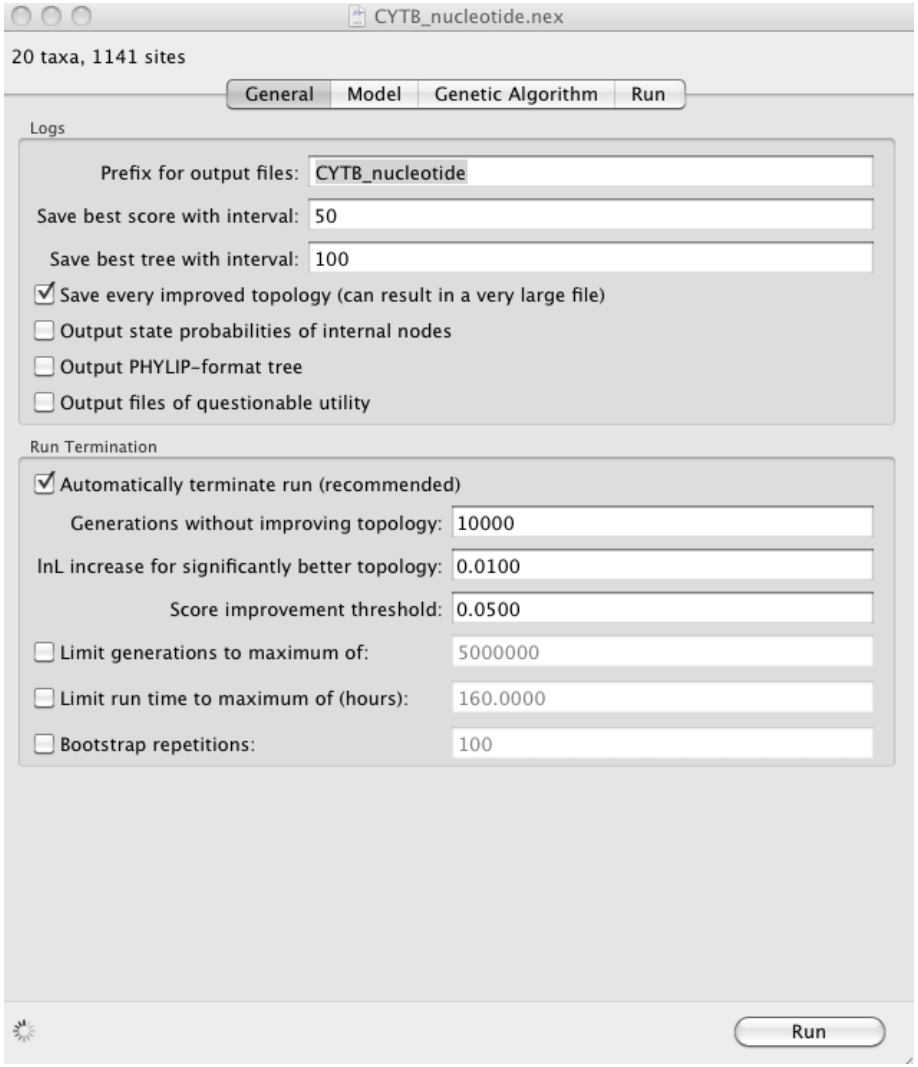


- Select our simplified .nex file and you will get the main console of garli



Phylogenetic Inference: Garli GUI

- Let's explore the console, that will help use to create the .conf file and can be used to run the Garli 2.0 more advanced version.



The general submenu:

Logs: this will save useful information as the search for the best tree progresses. I usually do not change the defaults

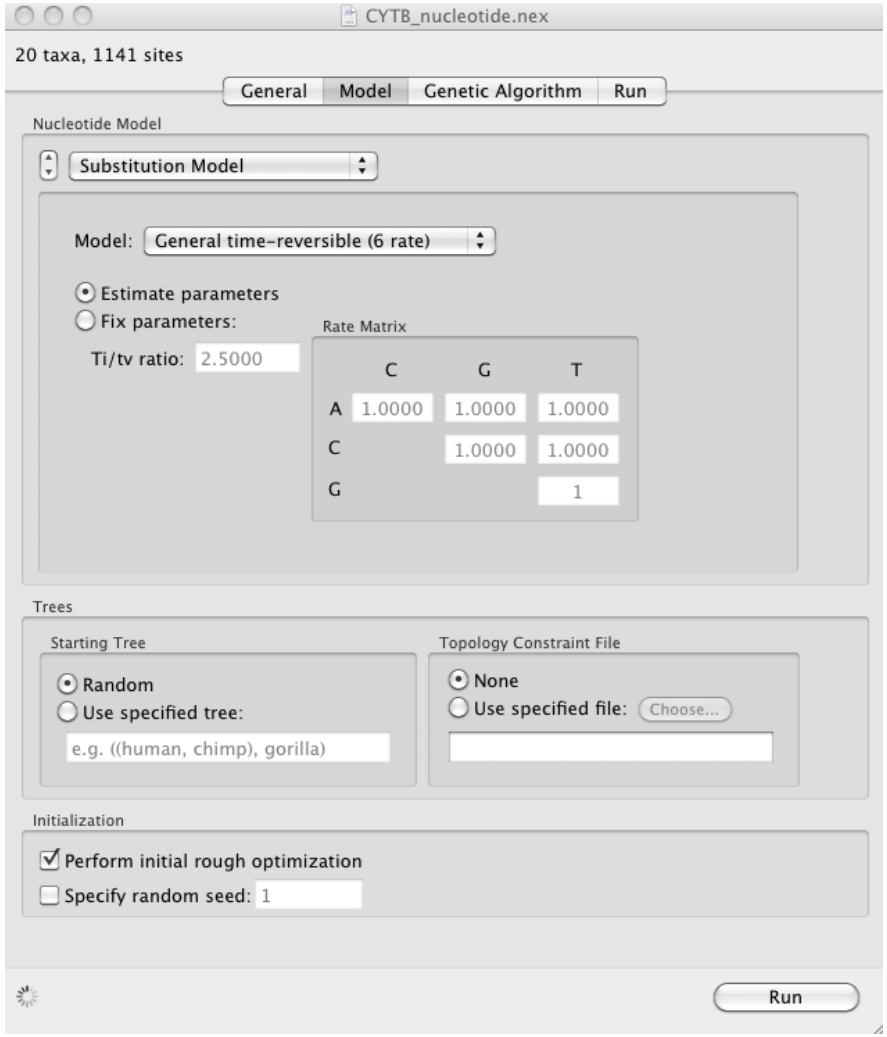
Run termination: this will determine the criteria to terminate the run (i.e., what to consider as the optimal solution).

The defaults are fine for searching the ML tree.

However, for bootstrap searches (if clicked) I will reduce the number of generation to 1000 or 2000

Phylogenetic Inference: Garli GUI

- Let's explore the console, that will help use to create the .conf file and can be used to run the Garli 2.0 more advanced version.



