MCB 5472

Supertrees vs Supermatrix
Assembly of Gene Families

Peter Gogarten
Office: *BSP 404*
phone: *860 486-4061,*
Email: *gogarten@uconn.edu*

Next Monday: Class meets in 201

Lab this Wednesday:
dN/dS, or assembly of gene families, or …

Presentations (less than 12 minutes each)

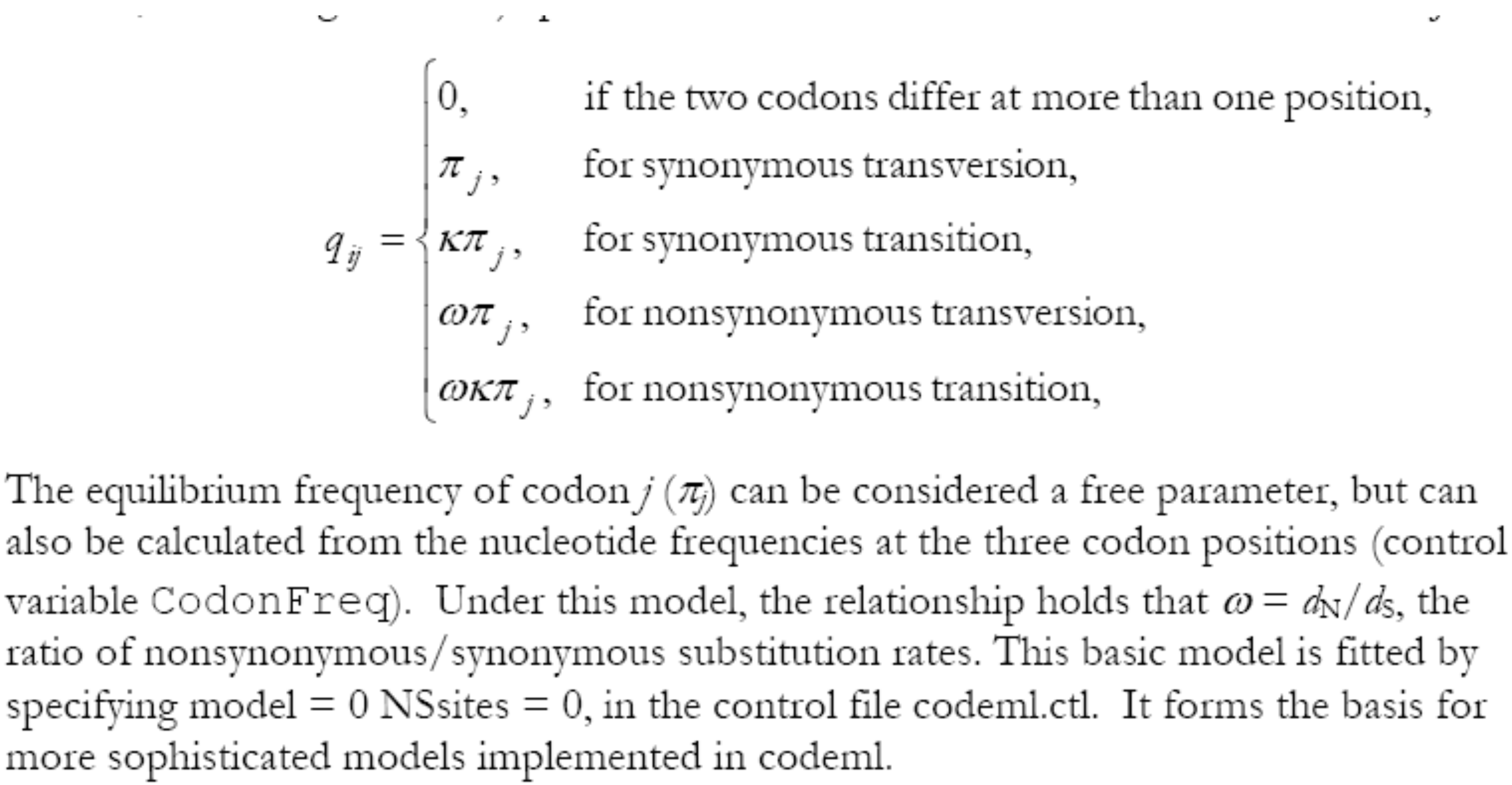
Monday 4/23:

- Shannon Soucy, Ajay Obla, Terrence Shin,
- Allison Kerwin, Jacquelynn Benjamino, Matthew Fullmer

Wednesday 4/25:

- Ursula King, Erin Duffy, Kunica Asija, Corey Bunce,
- Matt Ouellette, Emre Aksoy, Seila Omer,

PAML (codeml) the basic model



$$q_{ij} = \begin{cases} 0, & \text{if the two codons differ at more than one position,} \\ \pi_j, & \text{for synonymous transversion,} \\ \kappa\pi_j, & \text{for synonymous transition,} \\ \omega\pi_j, & \text{for nonsynonymous transversion,} \\ \omega\kappa\pi_j, & \text{for nonsynonymous transition,} \end{cases}$$

The equilibrium frequency of codon *j* (π_j) can be considered a free parameter, but can also be calculated from the nucleotide frequencies at the three codon positions (control variable `CodonFreq`). Under this model, the relationship holds that $\omega = d_N/d_S$, the ratio of nonsynonymous/synonymous substitution rates. This basic model is fitted by specifying model = 0 NSsites = 0, in the control file codeml.ctl. It forms the basis for more sophisticated models implemented in codeml.

Paml is available from the author at
http://abacus.gene.ucl.ac.uk/software/paml.html

sites versus branches

You can determine omega for the whole dataset; however, usually not all sites in a sequence are under selection all the time.

PAML (and other programs) allow to either determine omega for each site over the whole tree, *Branch Models* **,**
or determine omega for each branch for the whole sequence,
Site Models **.**

It would be great to do both, i.e., conclude codon 176 in the vacuolar ATPases was under positive selection during the evolution of modern humans – alas, a single site does not provide any statistics ….

PAML – codeml – sites model (cont.)

the program is invoked by typing codeml followed by the name of a control file that tells the program what to do.

paml can be used to find the maximum likelihood tree, however, the program is rather slow. Phyml is a better choice to find the tree, which then can be used as a user tree.

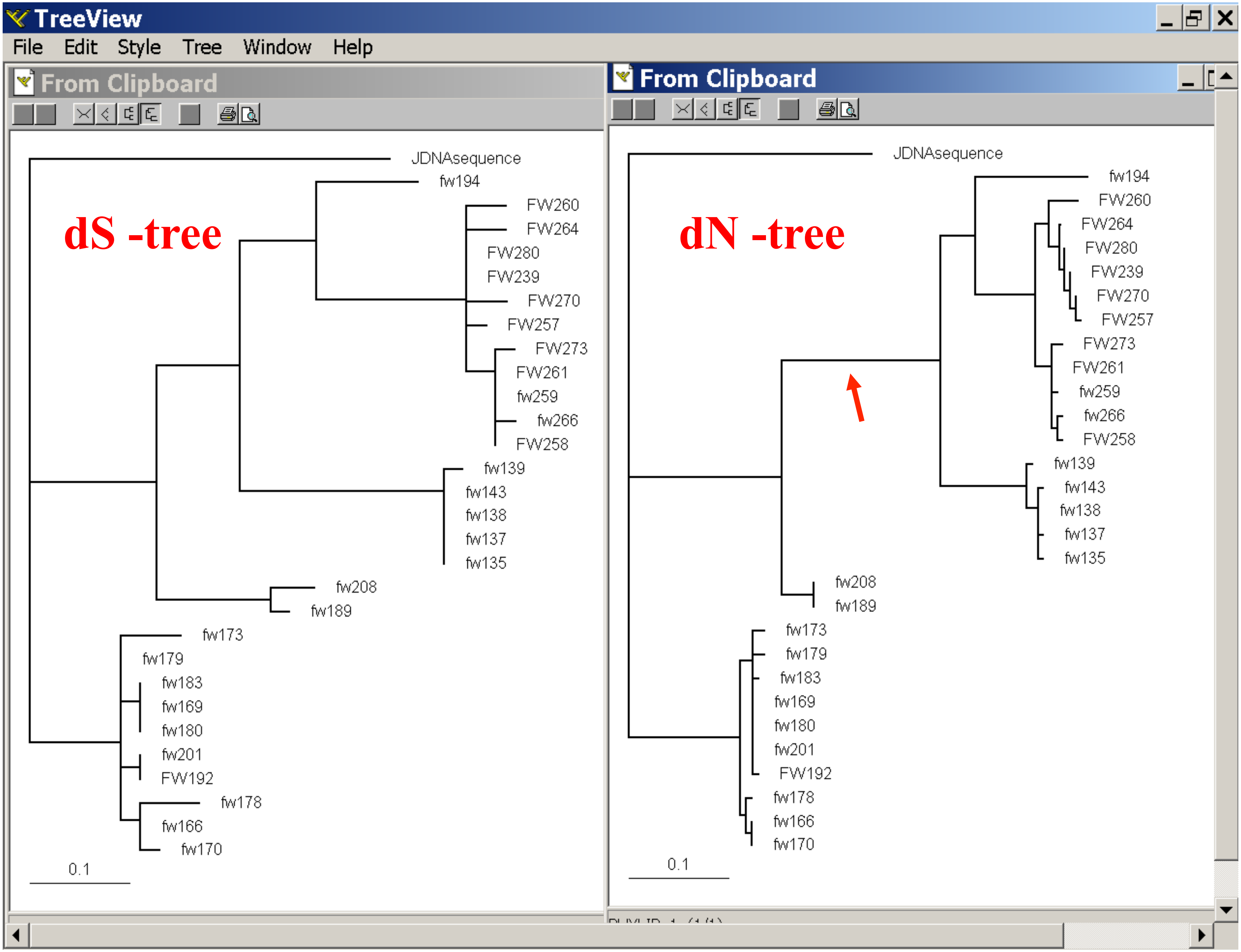
An example for a codeml.ctl file is codeml.hv1.sites.ctl
This file directs codeml to run three different models:
one with an omega fixed at 1, a second where each site can be either have an omega between 0 and 1, or an omega of 1, and third a model that uses three omegas as described before for MrBayes.
The output is written into a file called Hv1.sites.codeml_out (as directed by the control file).

Point out log likelihoods and estimated parameter line (kappa and omegas)

Additional useful information is in the rst file generated by the codeml

Discuss overall result.

PAML – codeml – branch model



where to get help

read the manuals and help files
check out the discussion board at
https://www.ucl.ac.uk/discussions/viewforum.php?f=54
pal2nal: http://www.bork.embl.de/pal2nal/

else

there is a new program on the block called hy-phy (=hypothesis testing using phylogenetics).

The easiest is probably to run the analyses on the authors datamonkey.

→

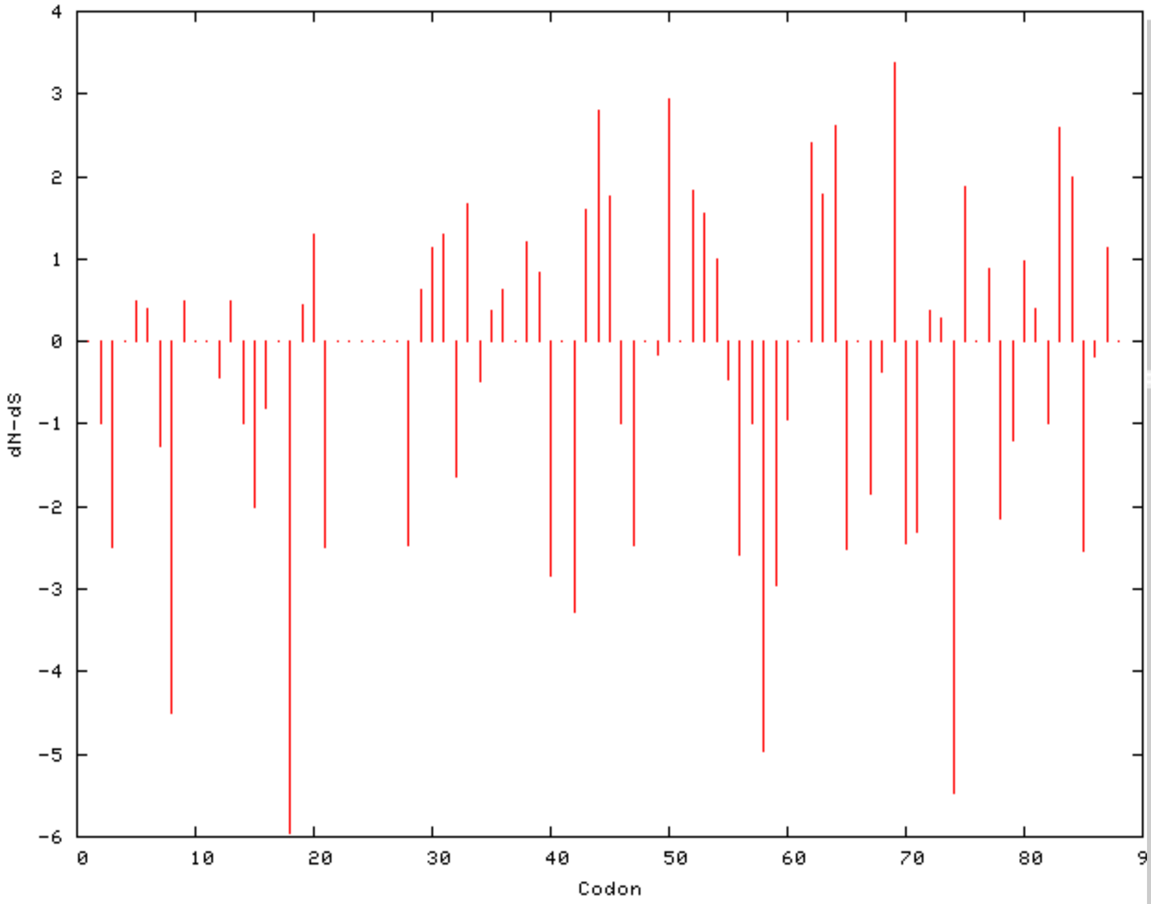
Discussion: Other ways to detect positive selection?

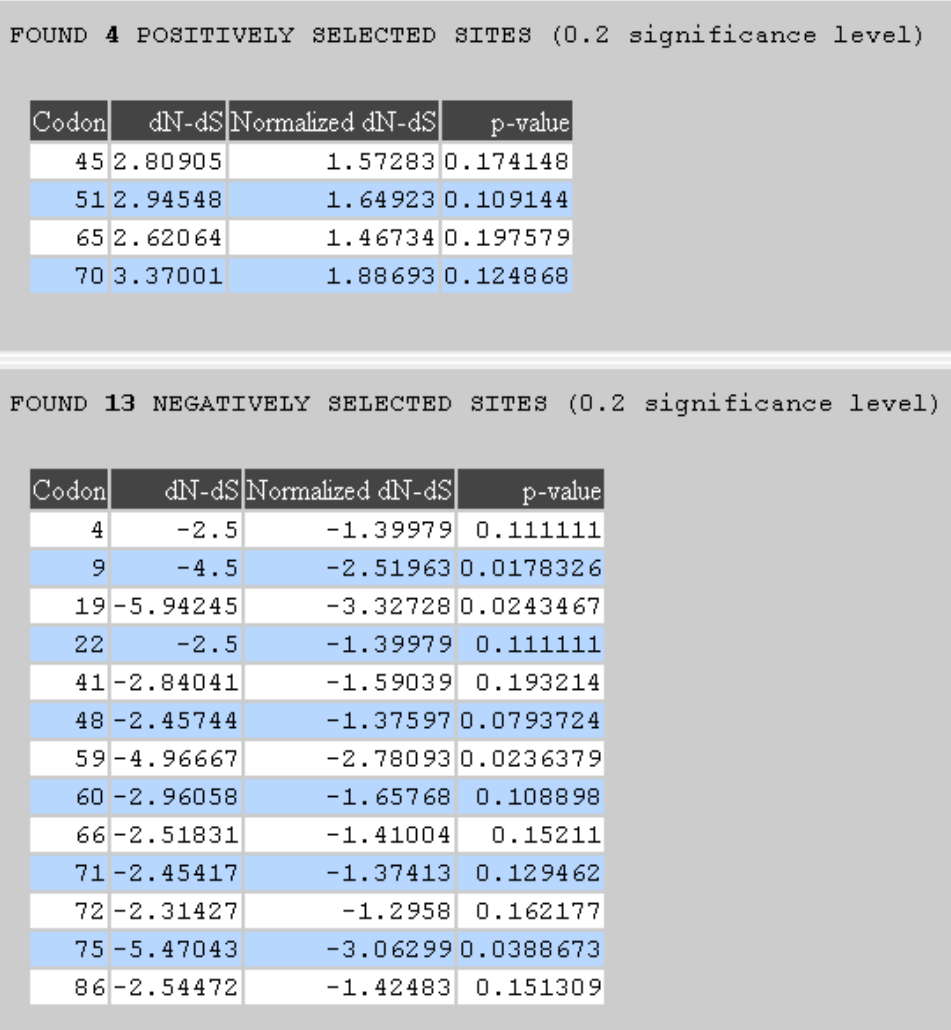
Selective sweep -> fewer alleles present in population

Repeated episodes of positive selection -> high dN

hy-phy

Results of an anaylsis using the SLAC approach





FOUND **4** POSITIVELY SELECTED SITES (0.2 significance level)

Codon	dN-dS	Normalized dN-dS	p-value
45	2.80905	1.57283	0.174148
51	2.94548	1.64923	0.109144
65	2.62064	1.46734	0.197579
70	3.37001	1.88693	0.124868

FOUND **13** NEGATIVELY SELECTED SITES (0.2 significance level)

Codon	dN-dS	Normalized dN-dS	p-value
4	-2.5	-1.39979	0.111111
9	-4.5	-2.51963	0.0178326
19	-5.94245	-3.32728	0.0243467
22	-2.5	-1.39979	0.111111
41	-2.84041	-1.59039	0.193214
48	-2.45744	-1.37597	0.0793724
59	-4.96667	-2.78093	0.0236379
60	-2.96058	-1.65768	0.108898
66	-2.51831	-1.41004	0.15211
71	-2.45417	-1.37413	0.129462
72	-2.31427	-1.2958	0.162177
75	-5.47043	-3.06299	0.0388673
86	-2.54472	-1.42483	0.151309

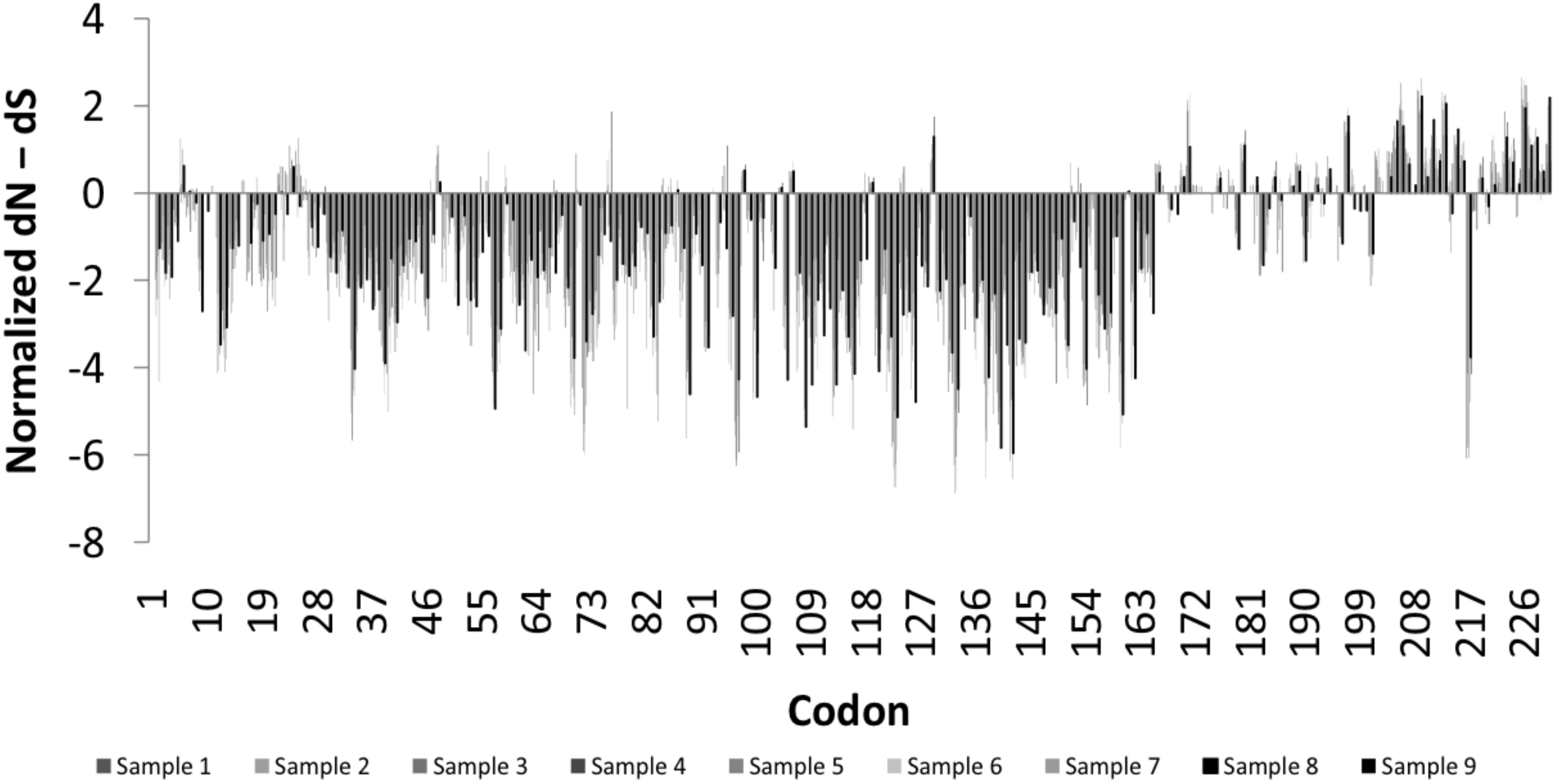


Fig 1. Patterns of substitutions: Bars represent dN > dS (positive) or dN < dS (negative) in random samples of 148 – 150 sequences (A) and the whole dataset of 1312 viruses (B). Included in B are regions of mapped activity and 3D structures of the RNA-binding domain (RBD, panel I) [21] and Effector domain (ED, rotated to expose the 7 β-sheets (panel II) and 2 α-helices (panel II)) [7] with residues under negative (yellow/brown), neutral (gray) or positive (red) selection highlighted. Residues 208-230 not included in the 3D structure of the ED are disordered (compare with figure 5). Note sites with dN > dS map on the helix motifs of the ED or the linkers flanking them or the disordered region.