The model submenu:

Nucleotide model: You can input the model chosen by Model Test including:

Substitution model (GTR is the default)

Base frequencies (estimate is the default)

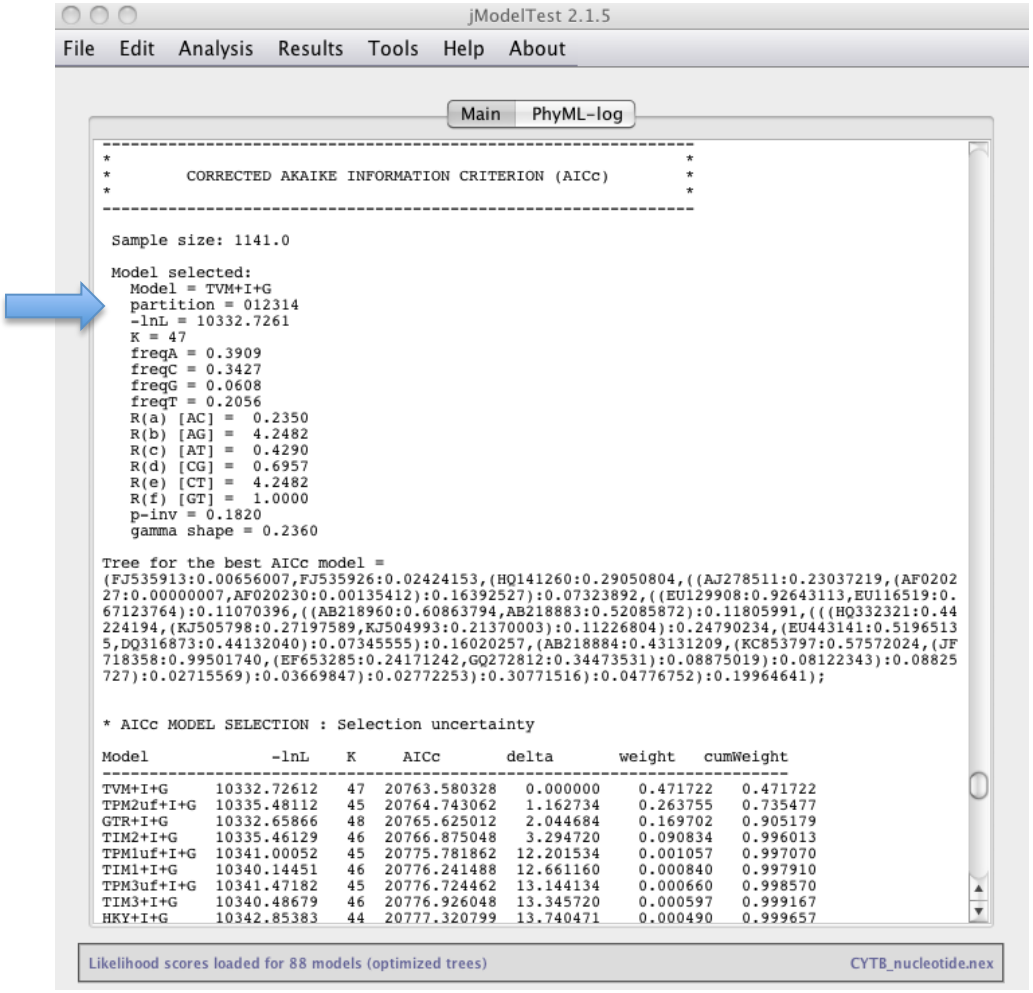
Among site rate variation (estimate invariant sites and gamma distribution are the default)

Trees: You can provide an starting tree and define constraints for certain taxa to be monophyletic

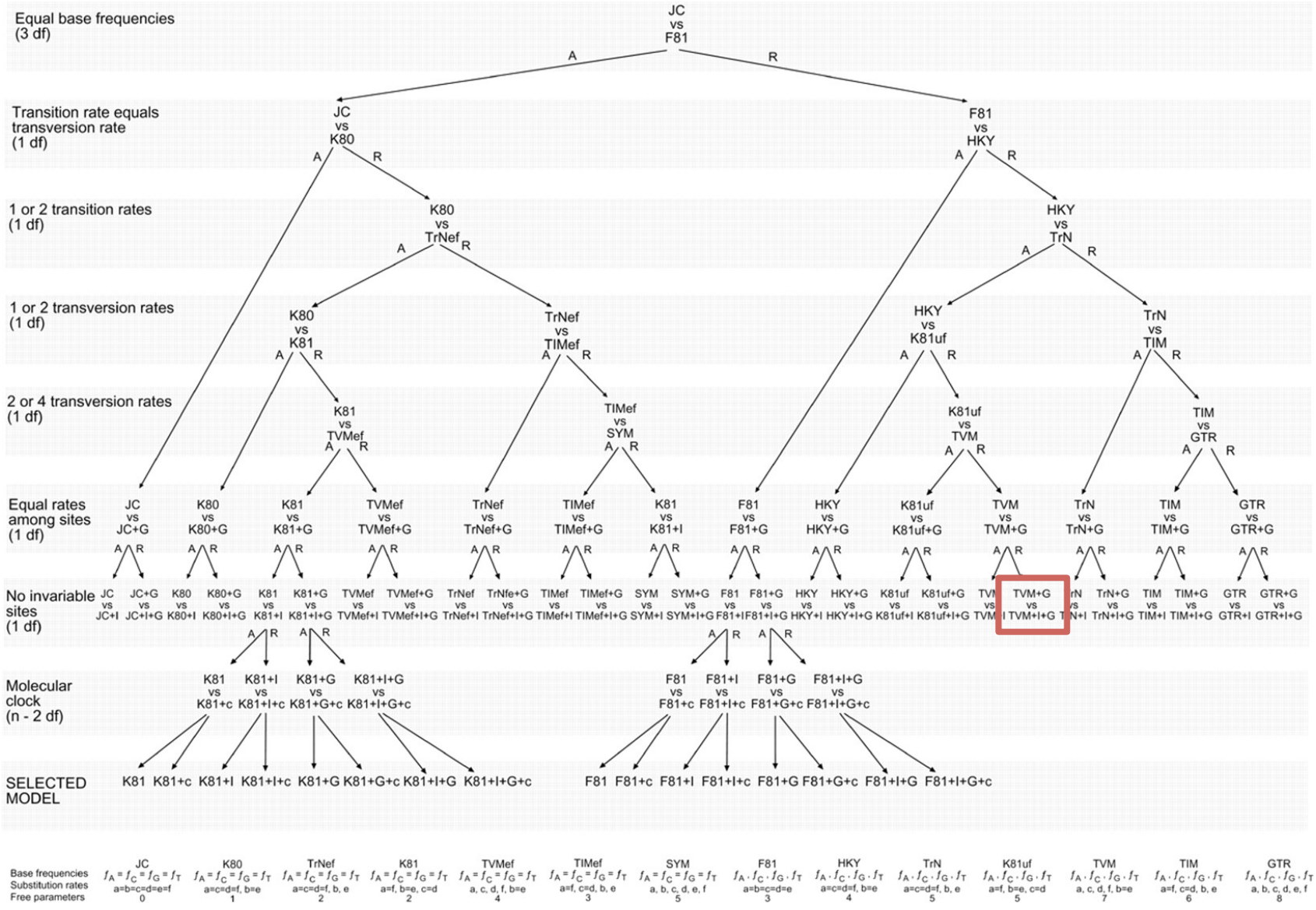
Initiation: Determine a random starting number or seed so you can repeate this search

Model Test: Running jmodeltest with our data

• You can copy the display panel in a text editor for reference and preparation of the command/parameter block necessary for the phylogenetic inference.

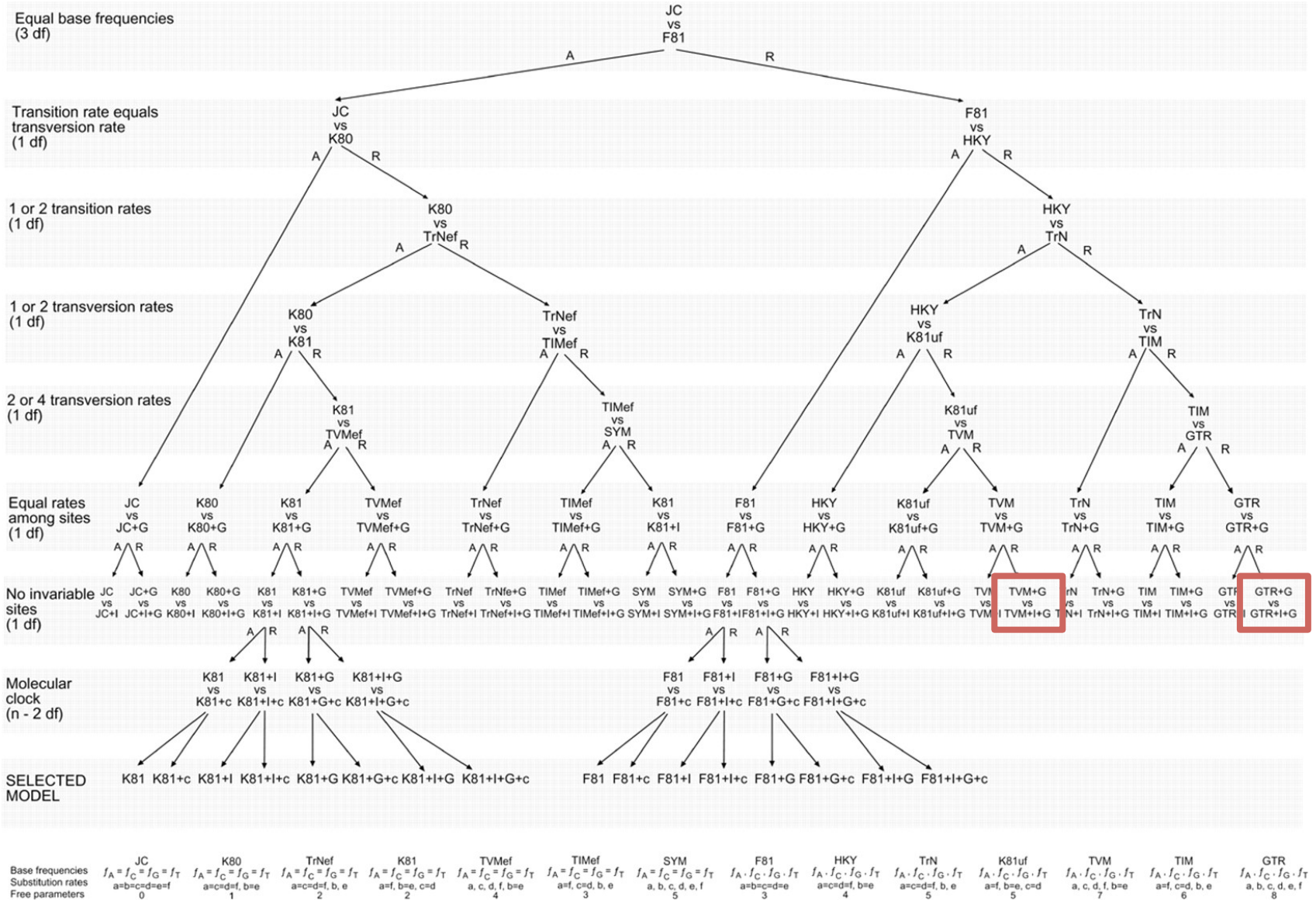


Models of Molecular Evolution: Hierarchical Order



Posada (2011)

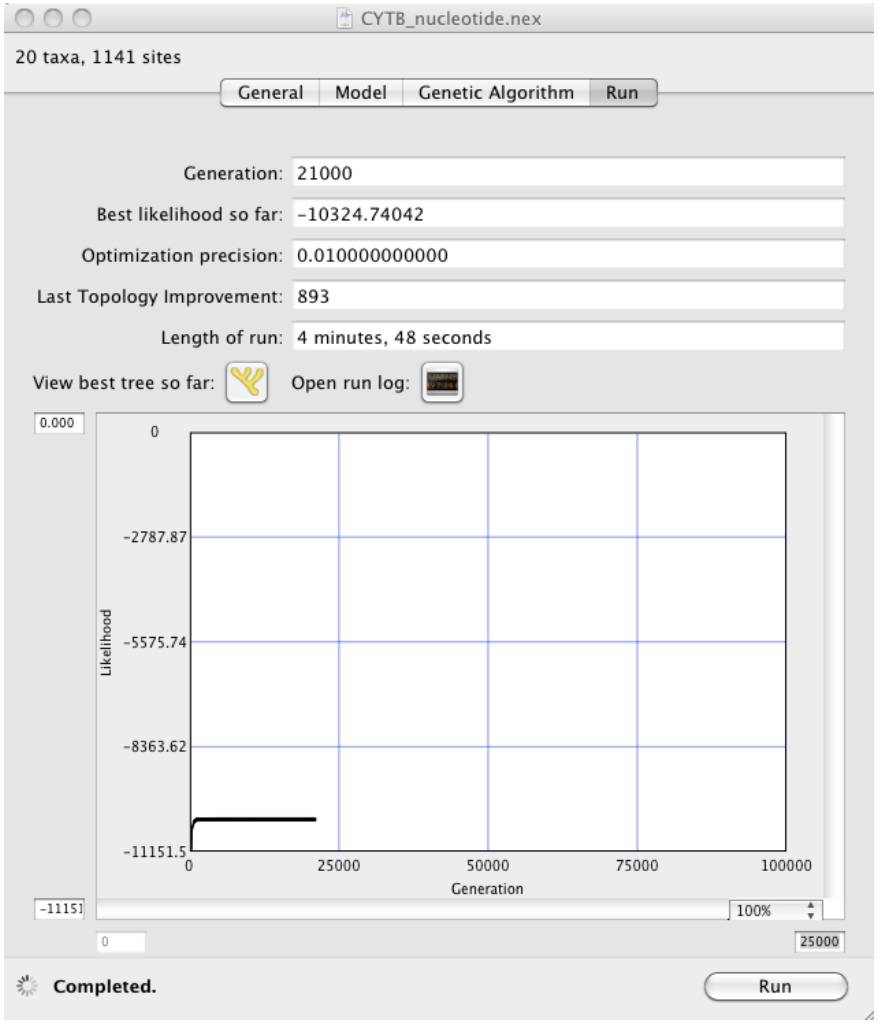
Models of Molecular Evolution: Hierarchical Order