Hy-Phy -

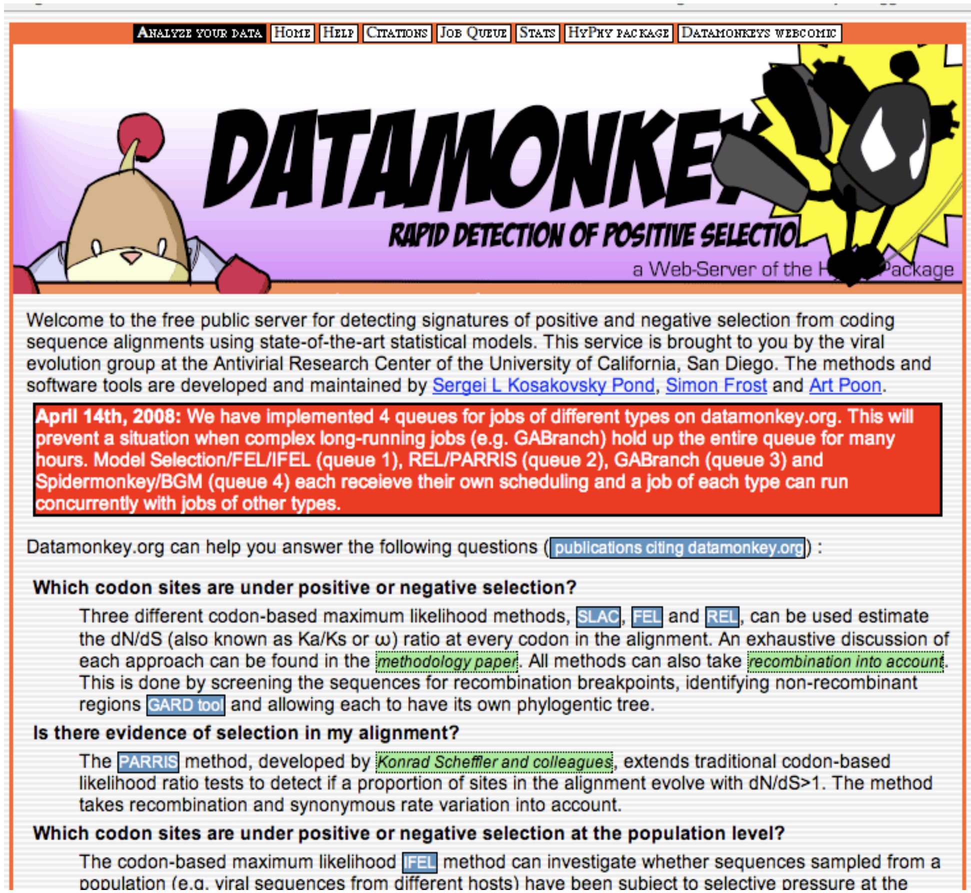
HYPHY PACKAGE
Hypothesis Testing Using Phylogenies

Hypothesis Testing using Phylogenies.

Using Batchfiles or GUI

Information at http://www.hyphy.org/

Selected analyses also can be performed online at http://www.datamonkey.org/



Welcome to the free public server for detecting signatures of positive and negative selection from coding sequence alignments using state-of-the-art statistical models. This service is brought to you by the viral evolution group at the Antivirial Research Center of the University of California, San Diego. The methods and software tools are developed and maintained by Sergei L Kosakovsky Pond, Simon Frost and Art Poon.

April 14th, 2008: We have implemented 4 queues for jobs of different types on datamonkey.org. This will prevent a situation when complex long-running jobs (e.g. GABranch) hold up the entire queue for many hours. Model Selection/FEL/IFEL (queue 1), REL/PARRIS (queue 2), GABranch (queue 3) and Spidermonkey/BGM (queue 4) each receieve their own scheduling and a job of each type can run concurrently with jobs of other types.

Datamonkey.org can help you answer the following questions (publications citing datamonkey.org) :

Which codon sites are under positive or negative selection?

Three different codon-based maximum likelihood methods, SLAC, FEL and REL, can be used estimate the dN/dS (also known as Ka/Ks or ω) ratio at every codon in the alignment. An exhaustive discussion of each approach can be found in the *methodology paper*. All methods can also take *recombination into account*. This is done by screening the sequences for recombination breakpoints, identifying non-recombinant regions GARD tool and allowing each to have its own phylogentic tree.

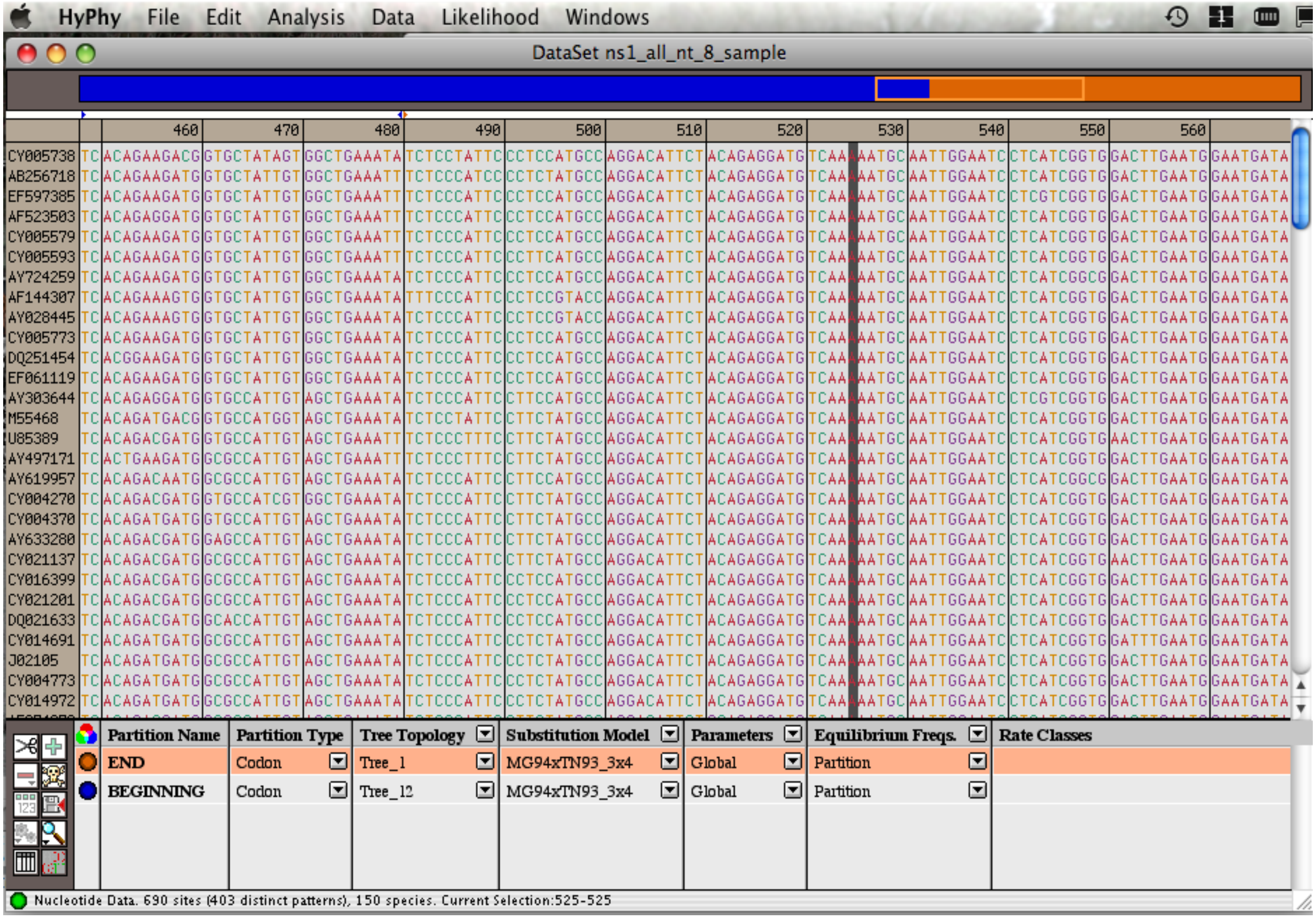
Is there evidence of selection in my alignment?

The PARRIS method, developed by *Konrad Scheffler and colleagues*, extends traditional codon-based likelihood ratio tests to detect if a proportion of sites in the alignment evolve with dN/dS>1. The method takes recombination and synonymous rate variation into account.

Which codon sites are under positive or negative selection at the population level?

The codon-based maximum likelihood IFEL method can investigate whether sequences sampled from a population (e.g. viral sequences from different hosts) have been subject to selective pressure at the

Example testing for dN/dS in two partitions of the data -- John’s dataset



DataSet ns1_all_nt_8_sample

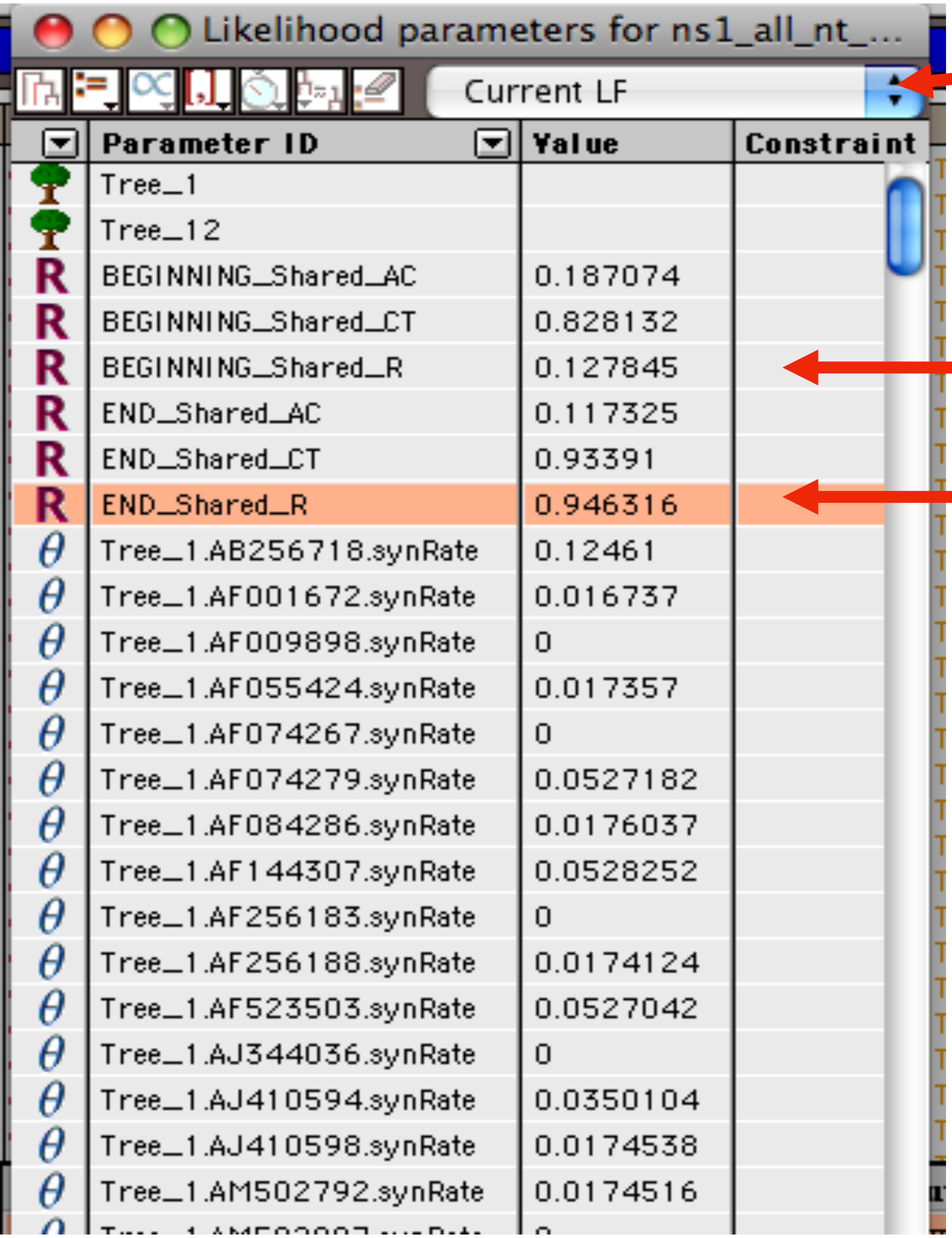
```
                  460        470        480        490        500        510        520        530        540        550        560
CY005738 TC ACAGAAGACG GTGCTATAGT GGCTGAAATA TCTCCTATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AB256718 TC ACAGAAGATG GTGCTATTGT GGCTGAAATT TCTCCCATCC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
EF597385 TC ACAGAAGATG GTGCTATTGT GGCTGAAATT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCGTCGGTG GACTTGAATG GAATGATA
AF523503 TC ACAGAGGATG GTGCTATTGT GGCTGAAATT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY005579 TC ACAGAAGATG GTGCTATTGT GGCTGAAATT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY005593 TC ACAGAAGATG GTGCTATTGT GGCTGAAATT TCTCCCATTC CCTTCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AY724259 TC ACAGAAGATG GTGCTATTGT GGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGCG GACTTGAATG GAATGATA
AF144307 TC ACAGAAAGTG GTGCTATTGT GGCTGAAATA TTTCCCATTC CCTCCGTACC AGGACATTTT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AY028445 TC ACAGAAAGTG GTGCTATTGT GGCTGAAATA TCTCCCATTC CCTCCGTACC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY005773 TC ACAGAAGATG GTGCTATTGT GGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
DQ251454 TC ACGGAAGATG GTGCTATTGT GGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
EF061119 TC ACAGAAGATG GTGCTATTGT GGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AY303644 TC ACAGAGGATG GTGCCATTGT AGCTGAAATA TCTCCCATTC CTTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCGTCGGTG GACTTGAATG GAATGATA
M55468   TC ACAGATGACG GTGCCATGGT AGCTGAAATA TCTCCTATTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
U85389   TC ACAGACGATG GTGCCATTGT AGCTGAAATT TCTCCCTTTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG AACTTGAATG GAATGATA
AY497171 TC ACTGAAGATG GCGCCATTGT AGCTGAAATT TCTCCCTTTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AY619957 TC ACAGACAATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CTTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGCG GACTTGAATG GAATGATA
CY004270 TC ACAGACGATG GTGCCATCGT GGCTGAAATA TCTCCCATTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY004370 TC ACAGATGATG GTGCCATTGT AGCTGAAATA TCTCCCATTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
AY633280 TC ACAGACGATG GAGCCATTGT AGCTGAAATA TCTCCCATTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY021137 TC ACAGACGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CTTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG AACTTGAATG GAATGATA
CY016399 TC ACAGACGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY021201 TC ACAGACGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
DQ021633 TC ACAGACGATG GCACCATTGT AGCTGAAATA TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY014691 TC ACAGATGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GATTTGAATG GAATGATA
J02105   TC ACAGATGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY004773 TC ACAGATGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
CY014972 TC ACAGATGATG GCGCCATTGT AGCTGAAATA TCTCCCATTC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAATC CTCATCGGTG GACTTGAATG GAATGATA
```

	Partition Name	Partition Type	Tree Topology	Substitution Model	Parameters	Equilibrium Freqs.	Rate Classes
●	END	Codon	Tree_1	MG94xTN93_3x4	Global	Partition	
●	BEGINNING	Codon	Tree_12	MG94xTN93_3x4	Global	Partition	

Nucleotide Data. 690 sites (403 distinct patterns), 150 species. Current Selection:525-525

Set up two partitions, define model for each, optimize likelihood

Example testing for dN/dS in two partitions of the data -- John’s dataset



Likelihood parameters for ns1_all_nt_...

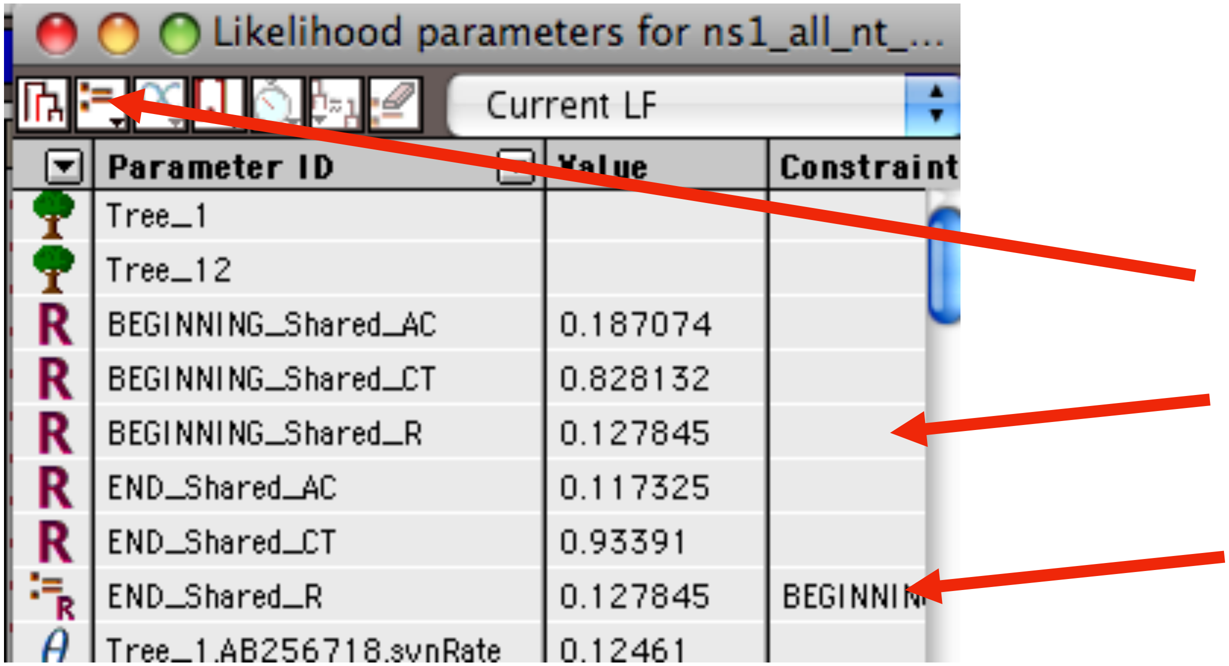
Current LF

	Parameter ID	Value	Constraint
🌳	Tree_1		
🌳	Tree_12		
R	BEGINNING_Shared_AC	0.187074	
R	BEGINNING_Shared_CT	0.828132	
R	BEGINNING_Shared_R	0.127845	
R	END_Shared_AC	0.117325	
R	END_Shared_CT	0.93391	
R	END_Shared_R	0.946316	
θ	Tree_1.AB256718.synRate	0.12461	
θ	Tree_1.AF001672.synRate	0.016737	
θ	Tree_1.AF009898.synRate	0	
θ	Tree_1.AF055424.synRate	0.017357	
θ	Tree_1.AF074267.synRate	0	
θ	Tree_1.AF074279.synRate	0.0527182	
θ	Tree_1.AF084286.synRate	0.0176037	
θ	Tree_1.AF144307.synRate	0.0528252	
θ	Tree_1.AF256183.synRate	0	
θ	Tree_1.AF256188.synRate	0.0174124	
θ	Tree_1.AF523503.synRate	0.0527042	
θ	Tree_1.AJ344036.synRate	0	
θ	Tree_1.AJ410594.synRate	0.0350104	
θ	Tree_1.AJ410598.synRate	0.0174538	
θ	Tree_1.AM502792.synRate	0.0174516	

Save Likelihood Function
then
select as alternative

The dN/dS ratios for the two partitions are different.

Example testing for dN/dS in two partitions of the data -- John’s dataset



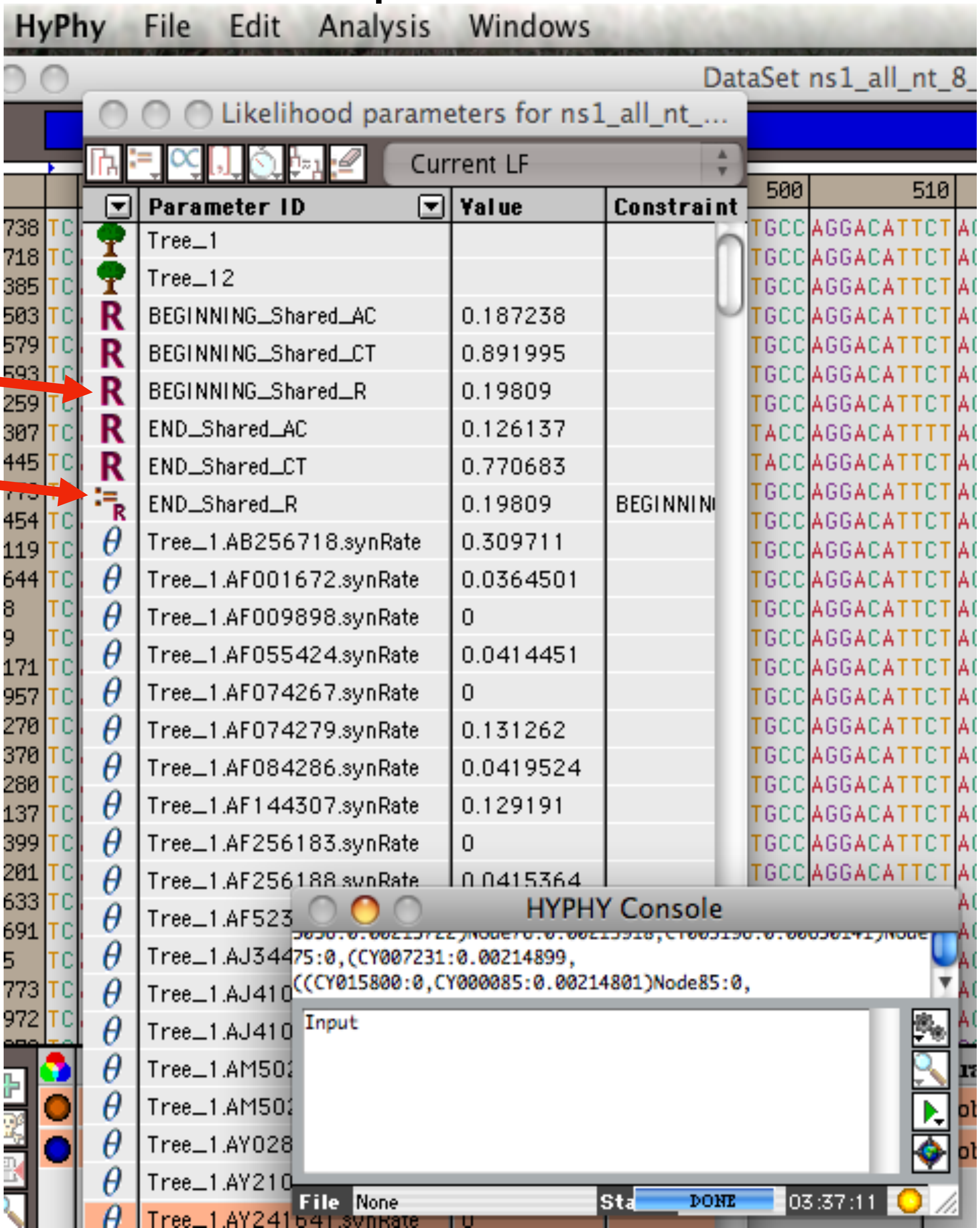
Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187074	
BEGINNING_Shared_CT	0.828132	
BEGINNING_Shared_R	0.127845	
END_Shared_AC	0.117325	
END_Shared_CT	0.93391	
END_Shared_R	0.127845	BEGINNIN
Tree_1.AB256718.synRate	0.12461	

Set up null hypothesis, i.e.:

The two dN/dS are equal

(to do, select both rows and then click the define as equal button on top)

Example testing for dN/dS in two partitions of the data -- John's dataset



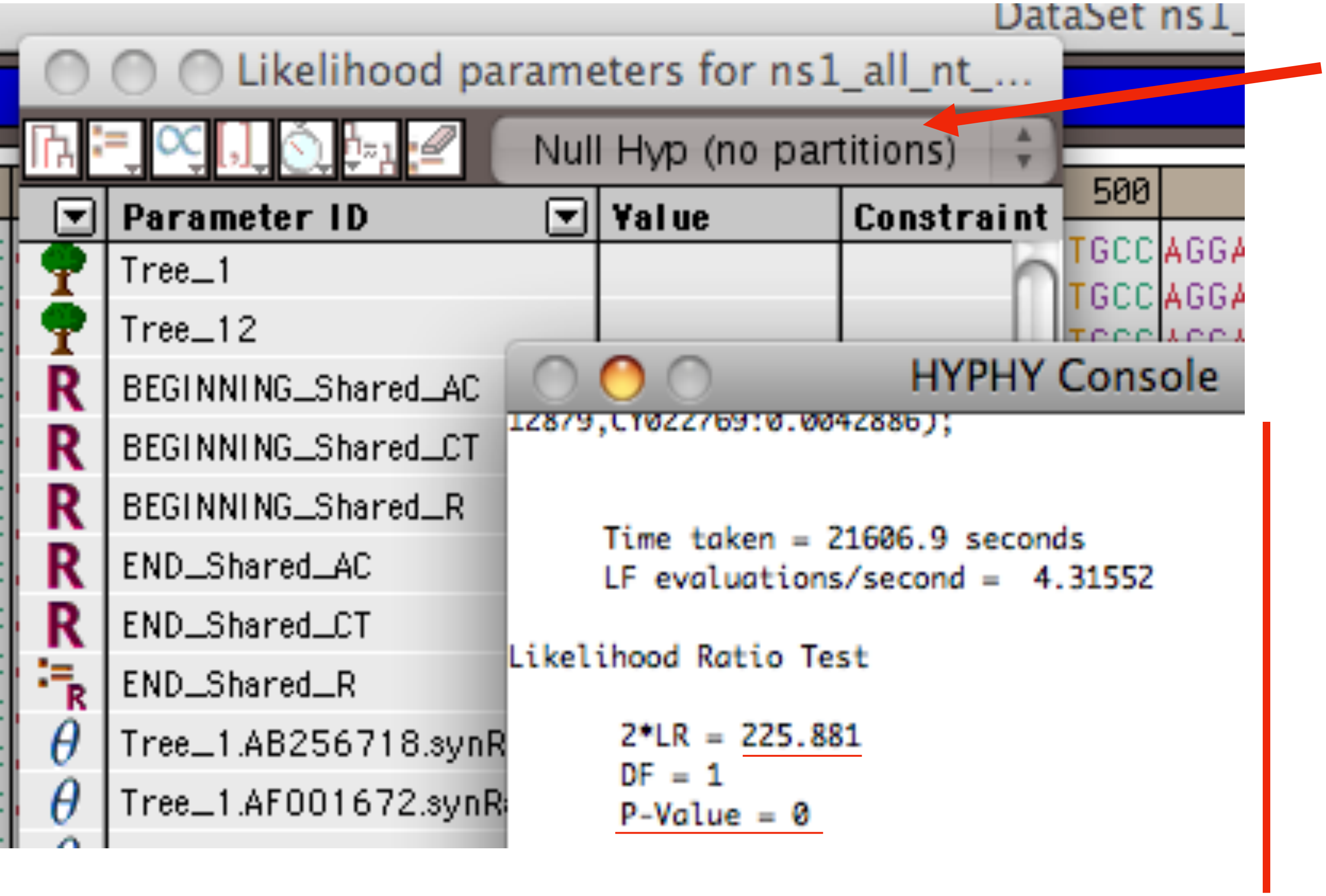
Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187238	
BEGINNING_Shared_CT	0.891995	
BEGINNING_Shared_R	0.19809	
END_Shared_AC	0.126137	
END_Shared_CT	0.770683	
END_Shared_R	0.19809	BEGINNIN
Tree_1.AB256718.synRate	0.309711	
Tree_1.AF001672.synRate	0.0364501	
Tree_1.AF009898.synRate	0	
Tree_1.AF055424.synRate	0.0414451	
Tree_1.AF074267.synRate	0	
Tree_1.AF074279.synRate	0.131262	
Tree_1.AF084286.synRate	0.0419524	
Tree_1.AF144307.synRate	0.129191	
Tree_1.AF256183.synRate	0	
Tree_1.AF256188.synRate	0.0415364	
Tree_1.AF523		
Tree_1.AJ344		
Tree_1.AJ410		
Tree_1.AJ410		
Tree_1.AM502		
Tree_1.AM502		
Tree_1.AY028		
Tree_1.AY210		
Tree_1.AY241641.synRate	0	

```
75:0,(CY007231:0.00214899,
((CY015800:0,CY000085:0.00214801)Node85:0,
```

Input

File None Sta DONE 03:37:11

Example testing for dN/dS in two partitions of the data -- John’s dataset



DataSet ns1_

Likelihood parameters for ns1_all_nt_...

Null Hyp (no partitions)

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC		
BEGINNING_Shared_CT		
BEGINNING_Shared_R		
END_Shared_AC		
END_Shared_CT		
END_Shared_R		
Tree_1.AB256718.synR		
Tree_1.AF001672.synR		

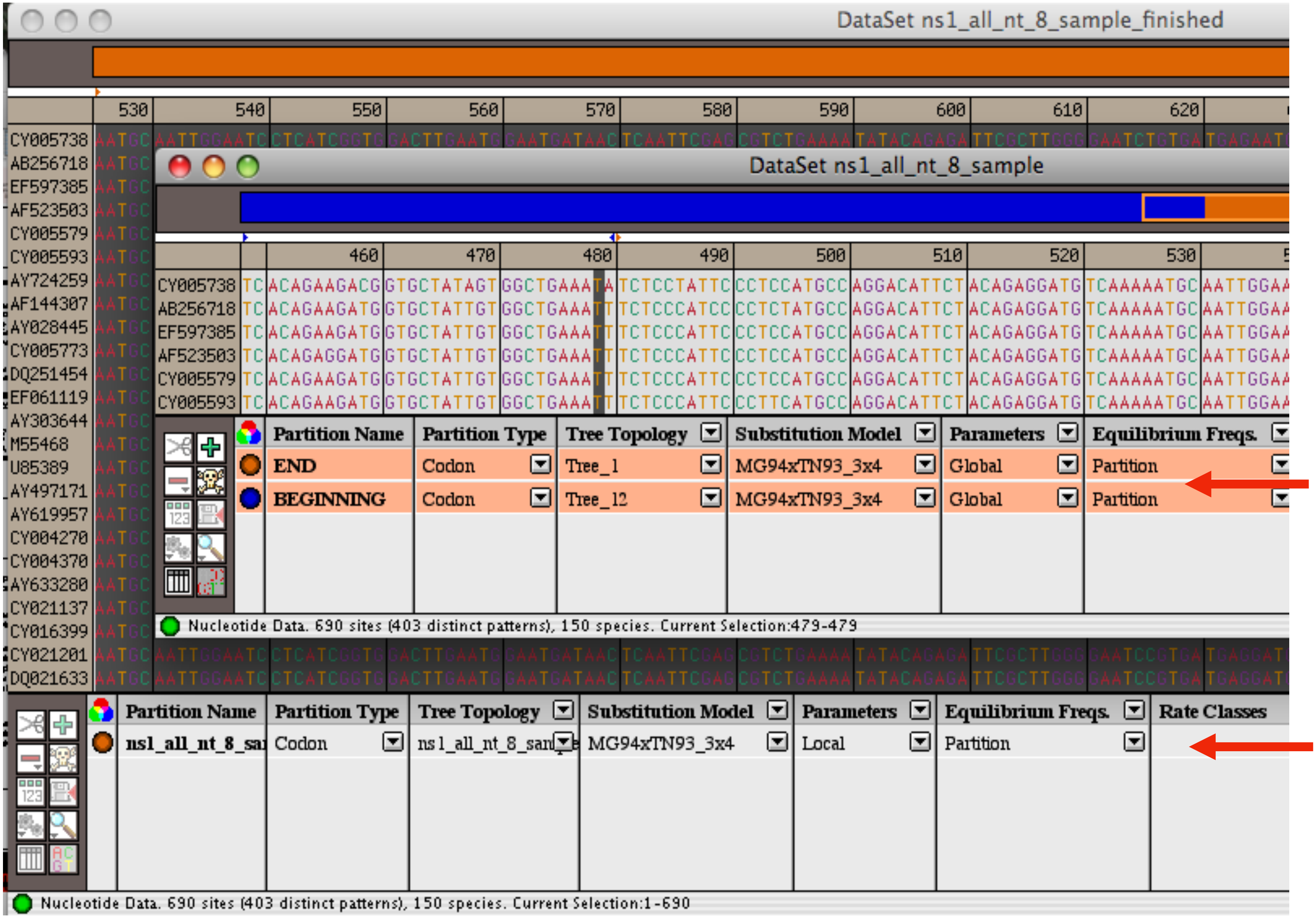
HYPHY Console

```
12879,CY022769:0.0042886);


          Time taken = 21606.9 seconds
          LF evaluations/second =   4.31552

Likelihood Ratio Test

           2*LR = 225.881
           DF = 1
           P-Value = 0
```

After selecting LRT (= Likelihood Ratio test), the console displays the result, i.e., **the beginning and end of the sequence alignment have significantly different dN/dS ratios.**

Example testing for dN/dS in two partitions of the data -- John’s dataset

Alternatively, especially if the the two models are not nested, one can set up two different windows with the same dataset:



DataSet ns1_all_nt_8_sample_finished

DataSet ns1_all_nt_8_sample

```
            460       470       480       490       500       510       520       530
CY005738 TC ACAGAAGACG GTGCTATAGT GGCTGAAA TA TCTCCTATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
AB256718 TC ACAGAAGATG GTGCTATTGT GGCTGAAA TT TCTCCCATCC CCTCTATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
EF597385 TC ACAGAAGATG GTGCTATTGT GGCTGAAA TT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
AF523503 TC ACAGAGGATG GTGCTATTGT GGCTGAAA TT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
CY005579 TC ACAGAAGATG GTGCTATTGT GGCTGAAA TT TCTCCCATTC CCTCCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
CY005593 TC ACAGAAGATG GTGCTATTGT GGCTGAAA TT TCTCCCATTC CCTTCATGCC AGGACATTCT ACAGAGGATG TCAAAAATGC AATTGGAA
```

	Partition Name	Partition Type	Tree Topology	Substitution Model	Parameters	Equilibrium Freqs.
●	END	Codon	Tree_1	MG94xTN93_3x4	Global	Partition
●	BEGINNING	Codon	Tree_12	MG94xTN93_3x4	Global	Partition

Nucleotide Data. 690 sites (403 distinct patterns), 150 species. Current Selection:479-479

Model 1

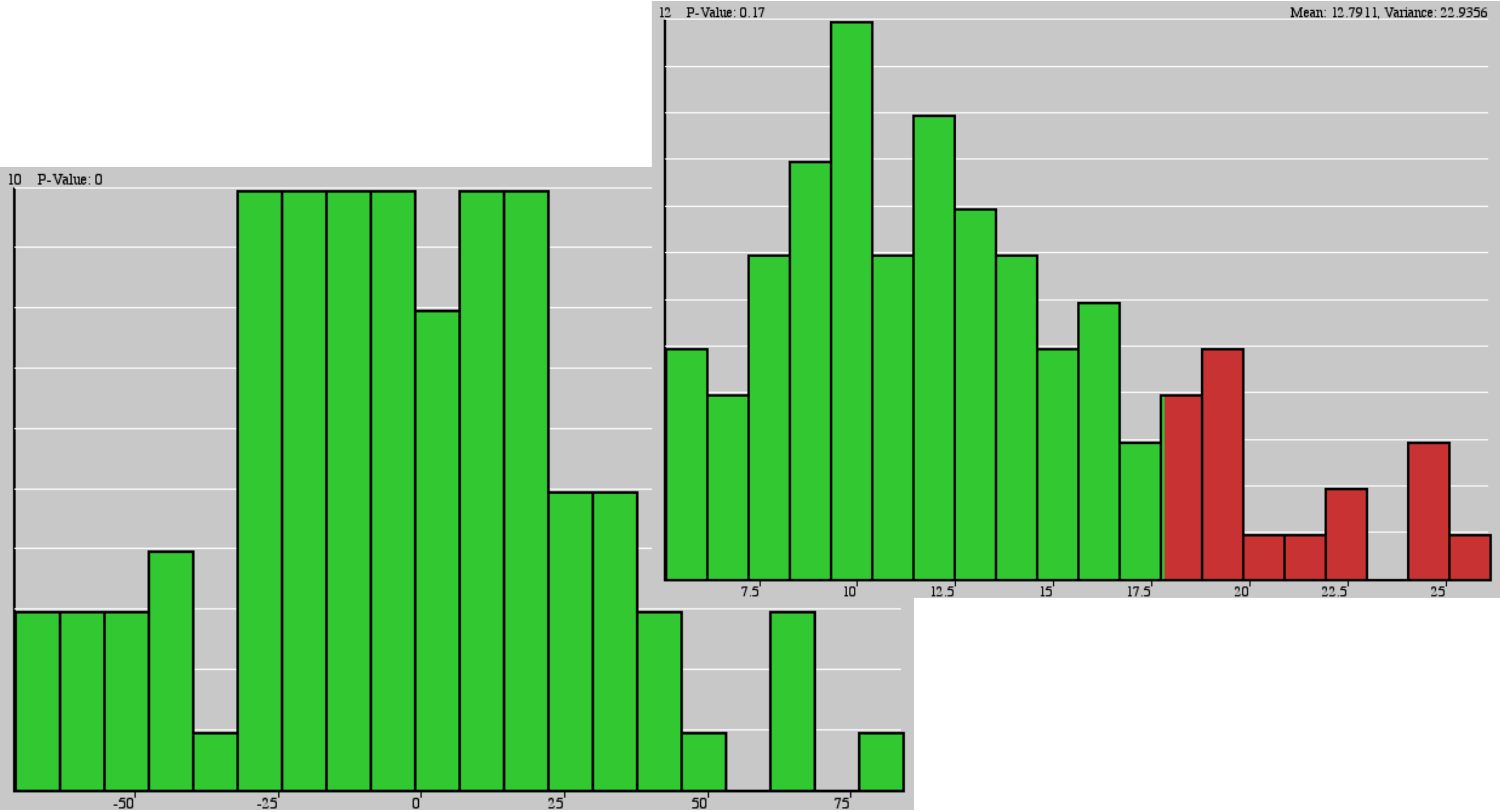
	Partition Name	Partition Type	Tree Topology	Substitution Model	Parameters	Equilibrium Freqs.	Rate Classes
●	nsl_all_nt_8_sa	Codon	ns1_all_nt_8_san	MG94xTN93_3x4	Local	Partition	

Nucleotide Data. 690 sites (403 distinct patterns), 150 species. Current Selection:1-690

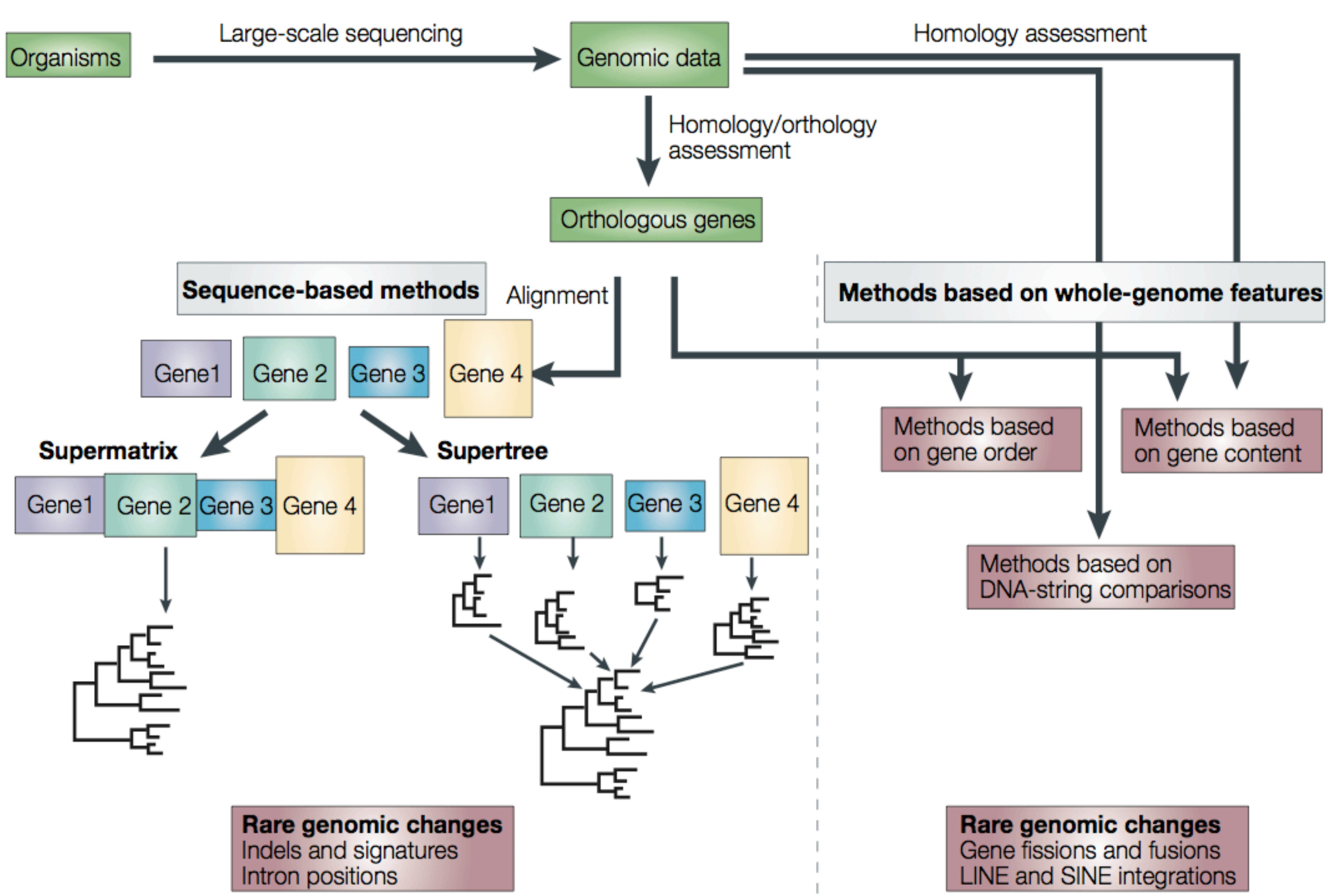
Model 2

Example testing for dN/dS in two partitions of the data -- John’s dataset

Simulation under model 2, evalutation under model 1, calculate LR
Compare real LR to distribution from simulated LR values. The result might look something like this or this



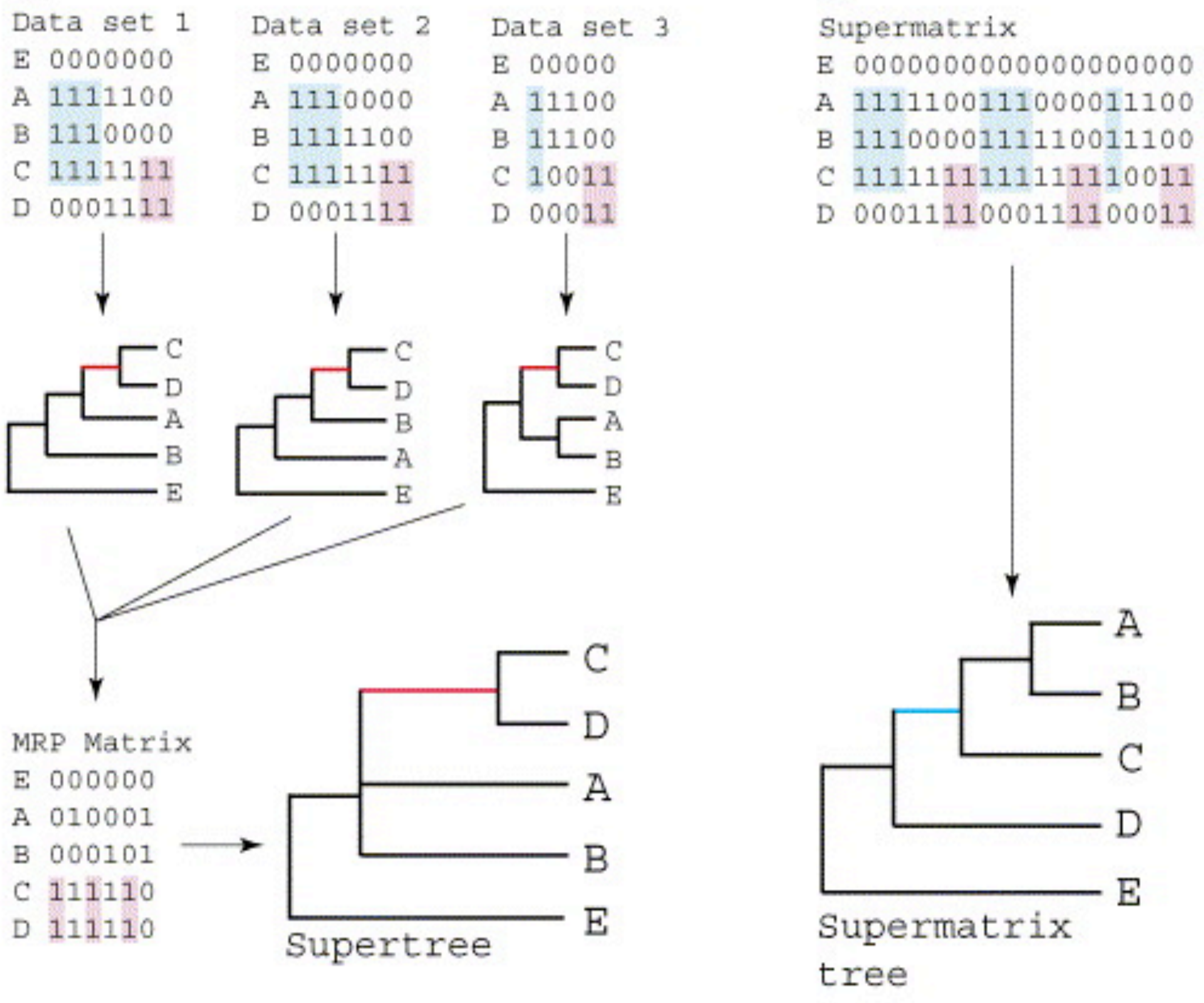
Box 2 | Methods of phylogenomic inference



The flowchart shows steps in the inference of evolutionary trees from genomic data. Genomic information is obtained by large-scale DNA sequencing. In general, sets of orthologous genes are then assembled from specific sets of species for phylogenetic analysis. This homology or orthology assessment is a crucial step that is almost always based on simple similarity comparisons (for example, BLAST[158] searches). Most methods used for the subsequent reconstruction of phylogenetic trees are either sequence-based or are based on whole-genome features.