Posada (2011)

Phylogenetic Inference: Garli GUI

- Let's explore the console, that will help use to create the .conf file and can be used to run the Garli 2.0 more advanced version.



If the file is correctly formatted, then you will see the progress of the run. In this case it took ~5 minutes

A series of intermediates files will be created and the ML tree will be written as:

CYTB_nucleotide.best.tre

We can open this tree using FigTree and take a look to our ML tree

Phylogenetic Inference: Rapid Visualization of Trees (Garli)

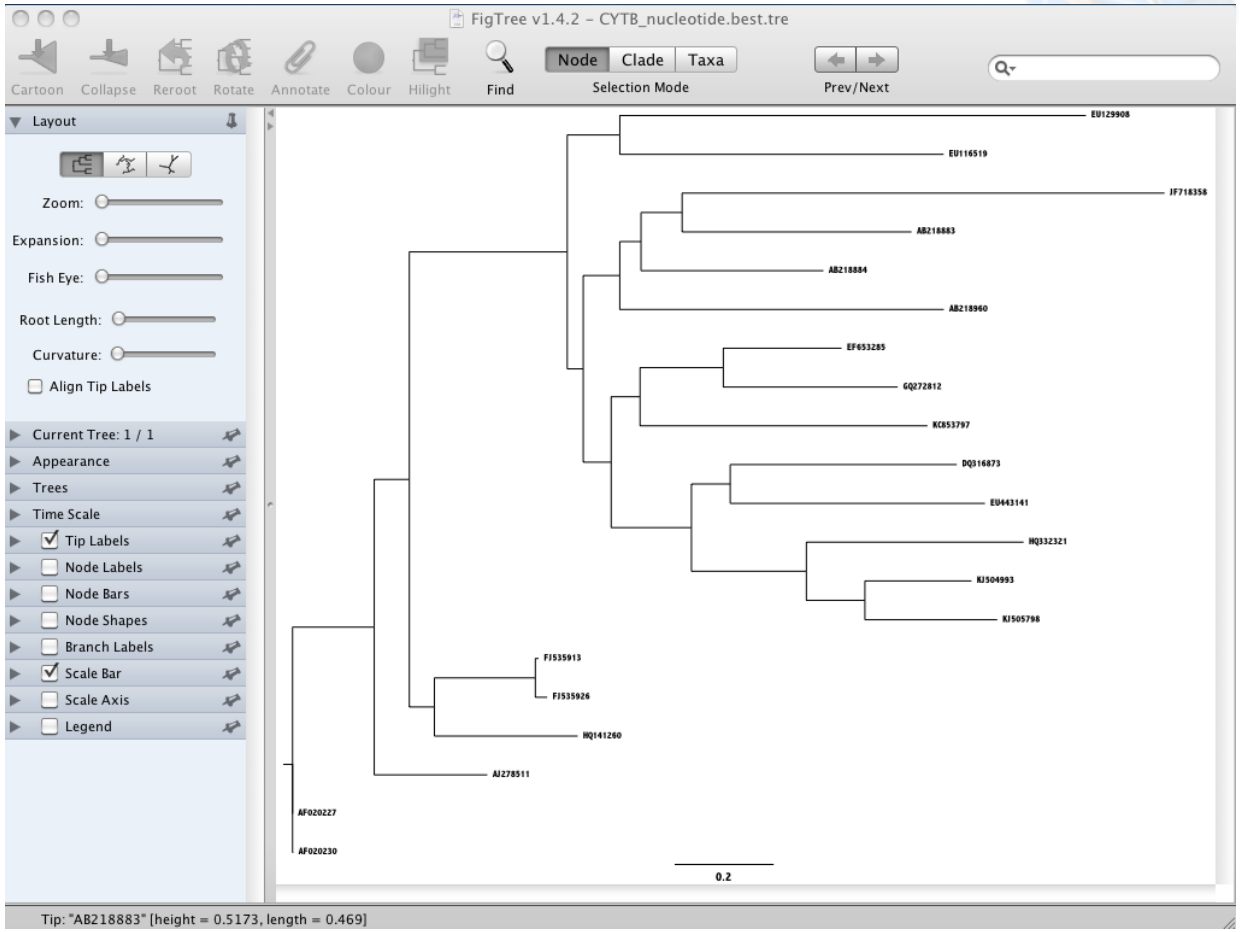
- Let's explore our ML tree using Fig Tree. You can drag and drop the tree file:

CYTB_nucleotide.best.tre

Our tree is unrooted and you can explore the different options of tree visualization.

FigTree will allow you to root, export as a pdf, and save your tree in different useful formats.

We will discuss about this software later



Phylogenetic Inference: Garli 2.0

- The current version is Garli 2.0 which is a command based software. This means that in order to run the program you will need to have a file with the commands that tell the software where your input file is and what assumptions and parameters will be used during the run.

- There is an extensive manual for how to run Garli 2.0:

https://www.nescent.org/wg_garli/FAQ

- However, most of the commands can be defined using the Garli GUI CYTB_nucleotide.conf file that was created during our previous run. To save me time, I usually run a quick Garli GUI and then modify the .conf file by renaming it as

garli.conf

• Then, I will create a folder for the corresponding file that has a Garli 2.0 compiled binary (follow the instructions in the above link).

• Garli 2.0 is much more flexible than the Garli GUI and you can have multiple markers and a mixture of morphological and nucleotide data.

Phylogenetic Inference: RAxML

- RAxML (Randomized Axelerated Maximum Likelihood) is probably the most widespread software used to infer ML phylogenies.

- The current version of RAxML is 8.1.16:

https://github.com/stamatak/standard-RAxML

- The software is flexible, extremely fast, and user friendly. Likewise, many downstream application uses RAxML as a component to get fast and reliable ML trees (remember SATe 2).

- RAxML has been used with success with large datasets, but competing software such as FastTree

http://meta.microbesonline.org/fasttree/

might be more suited for large alignments of nucleotide or protein sequences (e.g., up to a million of base pairs). The authors of FastTree claim that it is 100-1,000 times faster than RAxML 7.

Phylogenetic Inference: RAxML

- About this topic, one interesting paper to read is:

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0027731

That compare both methods (RAxML and FastTree), but overall both:

"…the relative accuracy of trees computed using FastTree and RAxML depends in part on the accuracy of the sequence alignment and dataset size, so that FastTree can be more accurate than RAxML on large datasets with relatively inaccurate alignments"

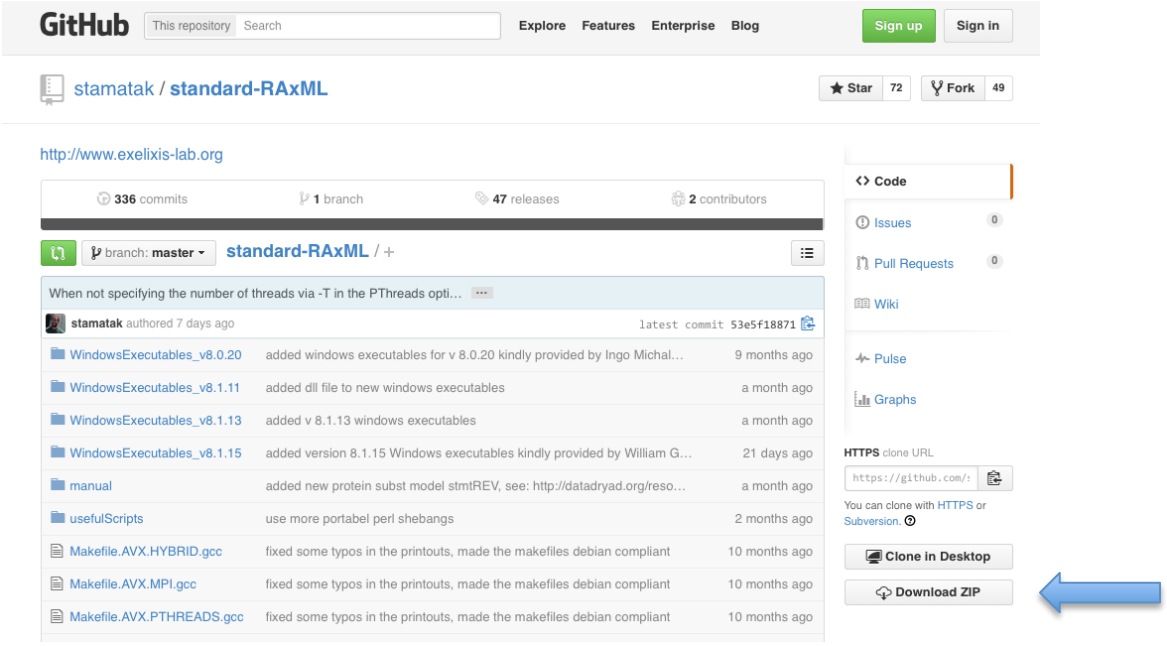
- This suggest that accurate phylogenies require good alignments, enough starting data with informative sites.

Phylogenetic Inference: RAxML

- RAxML is implement in a command-line interface, so the user has to input commands to determine the location of the input sequences and required commands to run the software.

- Like garli-2.01, you will need to compile the RAxML to the platform that your computer has. Follow the instructions in the manual (pages 3-5 of the manual)

- From the github.com page make sure that you download the entire folder



Phylogenetic Inference: RAxML GUI

- You will need to compile RAxML so follow the guidelines in the manual:

For my MAC PRO: make -f Makefile.SSE3.PTHREADS.gcc

- However, we can use a user-friendly graphical front-end for RAxML to generate the commands that can be later copy and edit for more intensive cluster analyses.

http://sourceforge.net/projects/raxmlgui/

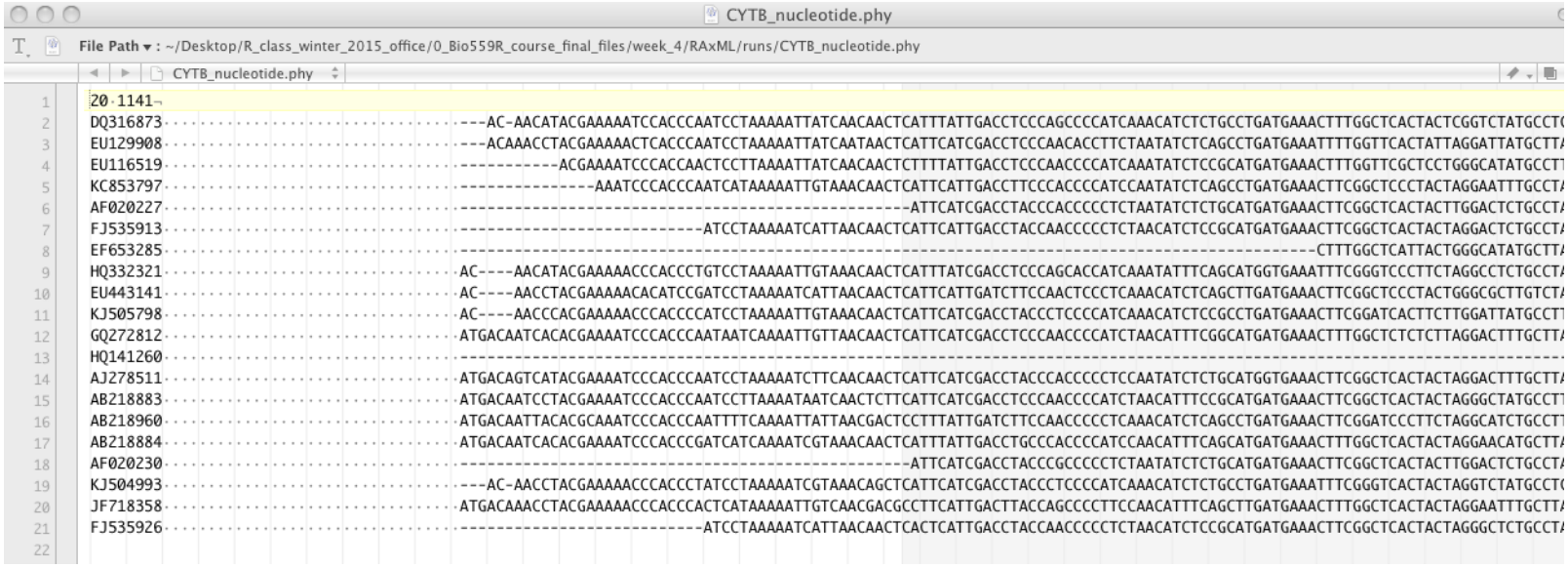
- Like garli, we need a input file with our sequences aligned and a prior knowledge of the model of molecular evolution for our sequences.