From:
Delsuc F, Brinkmann H, Philippe H.
Phylogenomics and the reconstruction of the tree of life.
Nat Rev Genet. 2005 May;6(5):361-75.

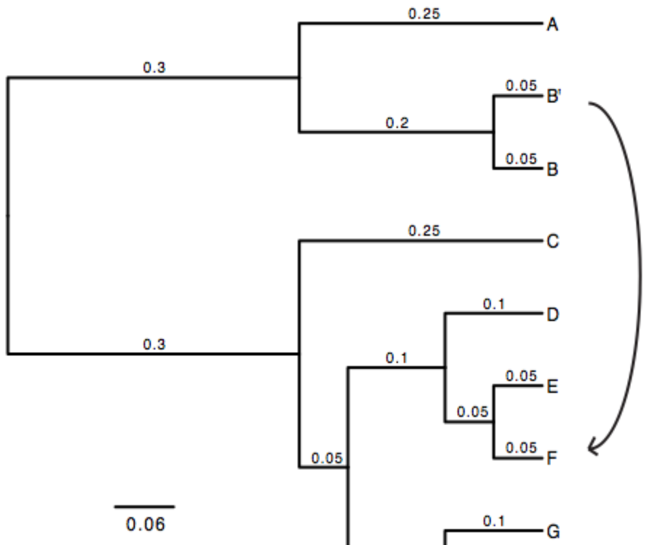
Supertree *vs.* Supermatrix



TRENDS in Ecology & Evolution

From:
Alan de Queiroz John Gatesy:
The supermatrix approach to systematics
Trends Ecol Evol. 2007 Jan;22(1):34-41

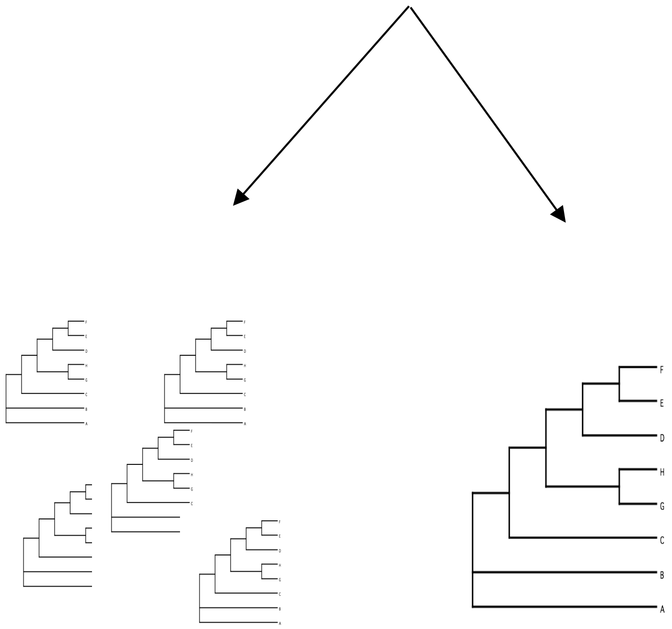
Schematic of MRP supertree (left) and parsimony supermatrix (right) approaches to the analysis of three data sets. Clade C+D is supported by all three separate data sets, but not by the supermatrix. Synapomorphies for clade C+D are highlighted in pink. Clade A+B+C is not supported by separate analyses of the three data sets, but is supported by the supermatrix. Synapomorphies for clade A+B+C are highlighted in blue. E is the outgroup used to root the tree.



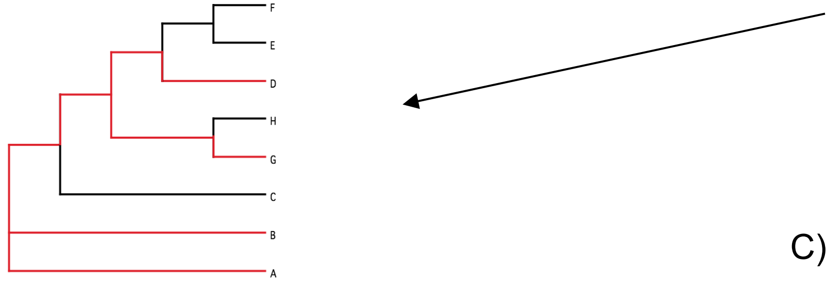
A) Template tree

```
A    IYQILLVNNSSLSTVNWALGQDEDLETQTKTAFLDMSFITKIKAVQDVGEYALFNAENAG
B    ILQILLVNLSSLSTVKWHLSQDEDLETQTESAFLDMTFVNKIEAVQDVGEYVVFNAENAW
C    ICQILMVNLSSFSTVSWQLAQDEDLETQTGGLLLDMRFITKVTTQQDVAEYPLFNAENAI
D    ICAILMINVSALSTVYWKLAQDEDLETQTSGLFLSMRFMAKIATQQDVGEYSLFNAKNTV
E    ICLILLINTSAESTVNWRLTQDEDLETQTGGFFLSMRFMTKIRTRQDVGEYSLFNAKNTV
F    ICAILLINTSAHSTVNWSLTQDEDLETQTGGCFLSMRFMTKIRTQQDVGEYSLFNAKNTE
G    ICAILPINASATSTVDWTLKQDEDLETQTGGFFLEMRFMPRISTQQDVAEYLLFNAENAS
H    ICAILLINASALSTVNWHLQQDEDLETQTGGFFLEMRFMTKISTQQDVAEYSLFNAENAT
     *  ** :* *: *** * * *********   :*.* *: :: : ***.** :***:*:
```

B) Generate 100 datasets using Evolver with certain amount of HGTs



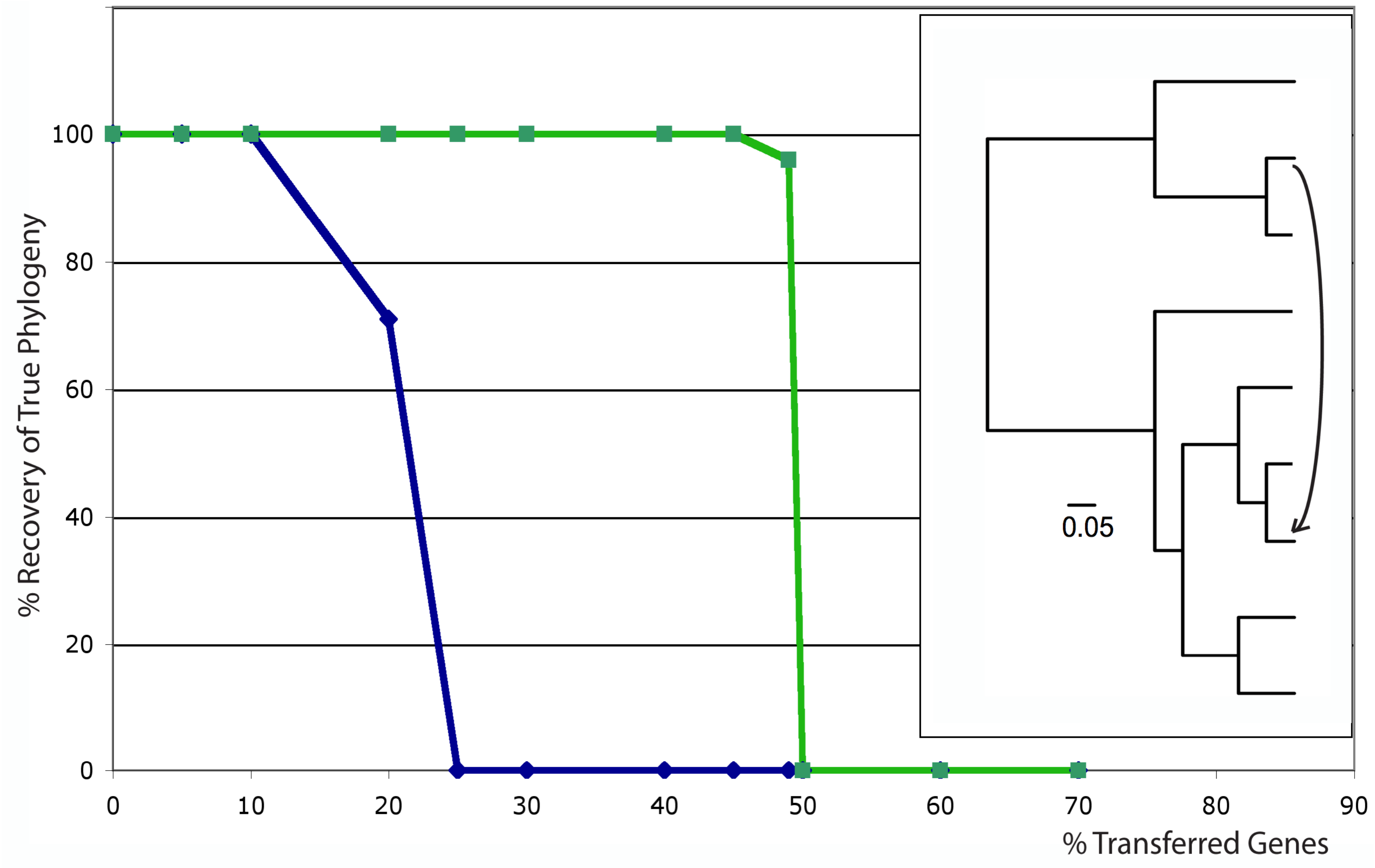
C) Calculate 1 tree using the concatenated dataset or 100 individual trees



D) Calculate Quartet based tree using Quartet Suite

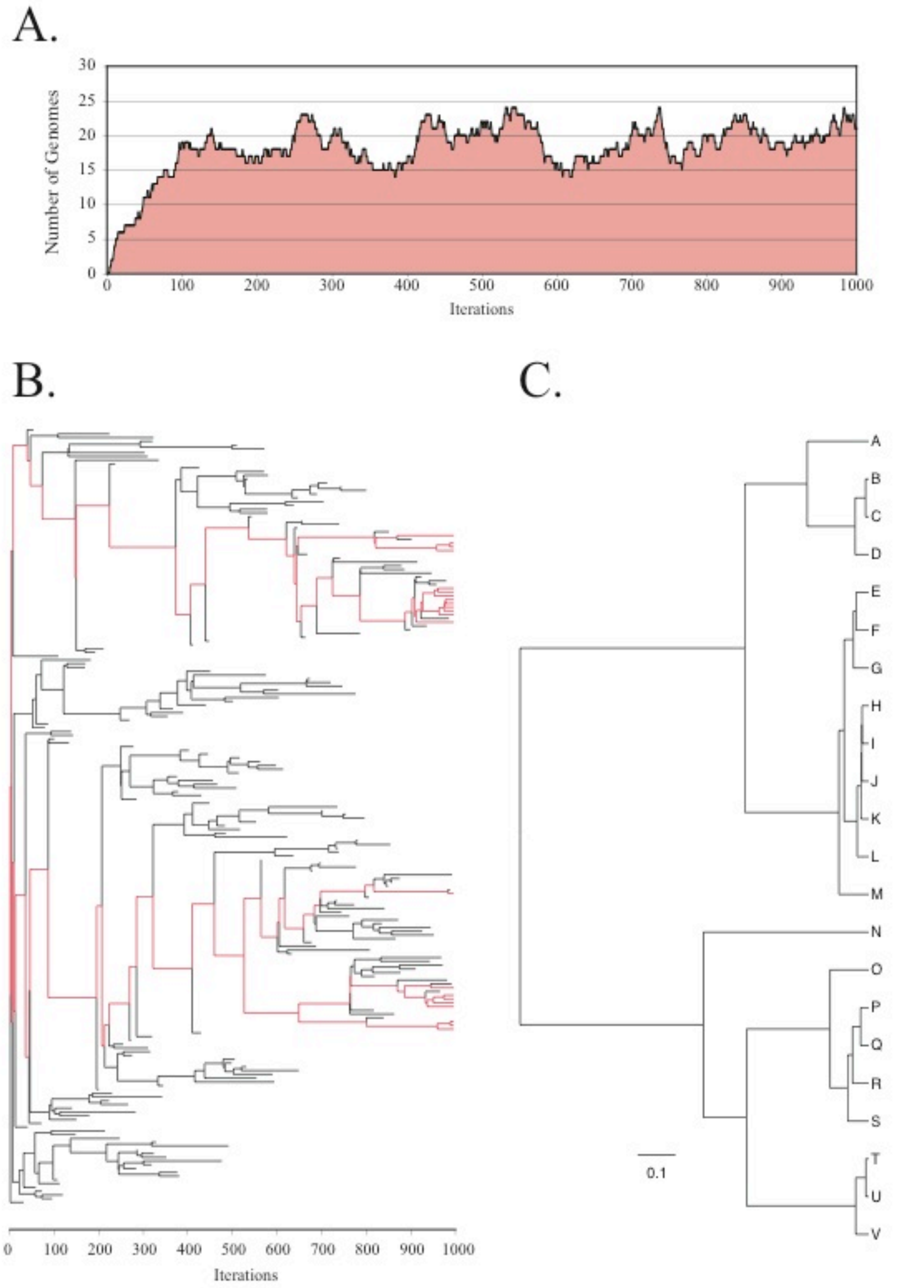
Repeated 100 times…

Supermatrix versus Quartet based Supertree

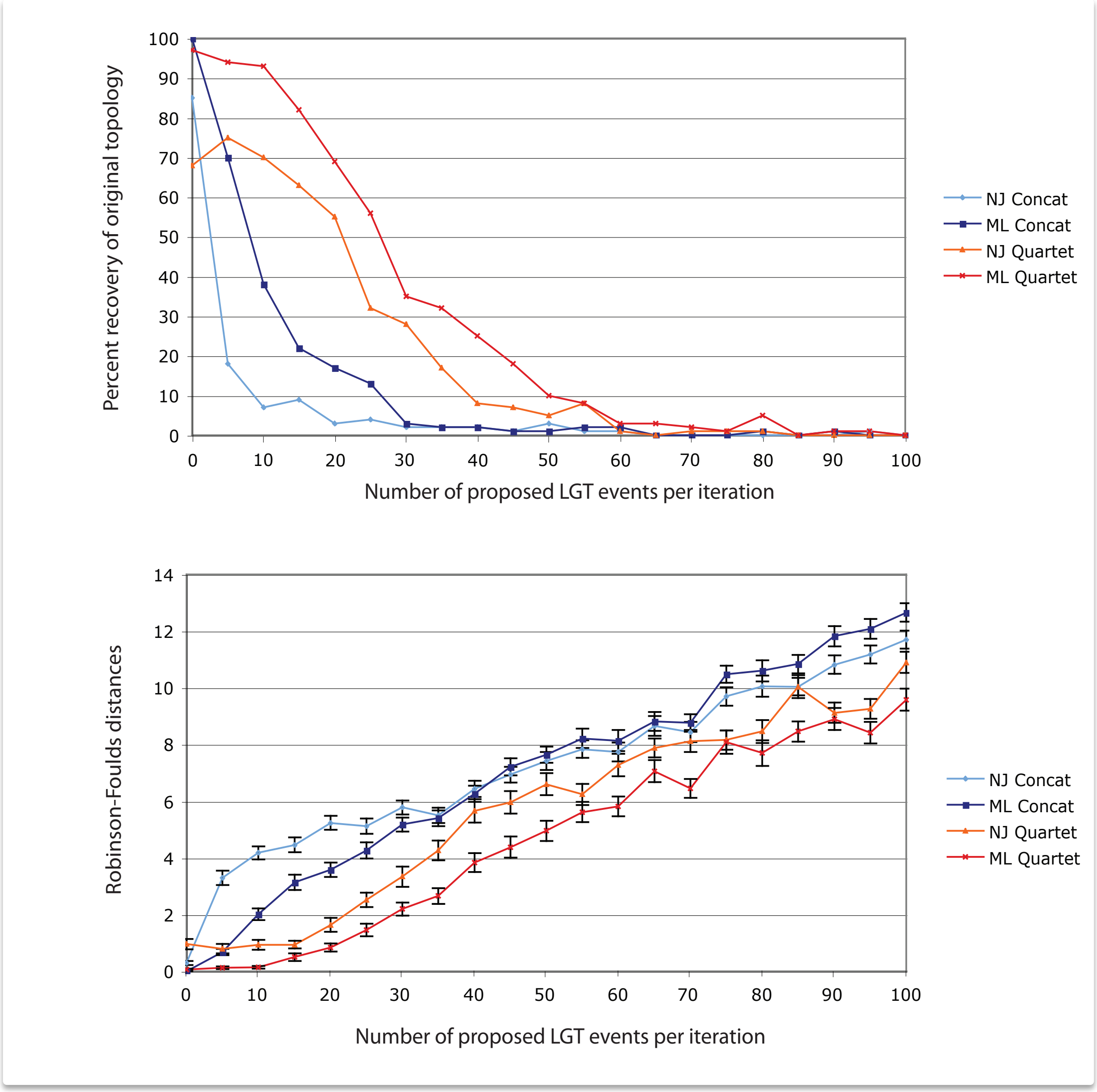


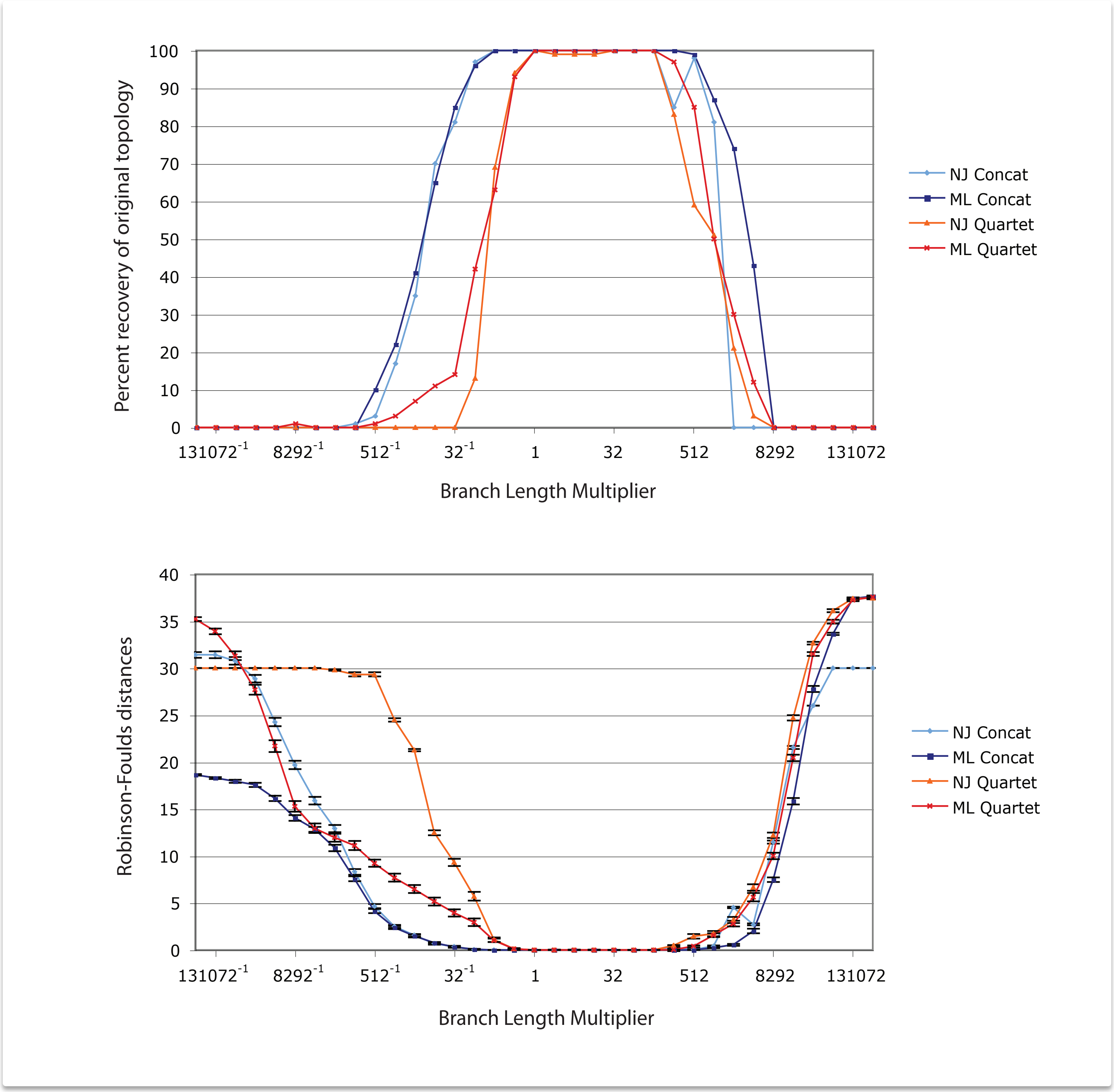
inset: simulated phylogeny

Note : Using same genome seed random number will reproduce same genome history



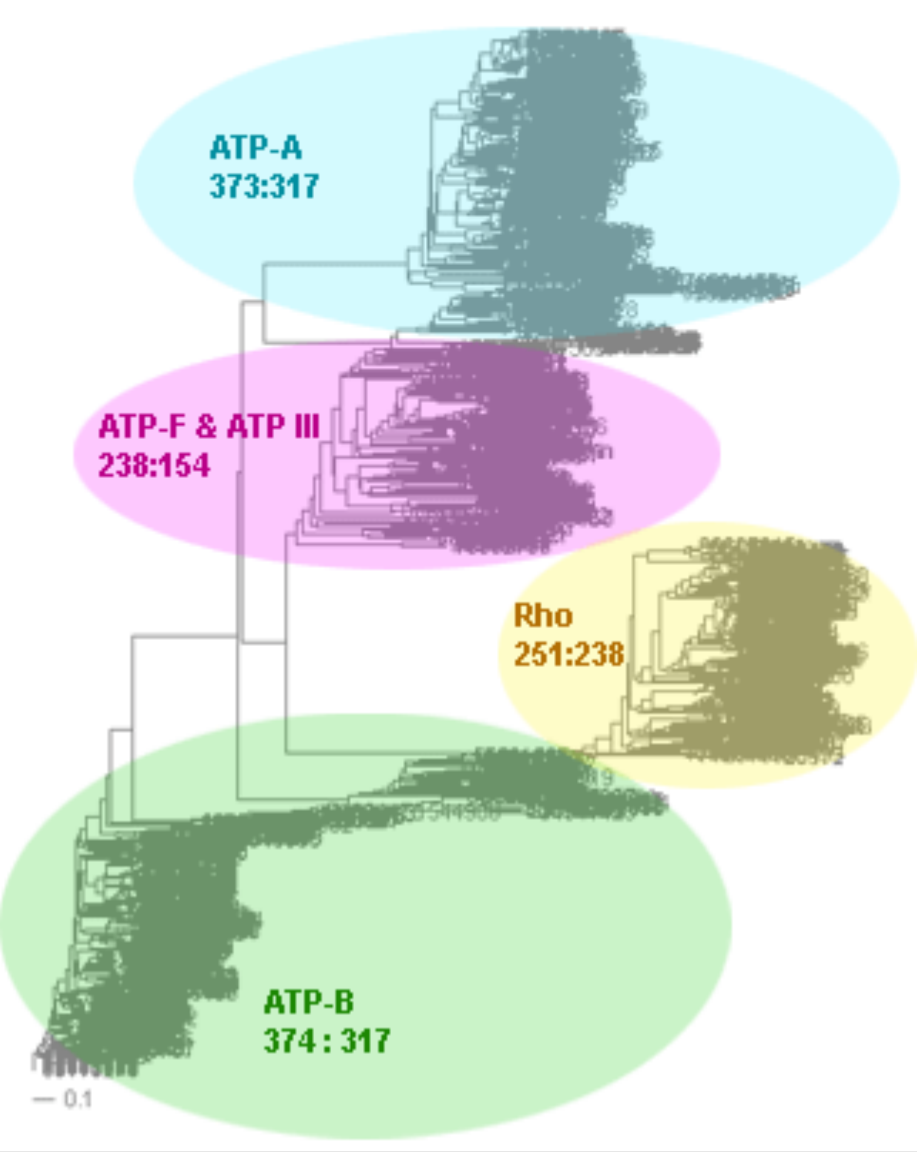
HGT EvolSimulator Results




Percent recovery of original topology
100
90
80
70
60
50
40
30
20
10
0
131072^{-1}
8292^{-1}
512^{-1}
32^{-1}
1
32
512
8292
131072
Branch Length Multiplier
NJ Concat
ML Concat
NJ Quartet
ML Quartet
Robinson-Foulds distances
40
35
30
25
20
15
10
5
0
131072^{-1}
8292^{-1}
512^{-1}
32^{-1}
1
32
512
8292
131072
Branch Length Multiplier
NJ Concat
ML Concat
NJ Quartet
ML Quartet

Automated Assembly of Gene Families Using BranchClust

J. Peter Gogarten
University of Connecticut
Dept. of Molecular and Cell Biol.



Collaborators:

Maria Poptsova (UConn)
Fenglou Mao (UGA)

Funded through the
Edmond J. Safra Bioinformatics Program.
Fulbright Fellowship,
NASA Exobiology Program,
NSF Assembling the Tree of Life Programm and
NASA Applied Information Systems Research Program

Why do we need gene families?