From the RAxML manual: "The input alignment format of RAxML is relaxed interleaved or sequential PHYLIP or FASTA. Relaxed means that sequence names can be of variable length between 1 up to 256 characters."

- I usually format my sequences as .phylip that was the standard RAxML input

Phylogenetic Inference: RAxML GUI

- You can save your alignment as .phylip using mesquite. Make sure that you increase the number of character for the sequence names (e.g., 100) and the line-breaks are for Linux platform.

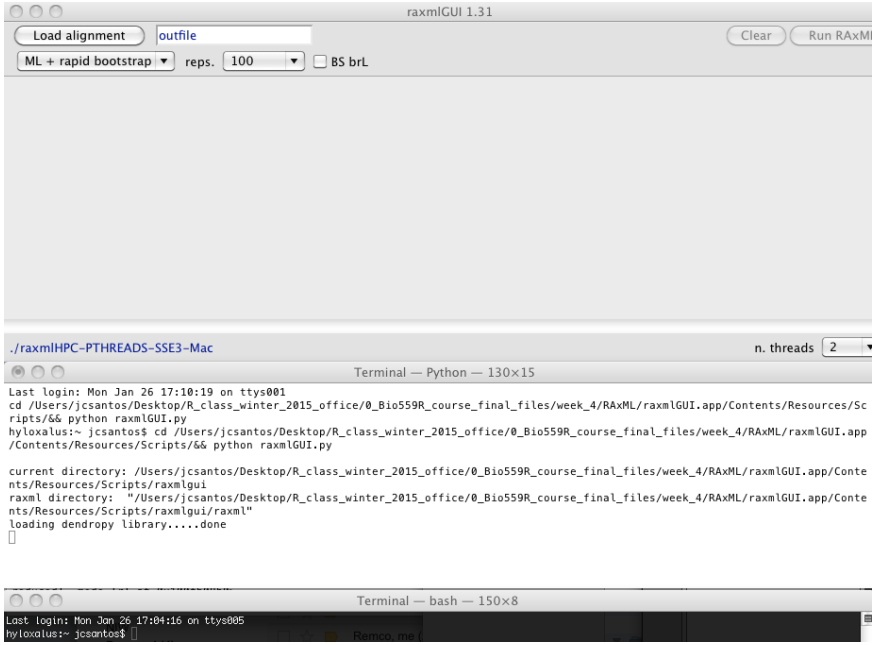


Phylogenetic Inference: RAxML GUI

• Now we can run RAxML GUI by loading our sequences. Given that this software is a python application that interacts with the RAxML executables, which are incorporated in the package.

• We will need to select input files, set the parameters and run ML analyses from the GUI console.

• I am not much fan of these easier point-click interphases, but it helps to rapidly create the set of command-line arguments that you can modify for further refinement.



Phylogenetic Inference: RAxML GUI

- Load your alignments and if some are duplicate, indicate to preserve those.

raxmlGUI 1.31

Add alignment | CYTB_nucleotide_red | Outgroup | <none> | Clear | Run RAxML

ML + rapid bootstrap | reps. | 100 | BS brL | GTRGAM...

```
20 1140
DQ316873 ---AC-AACATACGAAAAATCCCACCCAATCCTAAAAATTATCAACAACTCATTTATTGACCTCCCAGCCCCATCAAACATCTCTGCCTGATGAAACTTTGGCTCACTACTCGGTCTATGCCTCATCTCCCAAATTGTAACAGGA
EU129908 ---ACAAACCTACGAAAAACTCACCCAATCCTAAAAATTATCAATAACTCATTCATCGACCTCCCAACACCTTCTAATATCTCAGCCTGATGAAATTTTGGTTCACTATTAGGATTATGCTTACTTATTCAAATCCTCACAGGA
EU116519 ----------ACGAAAATCCCACCAACTCCTTAAAATTATCAACAACTCTTTTATTGACCTCCCAACCCCATCAAATATCTCCGCATGATGAAACTTTGGTTCGCTCCTGGGCATATGCCTTATTATTCAAATCTTAACCGGA
KC853797 ----------AAATCCCACCCAATCATAAAAATTGTAAACAACTCATTCATTGACCTTCCCACCCCATCCAATATCTCAGCCTGATGAAACTTCGGCTCCCTACTAGGAATTTGCCTAATTATCCAAATCTTAACAGGC
AF020227 ----------ATTCATCGACCTACCCACCCCCTCTAATATCTCTGCATGATGAAACTTCGGCTCACTACTTGGACTCTGCCTAATTATCCAAATCCTAACAGGC
FJ535913 ----------ATCCTAAAAATCATTAACAACTCATTCATTGACCTACCAACCCCCTCTAACATCTCCGCATGATGAAACTTCGGCTCACTACTAGGACTCTGCCTAATTATCCAAATCATAACAGGC
EF653285 ----------CTTTGGCTCATTACTGGGCATATGCTTAATTATTCAAGTATTAACCGGC
HQ332321 AC----AACATACGAAAAACCCACCCTGTCCTAAAAATTGTAAACAACTCATTTATCGACCTCCCAGCACCATCAAATATTTCAGCATGGTGAAATTTCGGGTCCCTTCTAGGCCTCTGCCTAATCGTACAAGTACTAACAGGC
EU443141 AC----AACCTACGAAAAACACATCCGATCCTAAAAATCATTAACAACTCATTCATTGATCTTCCAACTCCCTCAAACATCTCAGCTTGATGAAACTTCGGCTCCCTACTGGGCGCTTGTCTAATCGCACAAGTACTTACCGGC
KJ505798 AC----AACCCACGAAAAACCCACCCCATCCTAAAAATTGTAAACAACTCATTCATCGACCTACCCTCCCCATCAAACATCTCCGCCTGATGAAACTTCGGATCACTTCTTGGATTATGCCTTATCCTCCAAGTACTAACAGGA
GQ272812 ATGACAATCACACGAAAATCCCACCCAATAATCAAAATTGTTAACAACTCATTCATCGACCTCCCAACCCCATCTAACATTTCGGCATGATGAAACTTTGGCTCTCTCTTAGGACTTTGCTTAATCATTCAAATCCTAACAGGA
HQ141260 ----------GGC
AJ278511 ATGACAGTCATACGAAAATCCCACCCAATCCTAAAAATCTTCAACAACTCATTCATCGACCTACCCACCCCCTCCAATATCTCTGCATGGTGAAACTTCGGCTCACTACTAGGACTTTGCTTAATCGTACAAACACTAACAGGC
AB218883 ATGACAATCCTACGAAAATCCCACCCAATCCTTAAAATAATCAACTCTTCATTCATCGACCTCCCAACCCCATCTAACATTTCCGCATGATGAAACTTCGGCTCACTACTAGGGCTATGCCTTATTATCCAAATTACCACCGGC
AB218960 ATGACAATTACACGCAAATCCCACCCAATTTTCAAAATTATTAACGACTCCTTTATTGATCTTCCAACCCCCTCAAACATCTCAGCCTGATGAAACTTCGGATCCCTTCTAGGCATCTGCCTTATTATTCAAATCTTAACAGGC
AB218884 ATGACAATCACACGAAAATCCCACCCGATCATCAAAATCGTAAACAACTCATTTATTGACCTGCCCACCCCATCCAACATTTCAGCATGATGAAACTTTGGCTCACTACTAGGAACATGCTTAATCATCCAAATTCTAACCGGA
AF020230 ----------ATTCATCGACCTACCCGCCCCCTCTAATATCTCTGCATGATGAAACTTCGGCTCACTACTTGGACTCTGCCTAATTATCCAAATCCTAACAGGC
```