Which genes are common between different species?

Which genes were duplicated in which species?
(Lineage specific gene family expansions)

Do all the common genes share a common history?
Reconstruct (parts of) the tree/net of life /
Detect horizontally transferred genes.

Why do we need gene families?

Help in genome annotation.

A) Genes in a family should have same annotation across species (usually).

B) Genes present in almost all genomes of a group of closely related organisms, but absent in one or tow members, might represent genome annotation artifacts.

Detecting Errors in Genome Annotation

Analysis of 8 strains of Escherichia coli

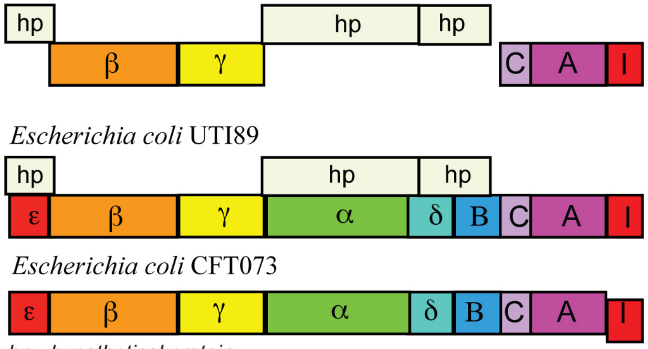
Number of families with 1 missing gene

Escherichia coli 536	56
Escherichia coli APEC_O1	196
Escherichia coli CFT073	45
Escherichia coli K12	4
Escherichia coli O157H7	33
Escherichia coli O157H7 EDL933	6
Escherichia coli UTI89	20
Escherichia coli W3110	8
Total:	***368***

Example of missed ORFs

ATP synthase operon

4 missing genes in Escherichia coli APEC O1



Escherichia coli UTI89

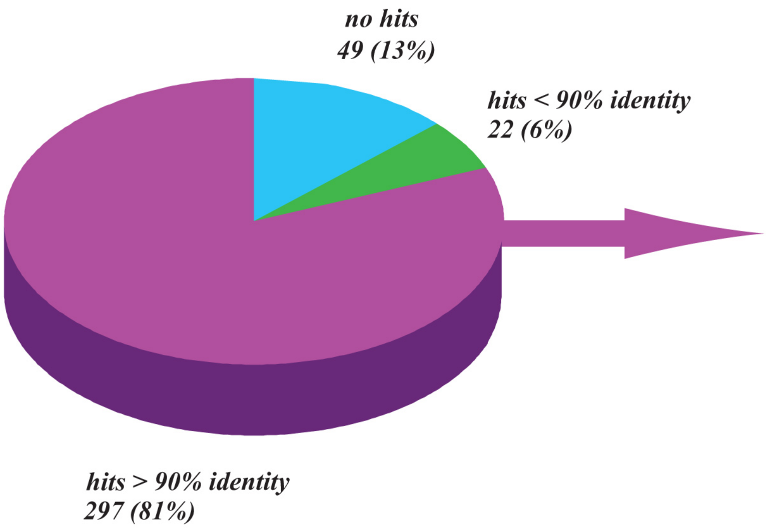
Escherichia coli CFT073

hp - hypothetical protein

ε, β, γ, α, δ, B, C, A, I - ATP synthase subunits

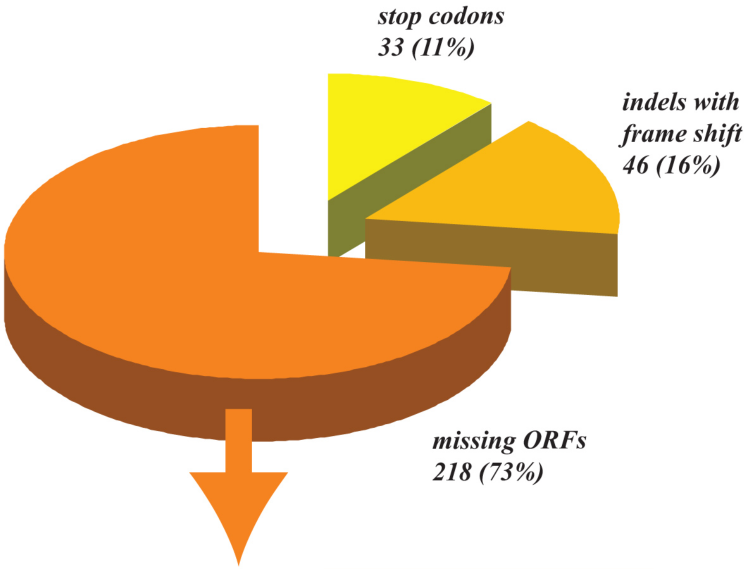
Analysis of 368 missing orthologs with blastn

An ortholog from a family with 1 missing gene was used as a query against nucleotide sequence of a full genome with missing gene



Analysis of 297 hits with > 90% identity in genomes with a missing gene

Each hit was analyzed and classified as it is depicted on plates (b),(c) and (d).



no annotation
114

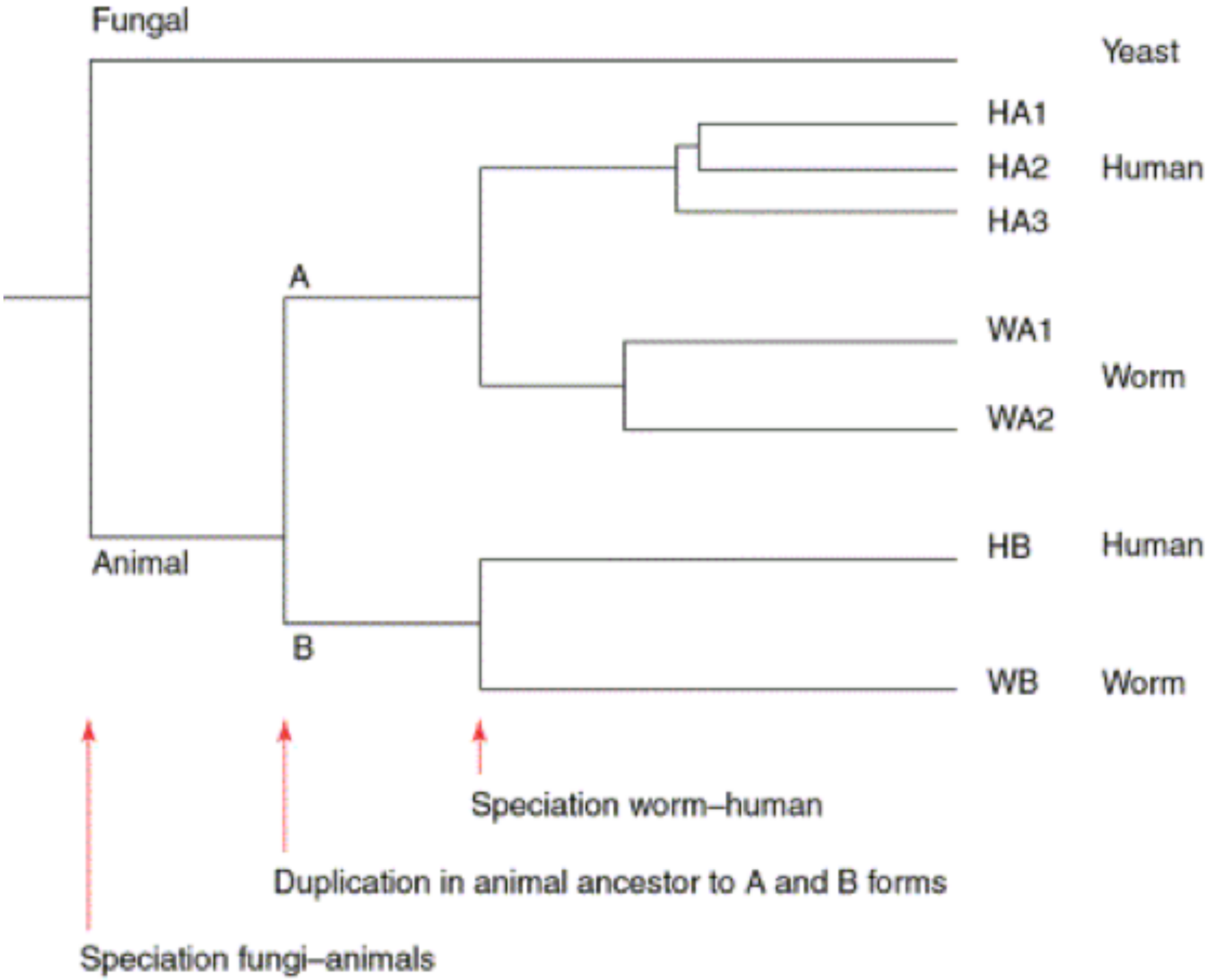
alternative annotation
104

same strand
44

opposite strand
60

Types of Paralogs: In- and Outparalogs

…. all genes in the HA* set are co-orthologous to all genes in the WA* set. The genes HA* are hence ‘inparalogs’ to each other when comparing human to worm. By contrast, the genes HB and HA* are ‘outparalogs’ when comparing human with worm. However, HB and HA*, and WB and WA* are inparalogs when comparing with yeast, because the animal–yeast split pre-dates the HA*–HB duplication.



From: Sonnhammer and Koonin: Orthology, paralogy and proposed classification for paralog TIG 18 (12) 2002, 619-620

Selection of Orthologous Gene Families

All automated methods for assembling sets of orthologous genes are based on sequence similarities.

↓

BLAST hits

Triangular circular BLAST significant hits

(COG, or Cluster of Orthologous Groups)

Sequence identity of 30% and greater

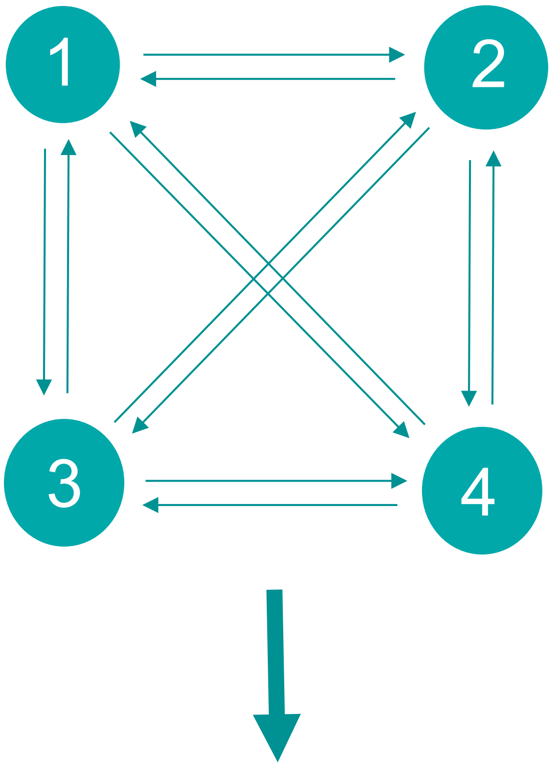
(SCOP database)

Similarity complemented by HMM-profile analysis

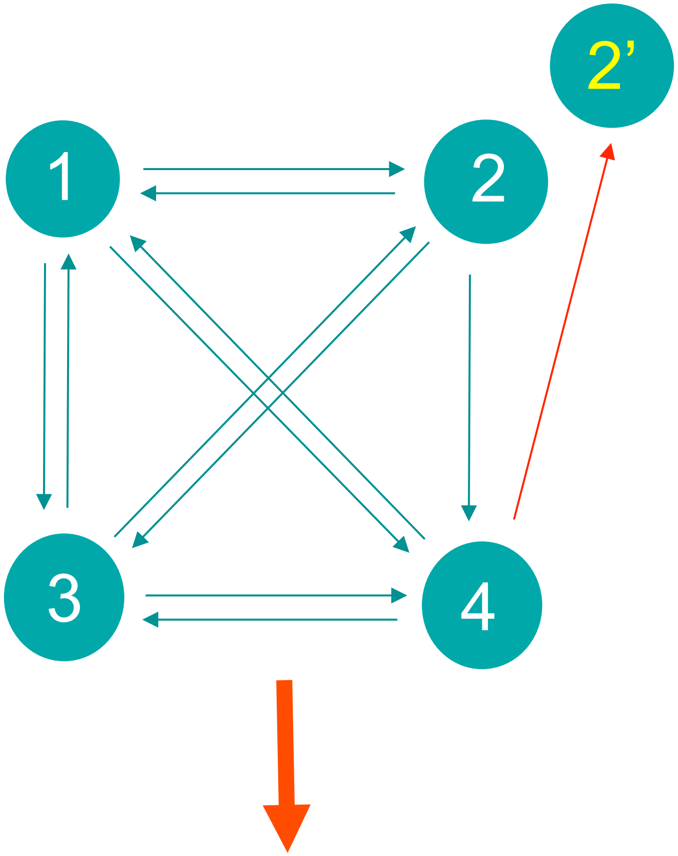
Pfam database

Reciprocal BLAST hit method

Strict Reciprocal BLAST Hit Method



1 gene family



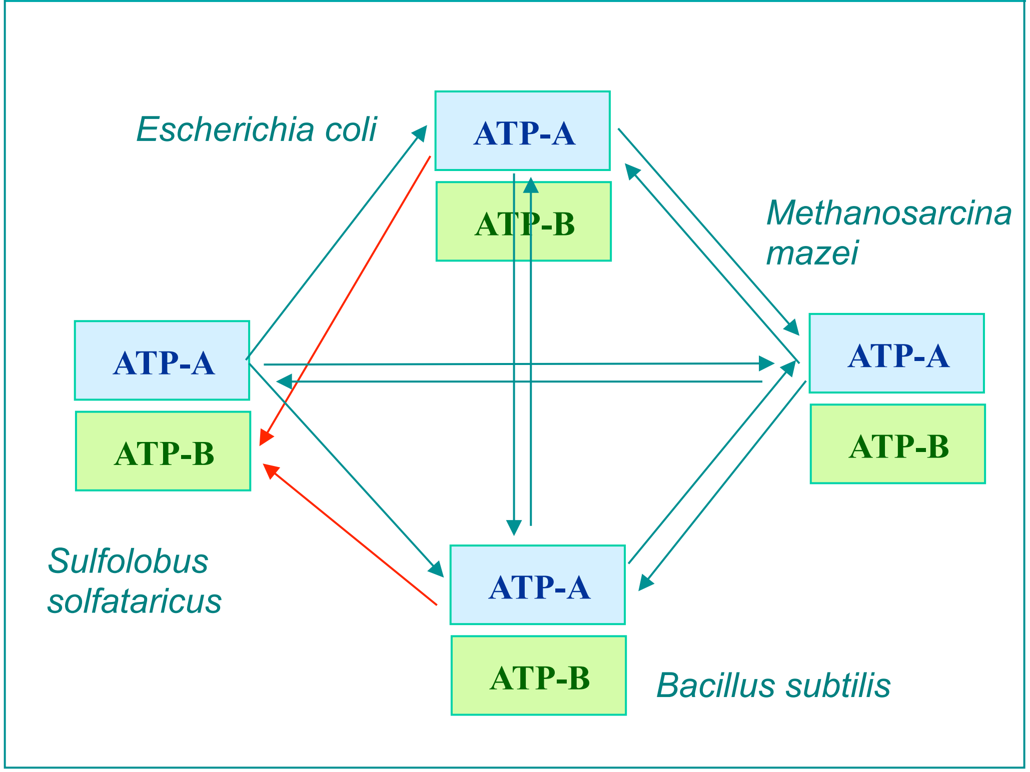
0 gene family

often fails in the presence of paralogs

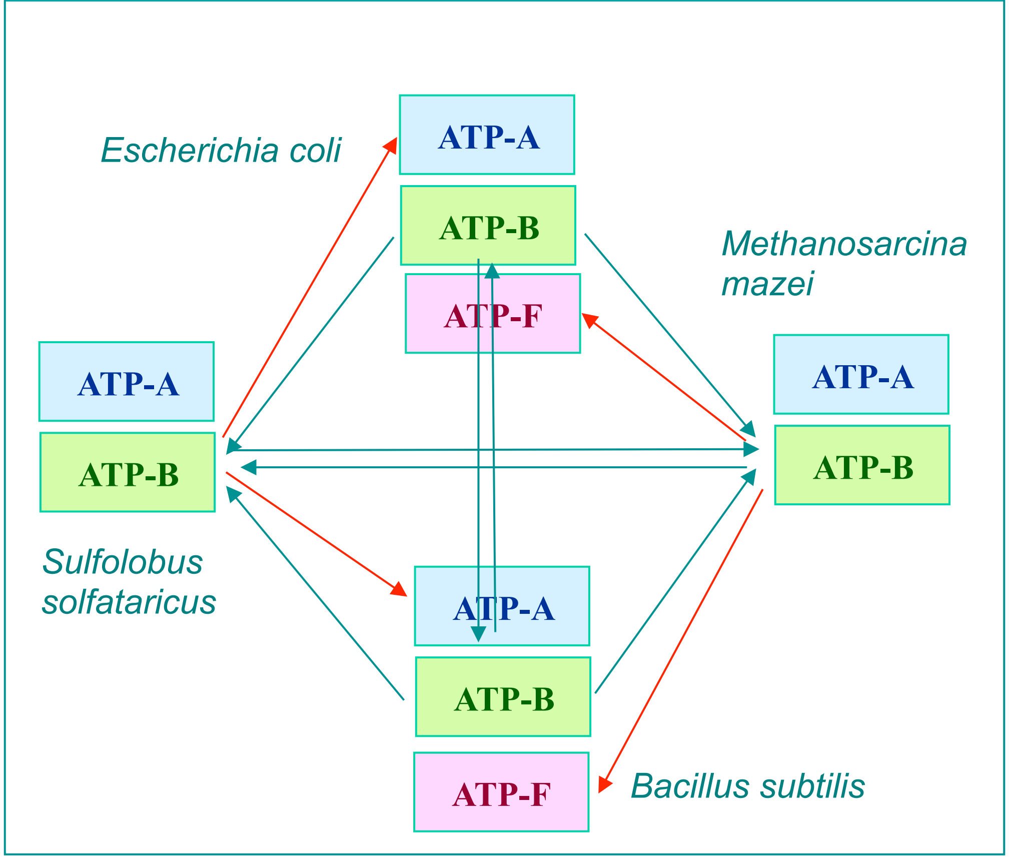
Families of ATP-synthases

Case of 2 bacteria and 2 archaea species

ATP-A (catalytic subunit)



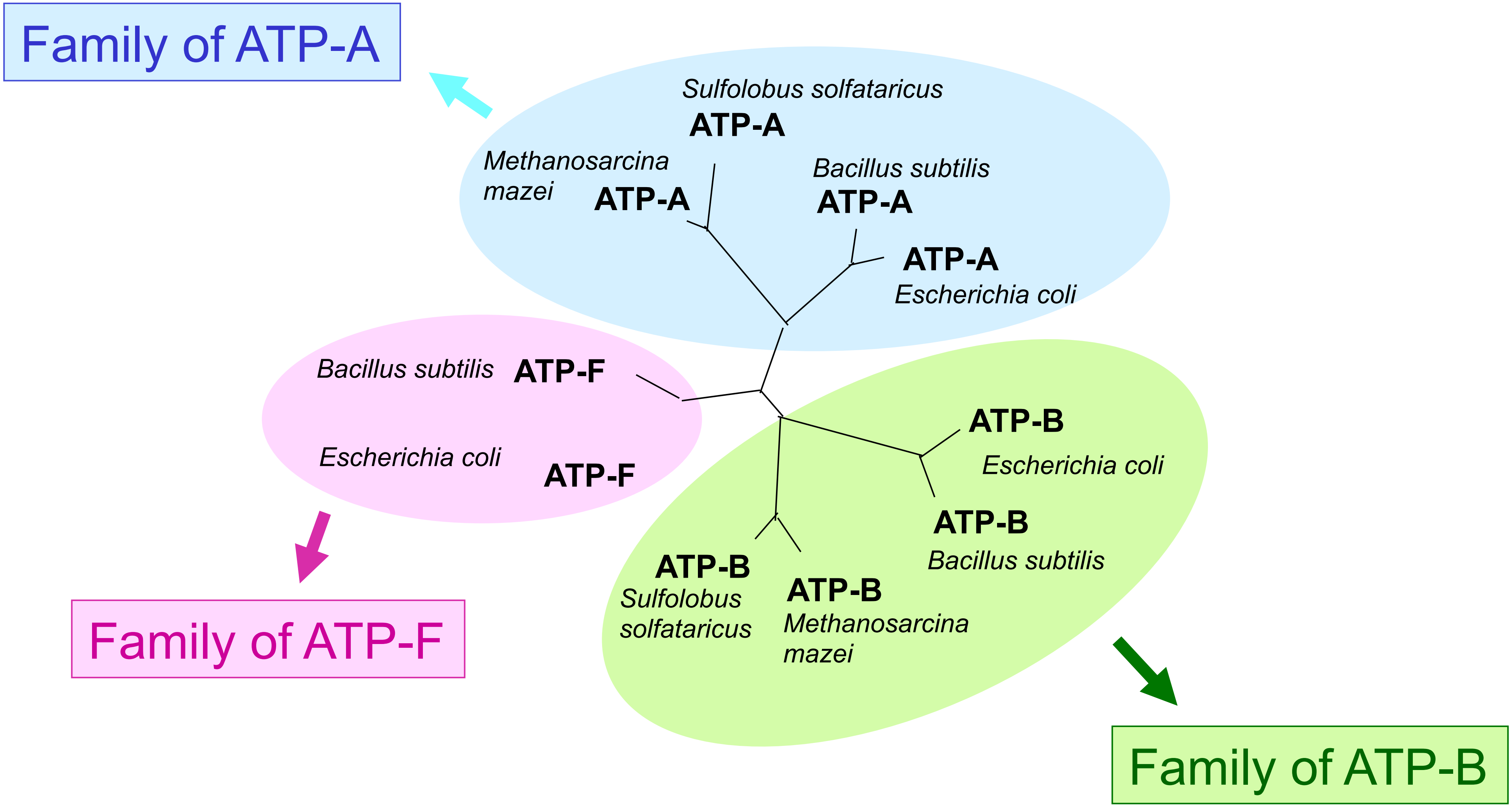
ATP-B (non-catalytic subunit)



Neither ATP-A nor ATB-B is selected by RBH method

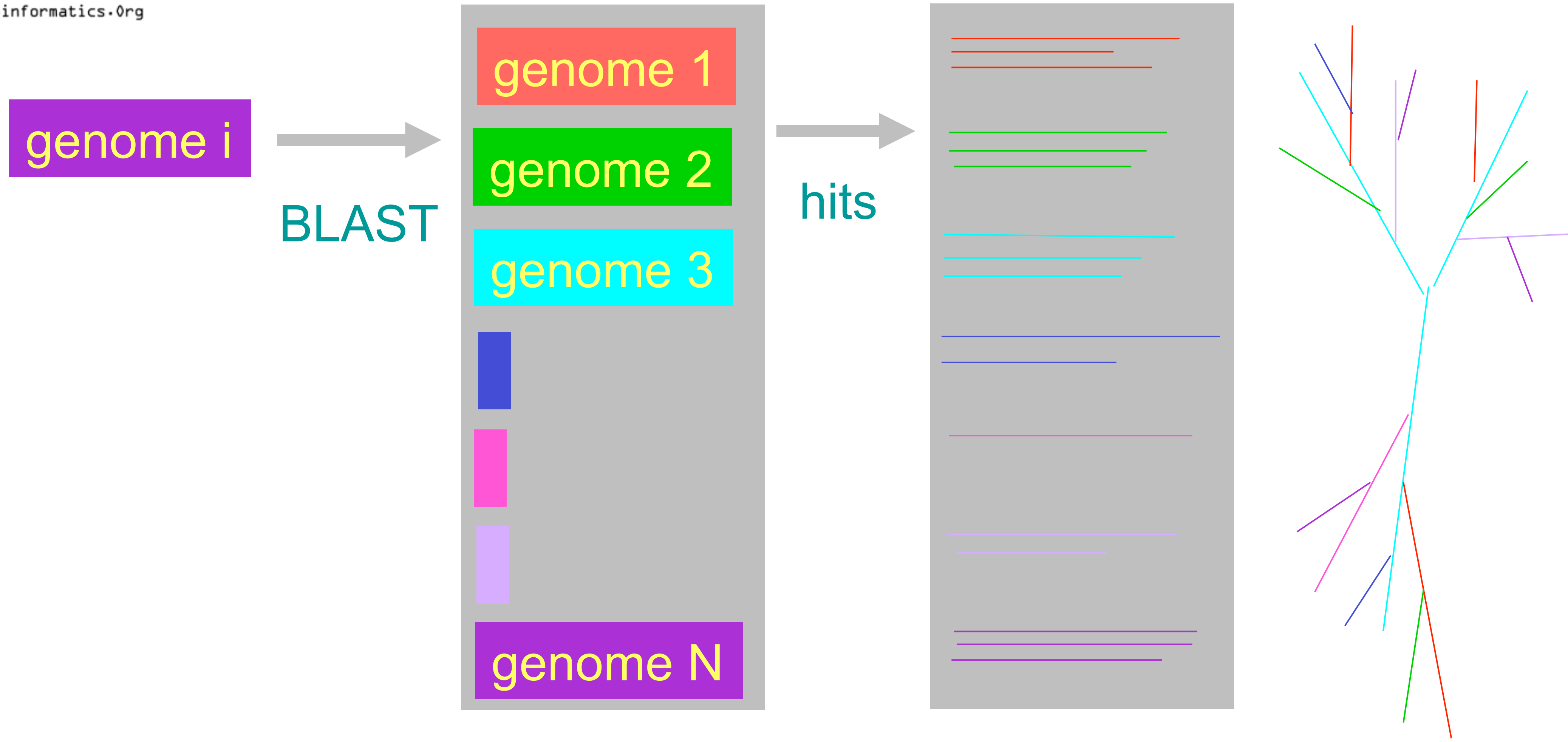
Families of ATP-synthases

Phylogenetic Tree



BranchClust Algorithm

Bioinformatics.Org



genome i

BLAST

genome 1

genome 2

genome 3

genome N

hits

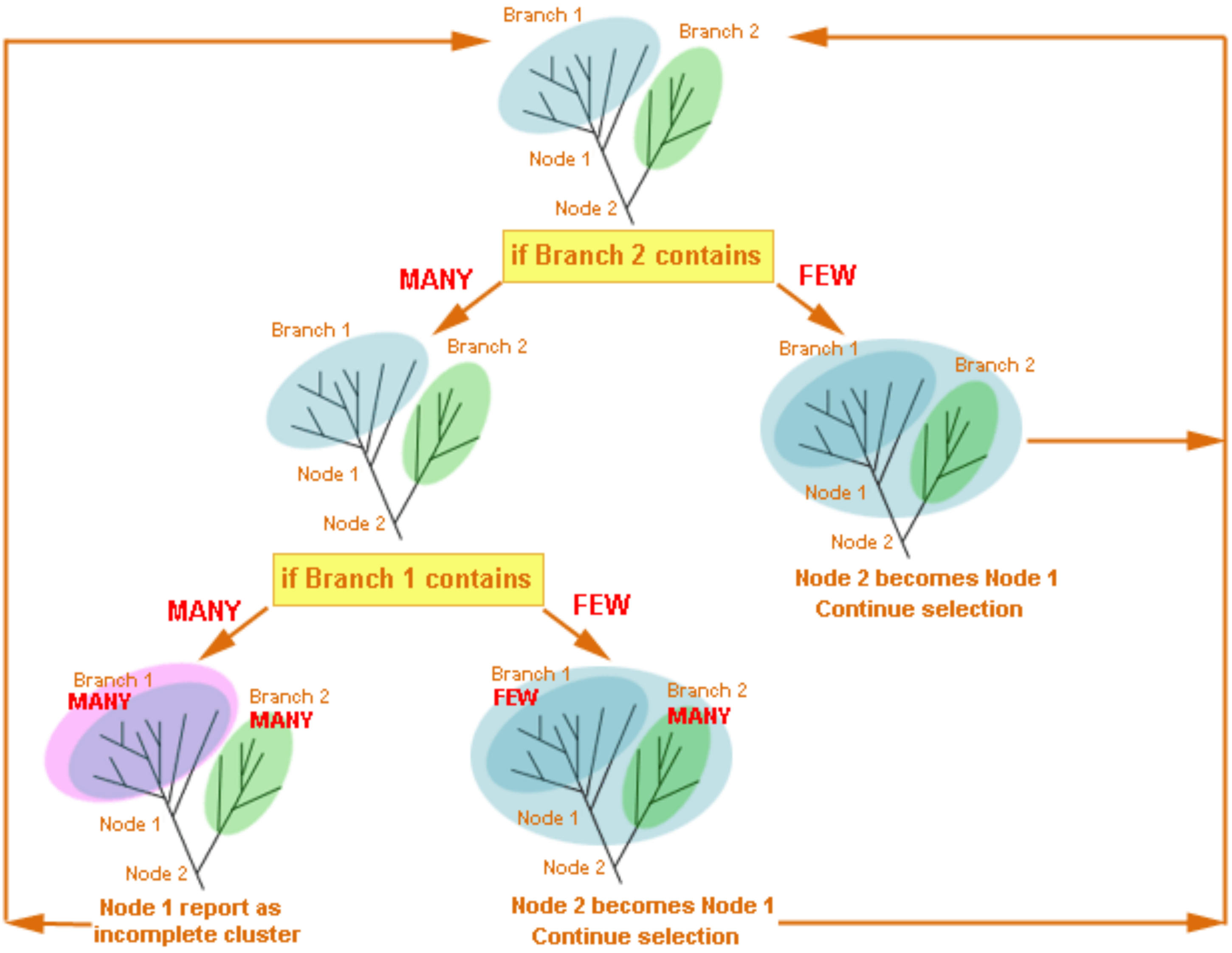
dataset of N genomes

superfamily

tree

BranchClust Algorithm

Bioinformatics.Org



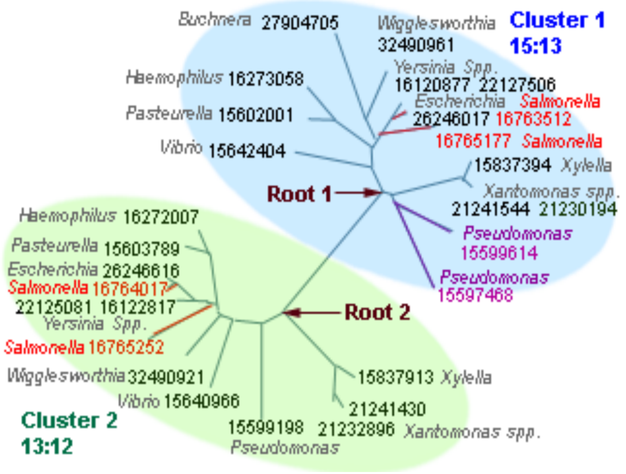
www.bioinformatics.org/branchclust

BranchClust Algorithm

Root positions

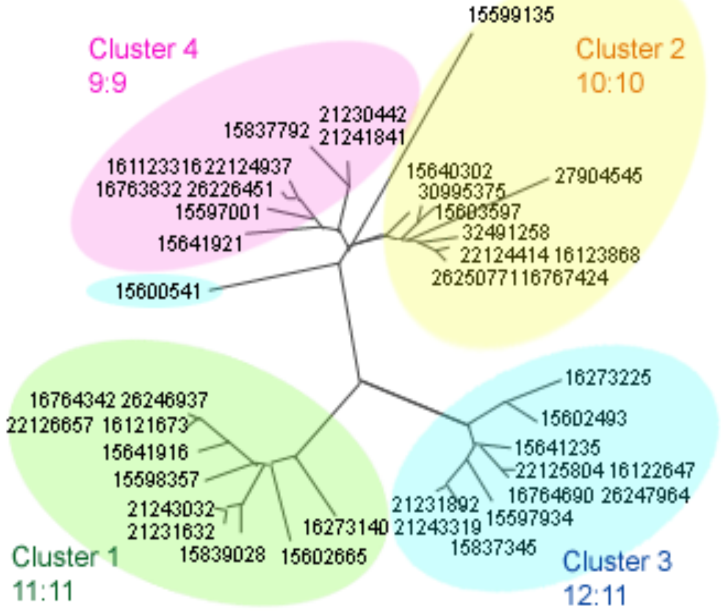
Superfamily of penicillin-binding protein

13 gamma proteo bacteria



Superfamily of DNA-binding protein

13 gamma proteo bacteria



Bioinformatics.Org

www.bioinformatics.org/branchclust

BranchClust Algorithm

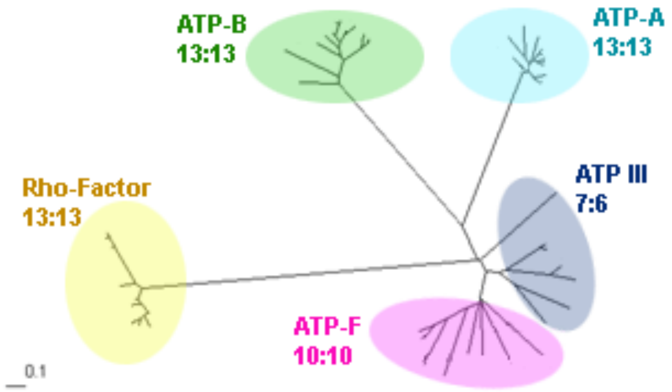
Comparison of the best BLAST hit method and BranchClust algorithm

Number of taxa - A: Archaea B: Bacteria	Number of selected families:	
	Reciprocal best BLAST hit	BranchClust
2A 2B	80	414 (all complete)
13B	236	409 (263 complete, 409 with n≥8)
16B 14A	12	126 (60 complete, 126 with n≥24).

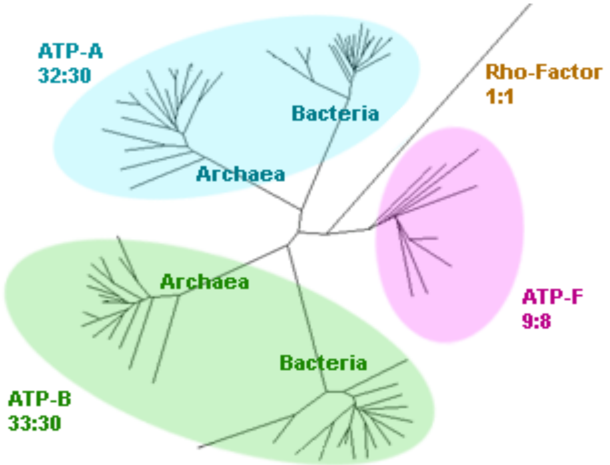
BranchClust Algorithm

ATP-synthases: Examples of Clustering

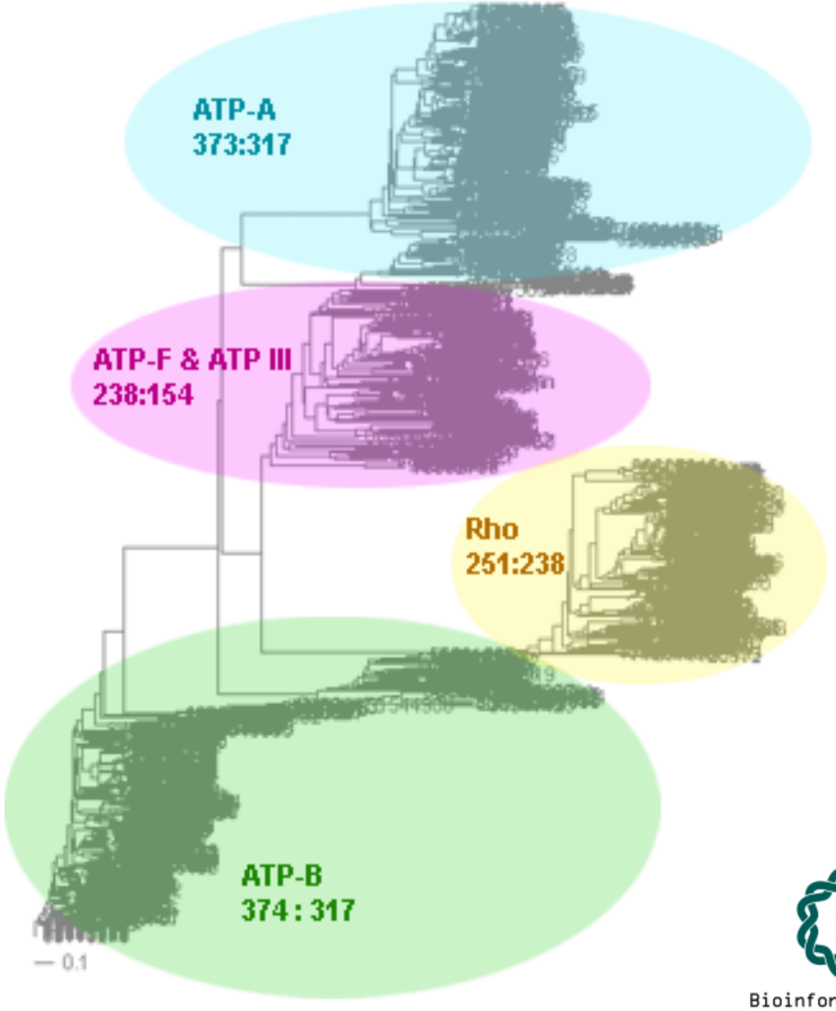
13 gamma proteobacteria



30 taxa: 16 bacteria and 14 archaea

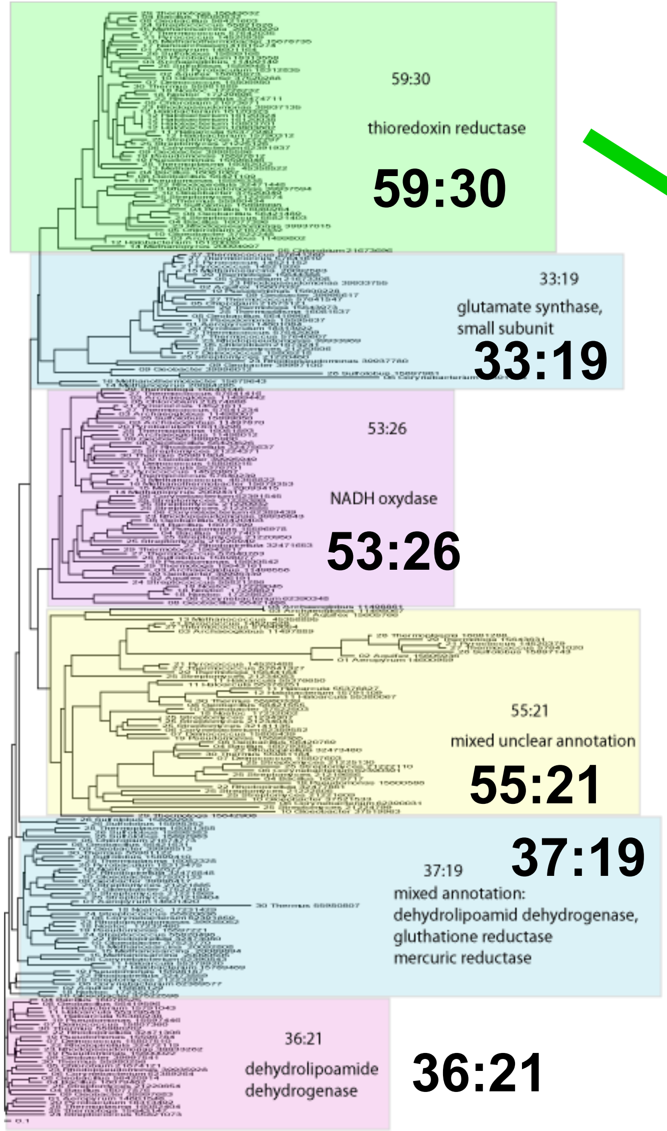


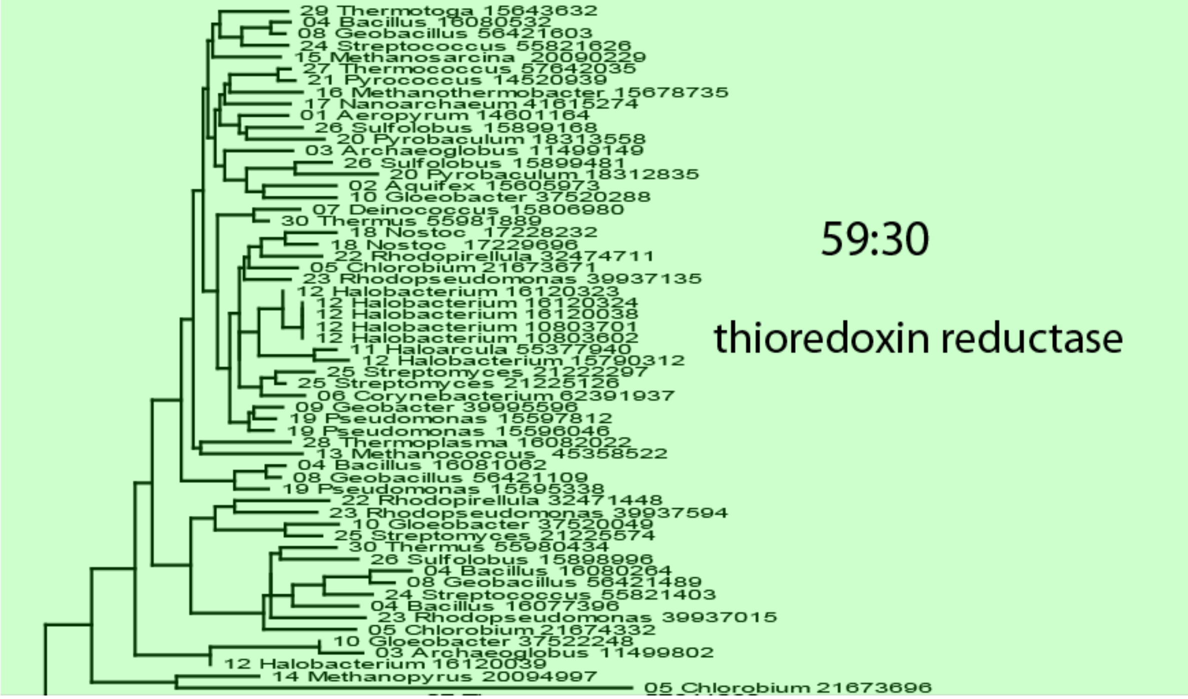
317 bacteria and archaea



BranchClust Algorithm

Typical Superfamily for 30 taxa (16 bacteria and 14 archaea)


59:30
thioredoxin reductase
59:30
33:19
glutamate synthase, small subunit
33:19
53:26
NADH oxydase
53:26
55:21
mixed unclear annotation
55:21
37:19
37:19
mixed annotation: dehydrolipoamid dehydrogenase, gluthatione reductase mercuric reductase
36:21
dehydrolipoamide dehydrogenase
36:21