Sequence file > CYTB_nucleotide.phy.reduced (20 taxa, 1140 characters, DNA) | n. threads | 2

Terminal — Python — 133×19

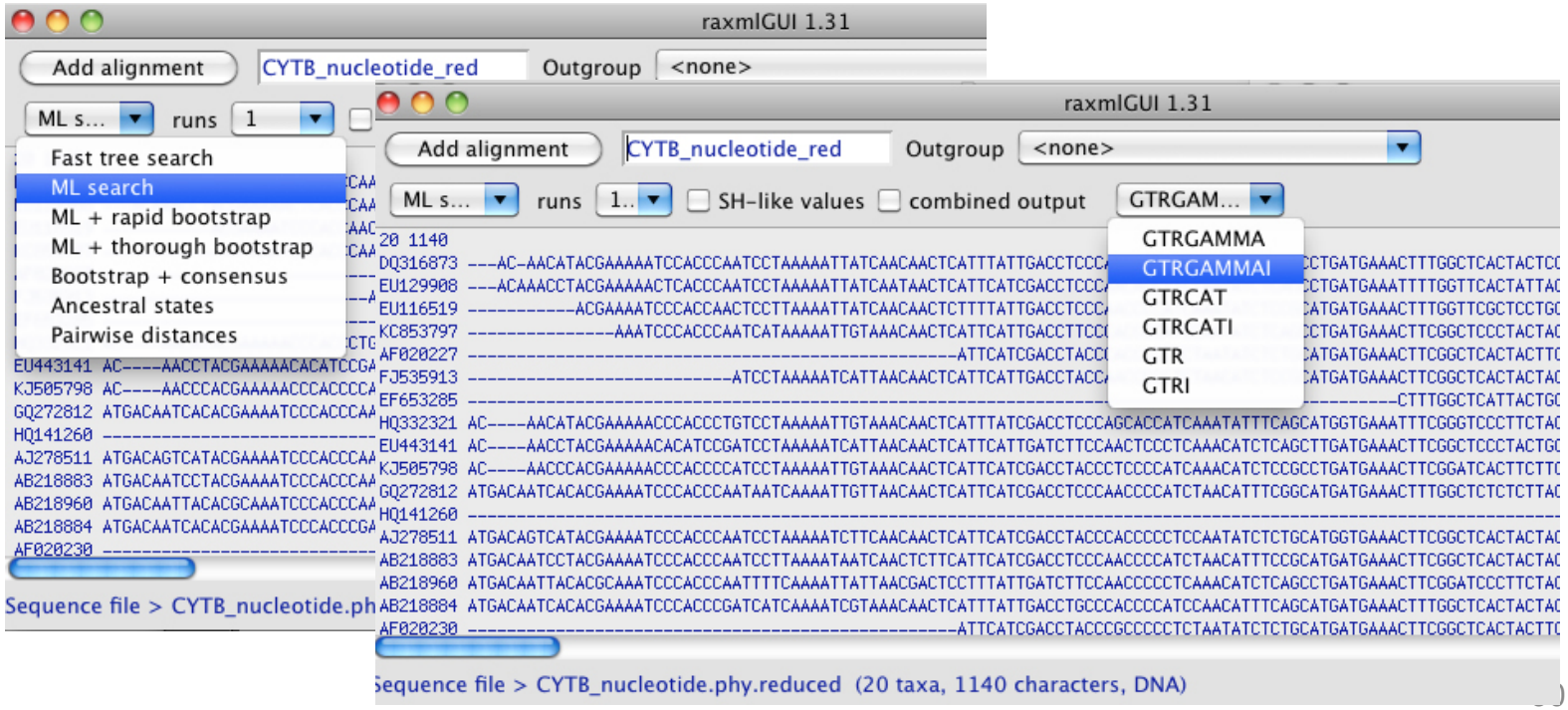
```
loading dendropy library.....done
1
cd  "/Users/jcsantos/Desktop/R_class_winter_2015_office/0_Bio559R_course_final_files/week_4/RAxML/raxmlGUI.app/Contents/Resources/Scr
ipts/raxmlgui/raxml" &&./raxmlHPC-PTHREADS-SSE3-Mac -T 2 -f c -m GTRGAMMA -s  "/Users/jcsantos/Desktop/R_class_winter_2015_office/0_B
io559R_course_final_files/week_4/RAxML/runs/CYTB_nucleotide.phy"  -n CYTB_nucleotide_red -w  "/Users/jcsantos/Desktop/R_class_winter_
2015_office/0_Bio559R_course_final_files/week_4/RAxML/runs/"  -O
IMPORTANT WARNING: Alignment column 1141 contains only undetermined values which will be treated as missing data

IMPORTANT WARNING
Found 1 column that contains only undetermined values which will be treated as missing data.
Normally these columns should be excluded from the analysis.

Just in case you might need it, an alignment file with
undetermined columns removed is printed to file /Users/jcsantos/Desktop/R_class_winter_2015_office/0_Bio559R_course_final_files/week_
4/RAxML/runs/CYTB_nucleotide.phy.reduced
Alignment format can be read by RAxML
<open file '/Users/jcsantos/Desktop/R_class_winter_2015_office/0_Bio559R_course_final_files/week_4/RAxML/runs/CYTB_nucleotide.phy.red
uced', mode 'r' at 0x100fb08b0>
```

Phylogenetic Inference: RAxML GUI

• RAxML implements complex (GTR-based) models of nucleotide substitution. The rational is that the GTR model is the most common and general one for real-world DNA analysis.

Thus, it is better to efficiently implement and optimize this model instead of offering a plethora of distinct models which are only special cases of GTR.

• So, we have less parameters to select before we set our run.



Phylogenetic Inference: RAxML GUI

- After the run, you will get your ML trees or tree (if you selected only one repetition).

- You can now copy and past the commands that were generated from the RAxML GUI to perform this initial run

```
This is the RAxML Master Pthread

This is RAxML Worker Pthread Number: 1


This is RAxML version 7.4.2 released by Alexandros Stamatakis on November 23 2012.

With greatly appreciated code contributions by:
Andre Aberer (HITS)
Simon Berger (HITS)
Nick Pattengale (Sandia)
Wayne Pfeiffer (SDSC)
Akifumi S. Tanabe (Univ. Tsukuba)


Alignment has 611 distinct alignment patterns

Proportion of gaps and completely undetermined characters in this alignment: 6.20%

RAxML rapid hill-climbing mode

Using 1 distinct models/data partitions with joint branch length optimization


Executing 10 inferences on the original alignment using 10 distinct randomized MP trees

All free model parameters will be estimated by RAxML
GAMMA model of rate heteorgeneity, ML estimate of alpha-parameter

GAMMA Model parameters will be estimated up to an accuracy of 0.1000000000 Log Likelihood units

Partition: 0
Alignment Patterns: 611
Name: No Name Provided
DataType: DNA
Substitution Matrix: GTR



RAxML was called as follows:

./raxmlHPC-PTHREADS-SSE3-Mac -T 2 -f d -m GTRGAMMA -N 10 -O -p 153 -s /Users/jcsantos/Desktop/R_
class_winter_2015_office/0_Bio559R_course_final_files/week_4/RAxML/runs/CYTB_nucleotide.phy.redu
ced -n CYTB_nucleotide_red.tre -w /Users/jcsantos/Desktop/R_class_winter_2015_office/0_Bio559R_c
ourse_final_files/week_4/RAxML/runs/


Partition: 0 with name: No Name Provided
Base frequencies: 0.296 0.305 0.127 0.272

Testing which likelihood implementation to use
Standard Implementation full tree traversal time: 0.003514
Subtree Equality Vectors for gap columns full tree traversal time: 0.003678
... using standard implementation

Inference[0]: Time 3.829743 GAMMA-based likelihood -10414.823772, best rearrangement setting 5
Inference[1]: Time 5.611086 GAMMA-based likelihood -10412.037806, best rearrangement setting 5
Inference[2]: Time 3.861977 GAMMA-based likelihood -10412.008328, best rearrangement setting 5
Inference[3]: Time 6.571734 GAMMA-based likelihood -10412.231655, best rearrangement setting 5
Inference[4]: Time 3.836976 GAMMA-based likelihood -10414.153198, best rearrangement setting 5
Inference[5]: Time 3.792474 GAMMA-based likelihood -10412.116127, best rearrangement setting 5
Inference[6]: Time 3.819714 GAMMA-based likelihood -10412.009619, best rearrangement setting 5
Inference[7]: Time 3.884528 GAMMA-based likelihood -10412.029583, best rearrangement setting 5
Inference[8]: Time 3.823763 GAMMA-based likelihood -10414.156901, best rearrangement setting 5
Inference[9]: Time 3.811830 GAMMA-based likelihood -10412.032449, best rearrangement setting 5
```

You can read about each of these commands in the RAxML manual for further information

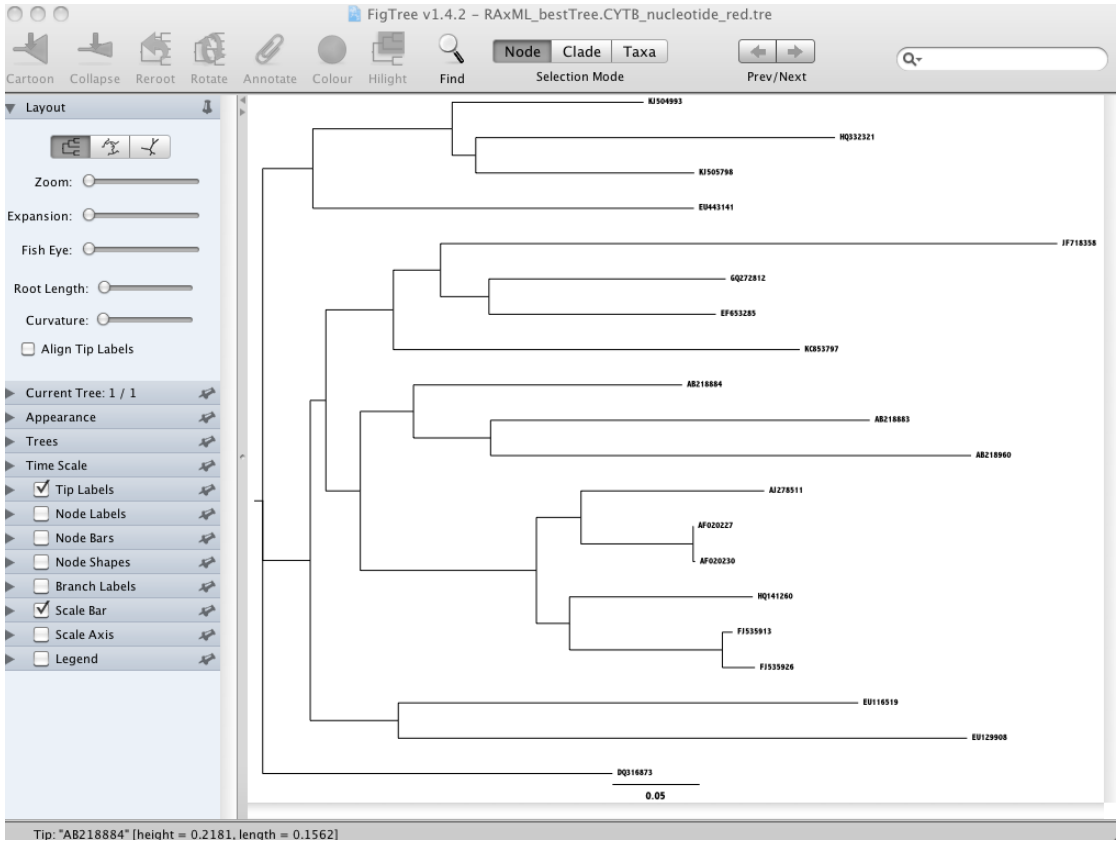
Phylogenetic Inference: Rapid Visualization of Trees (RAxML)

- Like after our garli run, we can explore our best ML tree using FigTree. You can drag and drop the tree file: RAxML_bestTree.CYTB_nucleotide_red.tre

Our tree is unrooted and you can explore the different options of tree visualization.

FigTree will allow you to root, export as a pdf, and save your tree in different useful formats.

We will discuss about this software later



Time Calibrated Trees: BEAST

- You can download these software for the following webpages:

BEAST: http://beast.bio.ed.ac.uk/

r8s: http://loco.biosci.arizona.edu/r8s/

- We also will like to visualize our trees with a user friendly software:

FigTree: http://tree.bio.ed.ac.uk/software/figtree/