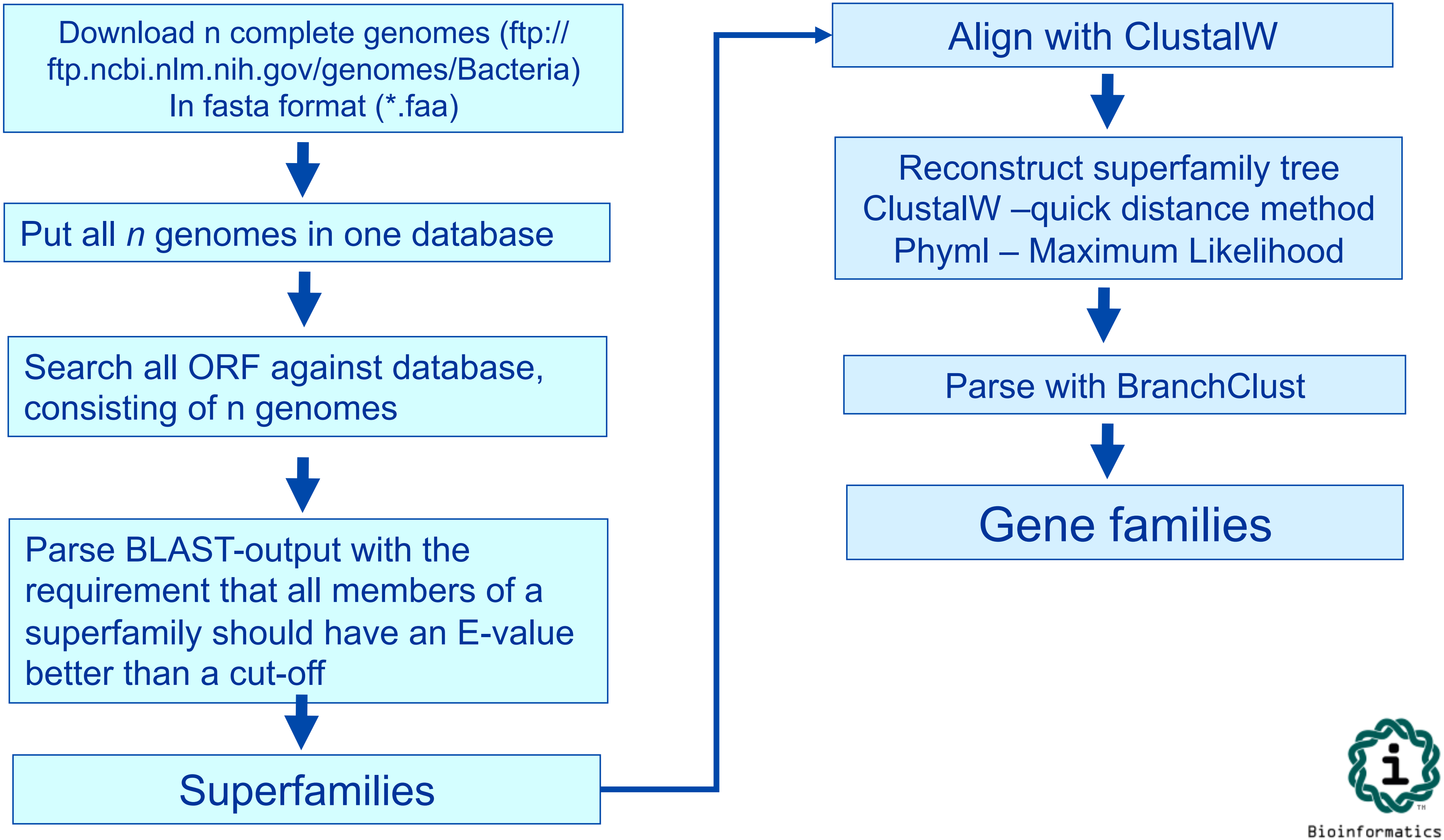


Bioinformatics.Org

www.bioinformatics.org/branchclust

BranchClust Algorithm

Data Flow



Download n complete genomes (ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria) In fasta format (*.faa)

Put all *n* genomes in one database

Search all ORF against database, consisting of n genomes

Parse BLAST-output with the requirement that all members of a superfamily should have an E-value better than a cut-off

Superfamilies

Align with ClustalW

Reconstruct superfamily tree
ClustalW –quick distance method
Phyml – Maximum Likelihood

Parse with BranchClust

Gene families

Bioinformatics.Org

BranchClust Algorithm

Implementation and Usage

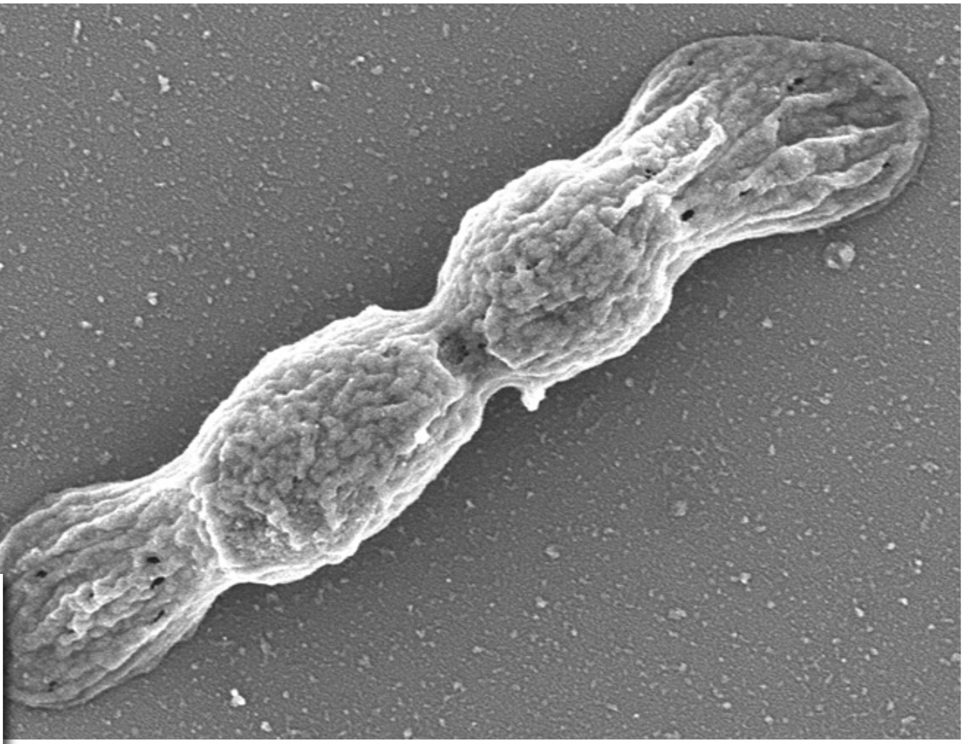
The BranchClust algorithm is implemented in Perl with the use of the BioPerl module for parsing trees and is freely available at http://bioinformatics.org/branchclust

Required:

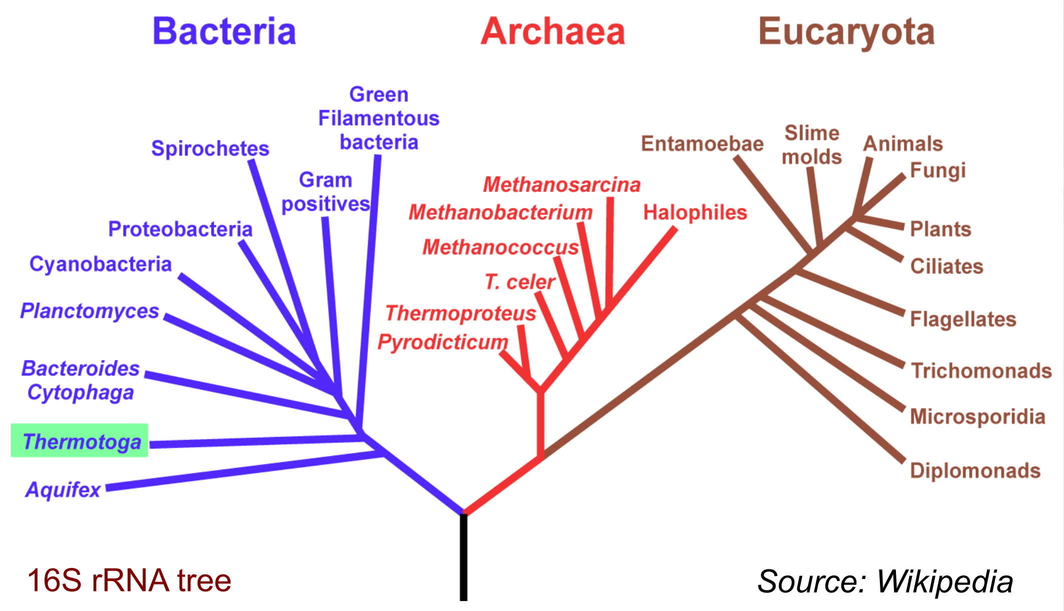
1.Bioperl module for parsing trees Bio::TreeIO
2. Taxa recognition file gi_numbers.out must be present in the current directory.
For information on how to create this file, read the Taxa recognition file section on the web-site.
3. Blastall from NCB needs to be installed.

Bioinformatics.Org

- *Thermotoga petrophila*
- *Thermotoga maritima*
- *Thermotoga* sp. strain RQ2
- *Thermotoga neapolitana*
- *Thermotoga naphthophila*



ga olearia. Courtesy of Kenneth Noll, UConn



16S rRNA tree

Source: Wikipedia

Olga Zhaxybayeva, Kristen S. Swithers, Pascal Lapierre, Gregory P. Fournier, Derek M. Bickhart, Robert T. DeBoy, Karen E. Nelson, Camilla L. Nesbø, W. Ford Doolittle, J. Peter Gogarten, and Kenneth M. Noll·
“On the Chimeric Nature, Thermophilic Origin and Phylogenetic Placement of the Thermotogales”, *Proc Natl Acad Sci U S A.*, Online Early, March 23, 2009.

to use other genomes:

- The easiest source for other genomes is via anonymous ftp from ftp.ncbi.nlm.nih.gov

  Genomes are in the subfolder genomes.

  Bacterial and Archaeal genomes are in the subfolder Bacteria
- For use with BranchClust you want to retrieve the .faa files from the folders of the individual organisms (in case there are multiple .faa files, download them all and copy them into a single file).
- Copy the genomes into the fasta folder in directory where the branchclust scripts are.
- To create a table that links GI numbers to genomes run perl extract_gi_numbers.pl or

  ```
  qsub extract_gi_numbers.sh
  ```

to copy files and scripts into your folder

```
• mkdir workshop
• cd workshop
• mkdir test
• cp -R /Users/jpgogarten/workshop/test/
  * /Users/mcb221_unnn/workshop/test/
```

This should be one line, and mcb221_u1nnn should be replaced with the name of your home directory.

The –R tells UNIX to copy recursively (including subdirectories)

This command also copies a directory called fasta that contains 5 genomes to work on. If you want to work on different genomes, delete the 5 *.faa files that contain the genomes from the Thermotogales and replace them with the genomes of your choice. (“genomes” really means all the proteins encoded by ORFs present in the genome).

If you use other genomes you will need to generate a file that contains assignments between name of the ORF and the name of the genome. This file should be called gi_numbers.out

If your genomes follow the JGI convention, every ORF starts with a four letters designating the species folloed by 4 numbers identifying the particular ORF. In this case the file gi_numbers.out should look as follows. It should be straight forward to create this file by hand ☺

```
Thermotoga maritima |   Tmar.....
Thermotoga naphthophila |       Tnap.....
Thermotoga neapolitana |        Tnea.....
Thermotoga petrophila | Tpet.....
Thermotoga sp. RQ2 |    TRQ2.....
```

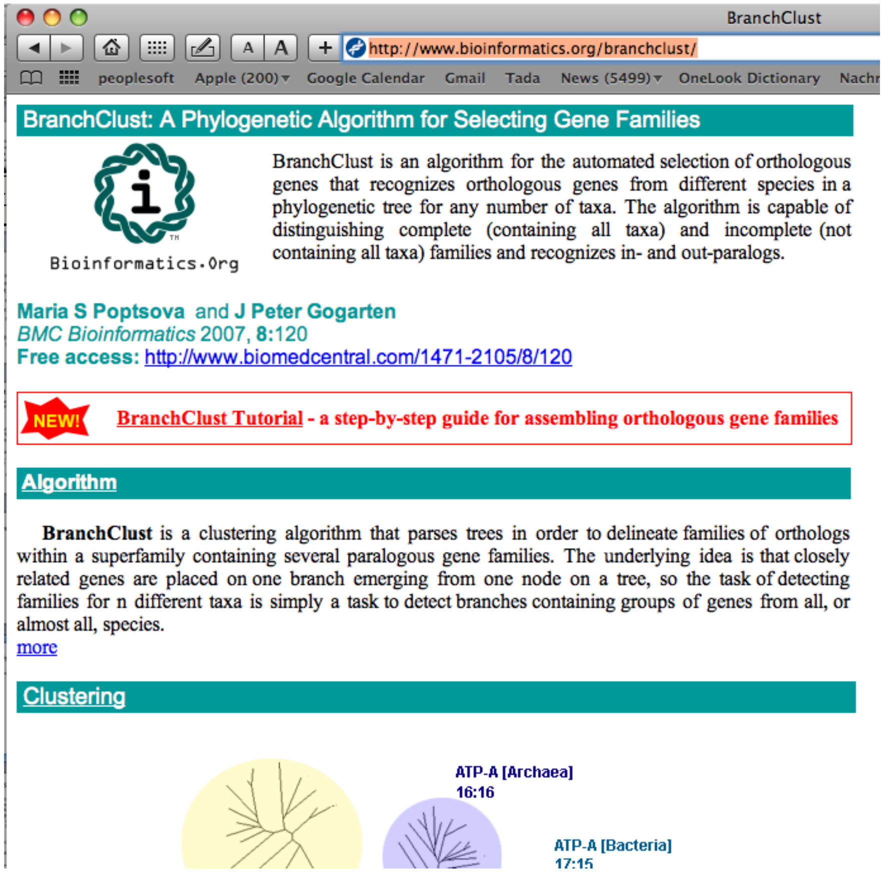
If your genomes conform to the NCBI *.faa convention, put the genomes into a subdirectory called fasta, and run the script extract_gi_numbers.pl in the parent directory. (Best is probably ~/workshop/test.)

The script should generate a log file and an output file called gi_numbers.out

```
Burkholderia phage Bcep781 |     2375....     4783....     1179.....
Enterobacteria phage K1F | 7711....
Enterobacteria phage N4 |  1199.....
Enterobacteria phage P22 | 5123....     9635...      1271....
       193433..
Enterobacteria phage RB43 |      6639....
Enterobacteria phage T1 |  4568....
Enterobacteria phage T3 |  1757....
Enterobacteria phage T5 |  4640....
Enterobacteria phage T7 |  9627...
Kluyvera phage Kvp1 |      2126.....
Lactobacillus phage phiAT3 |     4869....
Lactobacillus prophage Lj965 |   4117....
Lactococcus phage r1t |   2345....
Lactococcus phage sk1 |   9629...      193434..
Mycobacterium phage Bxz2 | 29566...
```

the branchclust scripts

- are available at http://www.bioinformatics.org/branchclust/
- A copy of the tutorial is in the folder you copied into your folder: BranchClustTutorial.pdf Consult the tutorial, if you want to use branchclust on other genomes.



BranchClust: A Phylogenetic Algorithm for Selecting Gene Families

BranchClust is an algorithm for the automated selection of orthologous genes that recognizes orthologous genes from different species in a phylogenetic tree for any number of taxa. The algorithm is capable of distinguishing complete (containing all taxa) and incomplete (not containing all taxa) families and recognizes in- and out-paralogs.

Maria S Poptsova and **J Peter Gogarten**
BMC Bioinformatics 2007, **8**:120
Free access: http://www.biomedcentral.com/1471-2105/8/120

BranchClust Tutorial - a step-by-step guide for assembling orthologous gene families

Algorithm

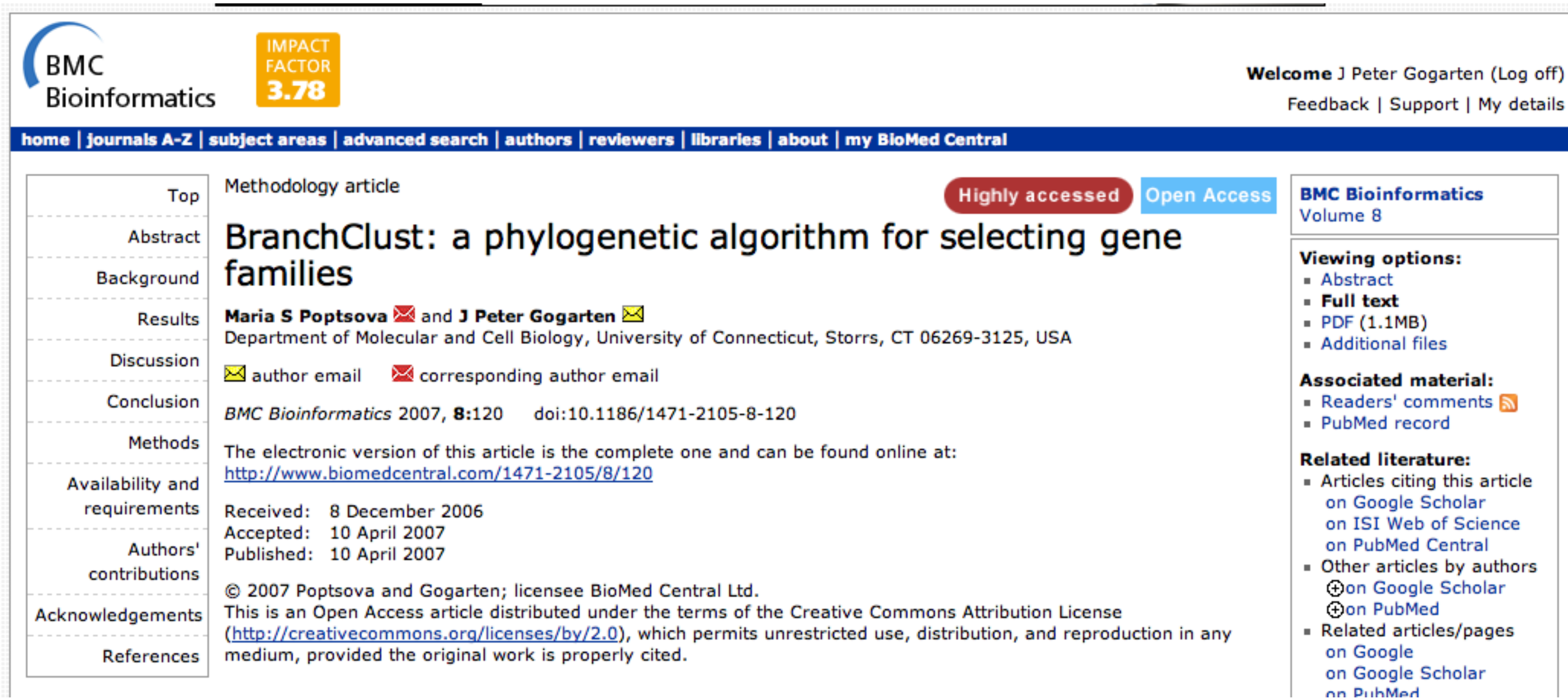
BranchClust is a clustering algorithm that parses trees in order to delineate families of orthologs within a superfamily containing several paralogous gene families. The underlying idea is that closely related genes are placed on one branch emerging from one node on a tree, so the task of detecting families for n different taxa is simply a task to detect branches containing groups of genes from all, or almost all, species.
more

Clustering

- The commands we use today are in a file in the test folder called *commands workshop tau one script*
This is a text file that you can open with any text editor.
(I use textwrangler on my mac, but you might want to use crimson)

BranchClust Article

- is available at http://www.biomedcentral.com/1471-2105/8/120


BMC Bioinformatics
IMPACT FACTOR 3.78
Welcome J Peter Gogarten (Log off)
Feedback | Support | My details
home | journals A-Z | subject areas | advanced search | authors | reviewers | libraries | about | my BioMed Central
Top
Abstract
Background
Results
Discussion
Conclusion
Methods
Availability and requirements
Authors' contributions
Acknowledgements
References
Highly accessed
Open Access

Methodology article

BranchClust: a phylogenetic algorithm for selecting gene families

Maria S Poptsova ✉ and **J Peter Gogarten** ✉
Department of Molecular and Cell Biology, University of Connecticut, Storrs, CT 06269-3125, USA

✉ author email ✉ corresponding author email

BMC Bioinformatics 2007, **8**:120 doi:10.1186/1471-2105-8-120

The electronic version of this article is the complete one and can be found online at: http://www.biomedcentral.com/1471-2105/8/120

Received: 8 December 2006
Accepted: 10 April 2007
Published: 10 April 2007

Create super families, alignments and trees

```
vi do_blast.pl
# to see what the parameters are doing type blastall  or
# bastall | more  at the commandline.
# If you move this to a different computer you might need to change a 2 to
a 1
```

```
vi parse_blast_cutoff_thermotoga.pl
# change bioperl directory; change cutoff E-value
# the script as written uses the bioperl library in my home directory
# Note: if using closely related genomes, you can cut back on the
#    size of the superfamilies by using a smaller E-value
# (if you genomes have normal GI numbers, use
# vi parse_blast_cutoff1.pl)
```

```
# check output:
more parsed/all_vs_all.parsed  ### type q to leave more
more parsed/all_vs_all.parsed | wc -l
# checks for number of lines=super famiies  output
```

Super Families to Trees

- `perl parse_superfamilies_singlelink.pl 1`
  # 1 gives the minimum size of the superfamily
- `perl prepare_fa_thermotoga.pl parsed/all_vs_all.fam`
  Creates a multiple fasta file for each superfamily
- `perl do_clustalw_aln.pl`
  aligns sequences using clustalw
- `perl do_clustalw_dist_kimura.pl`
  calcualtes trees using Kimura distances for all families in fa
  #trees stored in trees Check #1, 106, 1027, 111
- `perl prepare_trees.pl`
  reformats trees

Branchclust

```
perl branchclust_all_thermotoga.pl 2
 # Parameter 2 (MANY) says that a family needs to have
 # at least 2 members.
```

```
make_clusterlist.sh
# runs perl make_fam_list_inpar.pl 5 4 0
# results in test called families_inpar_5_4_0.list
# 5: number of genomes;
# 4: number of genomes in cluster ;
# 0: number of inparalogs
# (a 1 returns all the families with exactly 1 inparalog)
# you could add additional lines to the shell script:
# perl make_fam_list_inpar.pl 5 4 1
```

Process Branchclust output

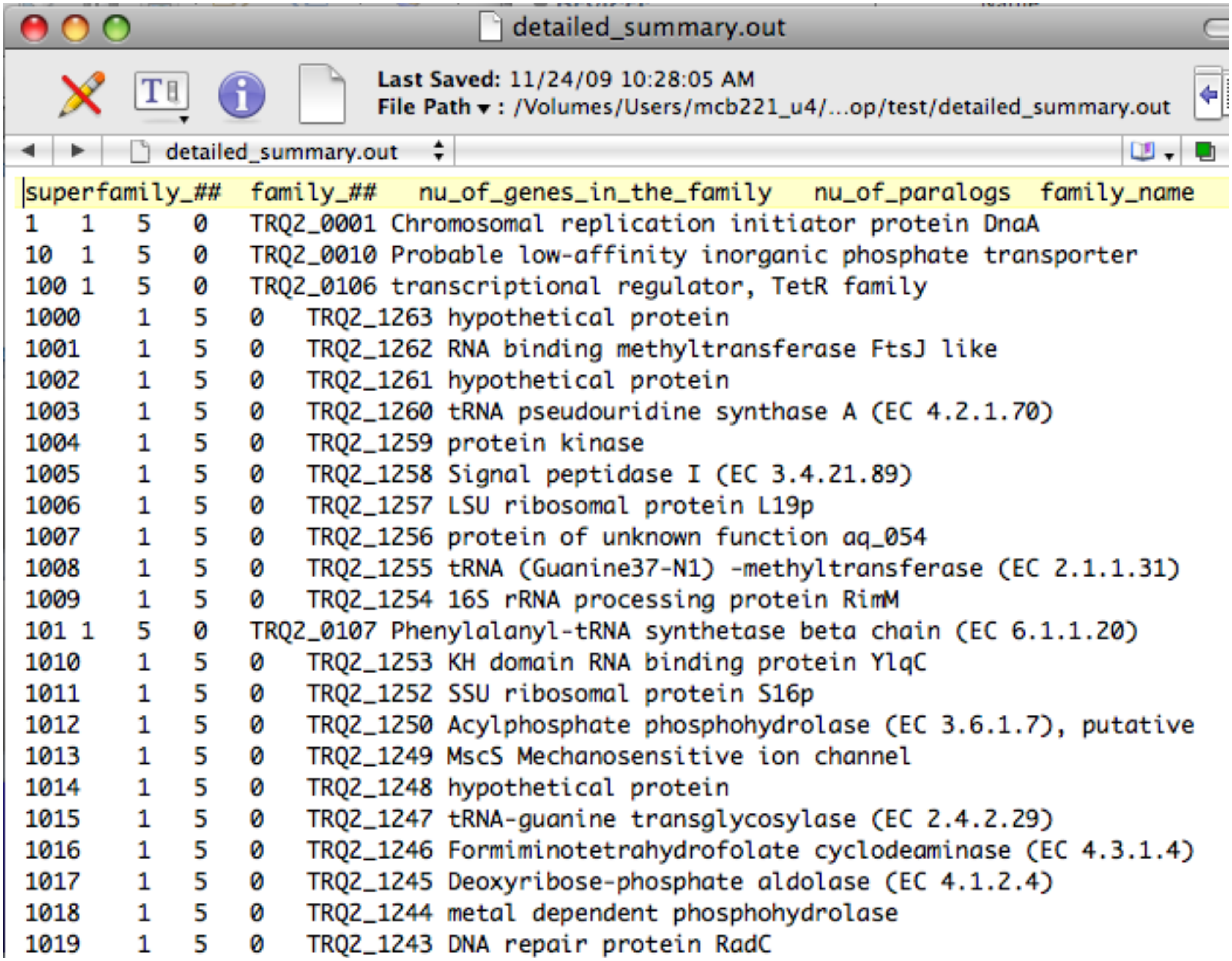
```
perl names_for_cluster_all.pl
#  (Parses clusters and attaches names.
#  Results in sub directory clusters. List in test)

perl summary.pl
# (makes list of number of complete and incomplete families
# file is stored in test)

perl detailed_summary_dashes.pl
# (result in test: detailed_summary.out - can be used in Excel)

perl prepare_bcfam_thermotoga.pl families_inpar_5_4_0.list
#(writes multiple fasta files into bcfam subdirectory.
# Can be used for alignment and phylogenetic reconstruction)
```

Summary Output

done with many = 3 and E-value cut-off of 10^{-25}

- `complete: 1564`
- `incomplete: 248`
- `total: 1812`
- `------ details -------`
- `incomplete 4: 87`
- `incomplete 3: 53`
- `incomplete 2: 66`
- `incomplete 1: 42`

Detailed Summary in Text Wrangler



Last Saved: 11/24/09 10:28:05 AM
File Path ▾ : /Volumes/Users/mcb221_u4/...op/test/detailed_summary.out

```
superfamily_##  family_##   nu_of_genes_in_the_family   nu_of_paralogs  family_name
1   1   5   0   TRQ2_0001 Chromosomal replication initiator protein DnaA
10  1   5   0   TRQ2_0010 Probable low-affinity inorganic phosphate transporter
100 1   5   0   TRQ2_0106 transcriptional regulator, TetR family
1000    1   5   0   TRQ2_1263 hypothetical protein
1001    1   5   0   TRQ2_1262 RNA binding methyltransferase FtsJ like
1002    1   5   0   TRQ2_1261 hypothetical protein
1003    1   5   0   TRQ2_1260 tRNA pseudouridine synthase A (EC 4.2.1.70)
1004    1   5   0   TRQ2_1259 protein kinase
1005    1   5   0   TRQ2_1258 Signal peptidase I (EC 3.4.21.89)
1006    1   5   0   TRQ2_1257 LSU ribosomal protein L19p
1007    1   5   0   TRQ2_1256 protein of unknown function aq_054
1008    1   5   0   TRQ2_1255 tRNA (Guanine37-N1) -methyltransferase (EC 2.1.1.31)
1009    1   5   0   TRQ2_1254 16S rRNA processing protein RimM
101 1   5   0   TRQ2_0107 Phenylalanyl-tRNA synthetase beta chain (EC 6.1.1.20)
1010    1   5   0   TRQ2_1253 KH domain RNA binding protein YlqC
1011    1   5   0   TRQ2_1252 SSU ribosomal protein S16p
1012    1   5   0   TRQ2_1250 Acylphosphate phosphohydrolase (EC 3.6.1.7), putative
1013    1   5   0   TRQ2_1249 MscS Mechanosensitive ion channel
1014    1   5   0   TRQ2_1248 hypothetical protein
1015    1   5   0   TRQ2_1247 tRNA-guanine transglycosylase (EC 2.4.2.29)
1016    1   5   0   TRQ2_1246 Formiminotetrahydrofolate cyclodeaminase (EC 4.3.1.4)
1017    1   5   0   TRQ2_1245 Deoxyribose-phosphate aldolase (EC 4.1.2.4)
1018    1   5   0   TRQ2_1244 metal dependent phosphohydrolase
1019    1   5   0   TRQ2_1243 DNA repair protein RadC
```

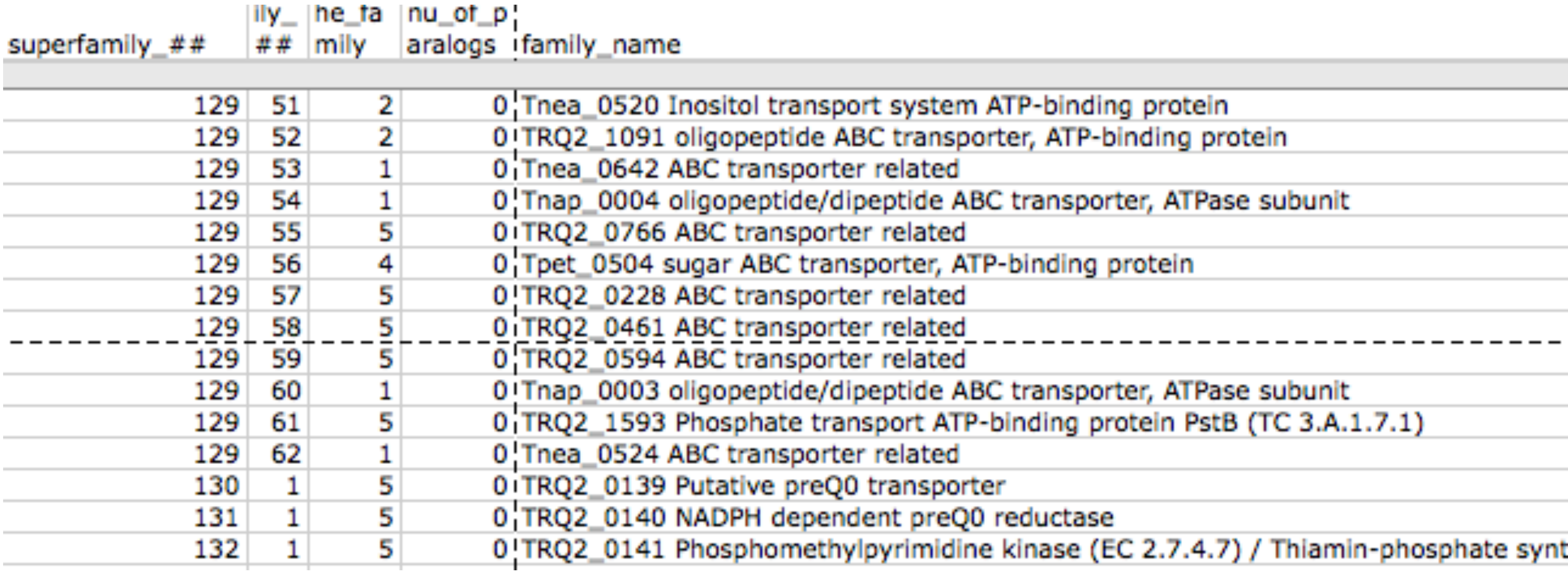
Detailed Summary in Excel

- copy detailed summary out onto your computer
- In EXEL Menu: Data -> get external data -> import text file -> in English version use defaults for other options.
- In EXEL Menu: Data -> sort -> sort by “superfamily number”-> if asked, check expand selection
- Scrolling down the list, search for a superfamily that was broken down into many families.

Do the families that were part of a superfamily have similar annotation lines?

How many of the families were complete?

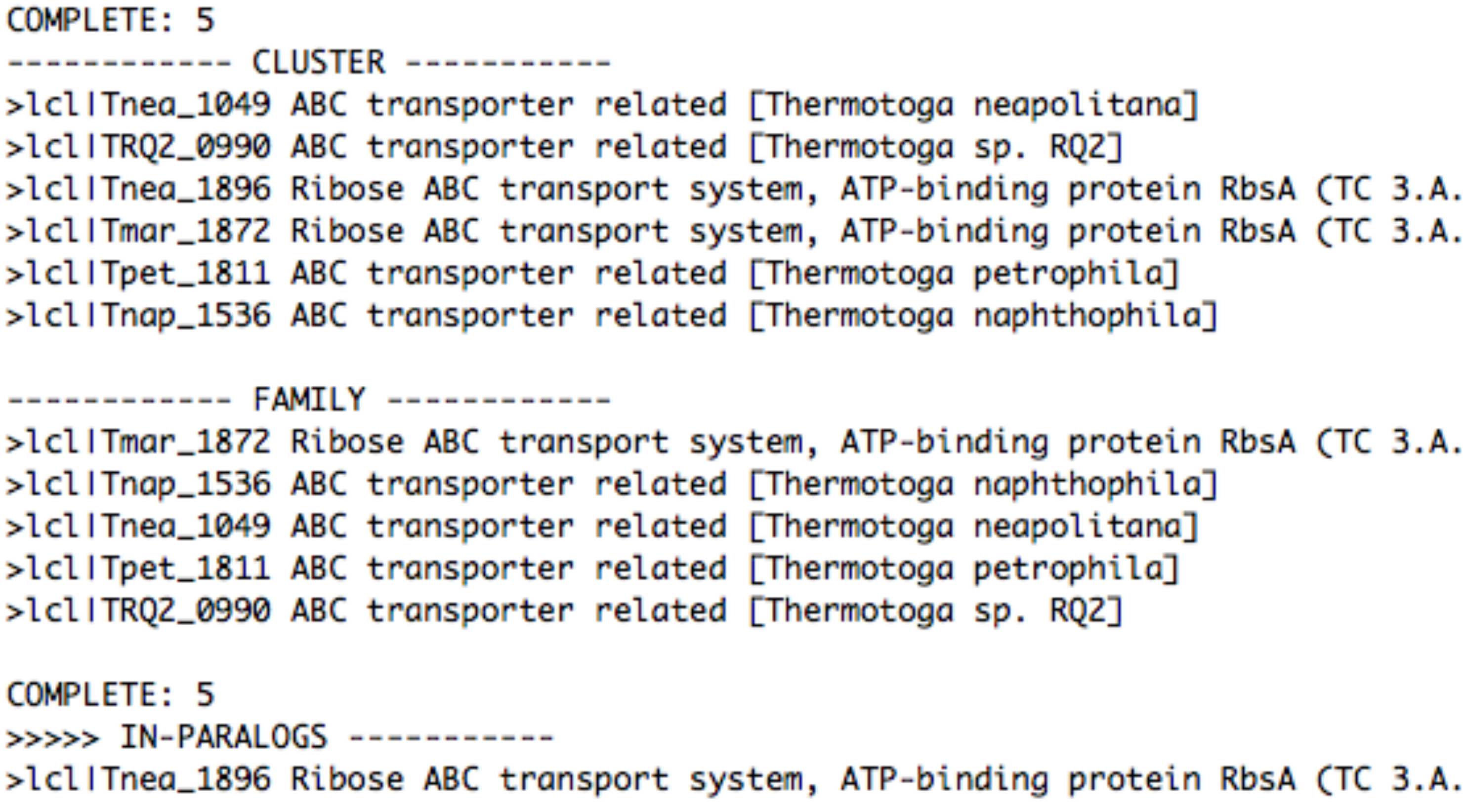
Do any have inparalogs? Take note of a few super families.



	ily_	he_fa	nu_of_p	
superfamily_##	##	mily	aralogs	family_name
129	51	2	0	Tnea_0520 Inositol transport system ATP-binding protein
129	52	2	0	TRQ2_1091 oligopeptide ABC transporter, ATP-binding protein
129	53	1	0	Tnea_0642 ABC transporter related
129	54	1	0	Tnap_0004 oligopeptide/dipeptide ABC transporter, ATPase subunit
129	55	5	0	TRQ2_0766 ABC transporter related
129	56	4	0	Tpet_0504 sugar ABC transporter, ATP-binding protein
129	57	5	0	TRQ2_0228 ABC transporter related
129	58	5	0	TRQ2_0461 ABC transporter related
129	59	5	0	TRQ2_0594 ABC transporter related
129	60	1	0	Tnap_0003 oligopeptide/dipeptide ABC transporter, ATPase subunit
129	61	5	0	TRQ2_1593 Phosphate transport ATP-binding protein PstB (TC 3.A.1.7.1)
129	62	1	0	Tnea_0524 ABC transporter related
130	1	5	0	TRQ2_0139 Putative preQ0 transporter
131	1	5	0	TRQ2_0140 NADPH dependent preQ0 reductase
132	1	5	0	TRQ2_0141 Phosphomethylpyrimidine kinase (EC 2.7.4.7) / Thiamin-phosphate synt

clusters/clusters_NNN.out.names

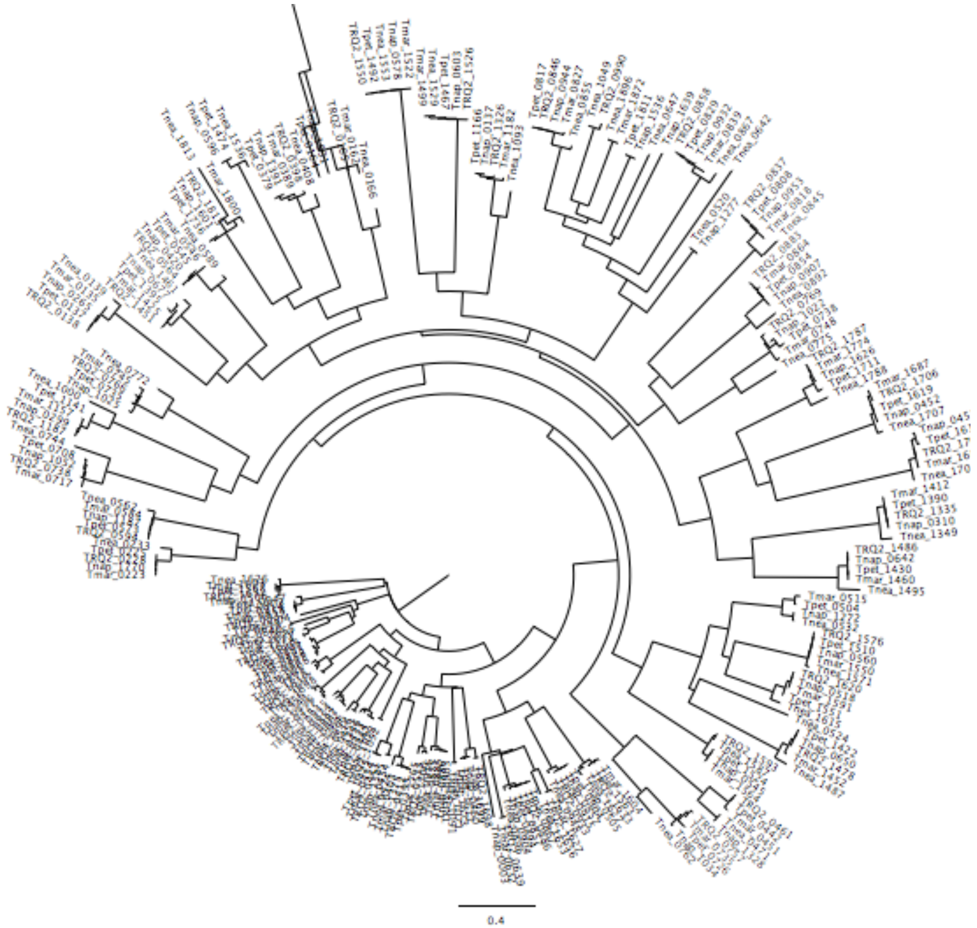
- Check a superfamily of your choice.
  Within a family, are all the annotation lines uniform?
- Within this report, if there are inparalogs, one is listed as a family member, the other one as inparalog. This is an arbitrary choice, both inparalogs from the same genome should be considered as being part of of the family.
- Out of cluster paralogs are paralogs that did not make it into a cluster with “many” genomes.

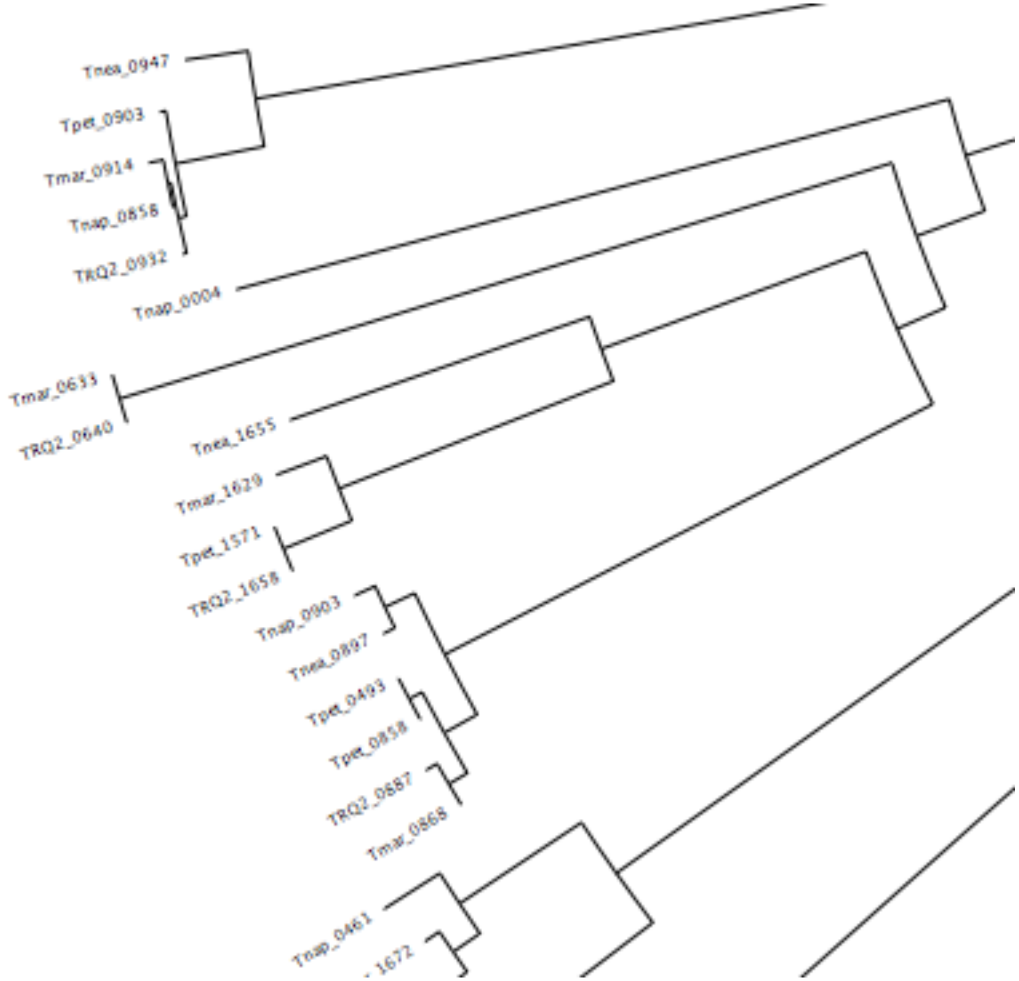


```
COMPLETE: 5
------------ CLUSTER -----------
>lcl|Tnea_1049 ABC transporter related [Thermotoga neapolitana]
>lcl|TRQ2_0990 ABC transporter related [Thermotoga sp. RQ2]
>lcl|Tnea_1896 Ribose ABC transport system, ATP-binding protein RbsA (TC 3.A.
>lcl|Tmar_1872 Ribose ABC transport system, ATP-binding protein RbsA (TC 3.A.
>lcl|Tpet_1811 ABC transporter related [Thermotoga petrophila]
>lcl|Tnap_1536 ABC transporter related [Thermotoga naphthophila]

------------ FAMILY ------------
>lcl|Tmar_1872 Ribose ABC transport system, ATP-binding protein RbsA (TC 3.A.
>lcl|Tnap_1536 ABC transporter related [Thermotoga naphthophila]
>lcl|Tnea_1049 ABC transporter related [Thermotoga neapolitana]
>lcl|Tpet_1811 ABC transporter related [Thermotoga petrophila]
>lcl|TRQ2_0990 ABC transporter related [Thermotoga sp. RQ2]

COMPLETE: 5
>>>>> IN-PARALOGS -----------
>lcl|Tnea_1896 Ribose ABC transport system, ATP-binding protein RbsA (TC 3.A.
```

trees/fam_XYZ.tre

- Check the tree for a superfamily of your choice. Copy the file to your computer and open it in TreeView, NJPLOT, or FigTree (check with your neighbor on which program works).
- For at least one cluster, in the tree, check if branchclust came to the same conclusion you would have reached





`prepare_bcfam_thermotoga.pl families_inpar_5_4_0.list`

The script prepare_bcfam_thermotoga.pl takes a list of families (created by `make_fam_list_inpar.pl`) and for each family retrieves the fasta sequences from the combined genome databank and stores the sequences in the BCfam folder, one multiple sequence file per family.

One possibility for further evaluation is to take multiple sequence files, align the sequences and perform a phylogenetic reconstruction (including boostrap analysis) using programs like phyml or Raxml.

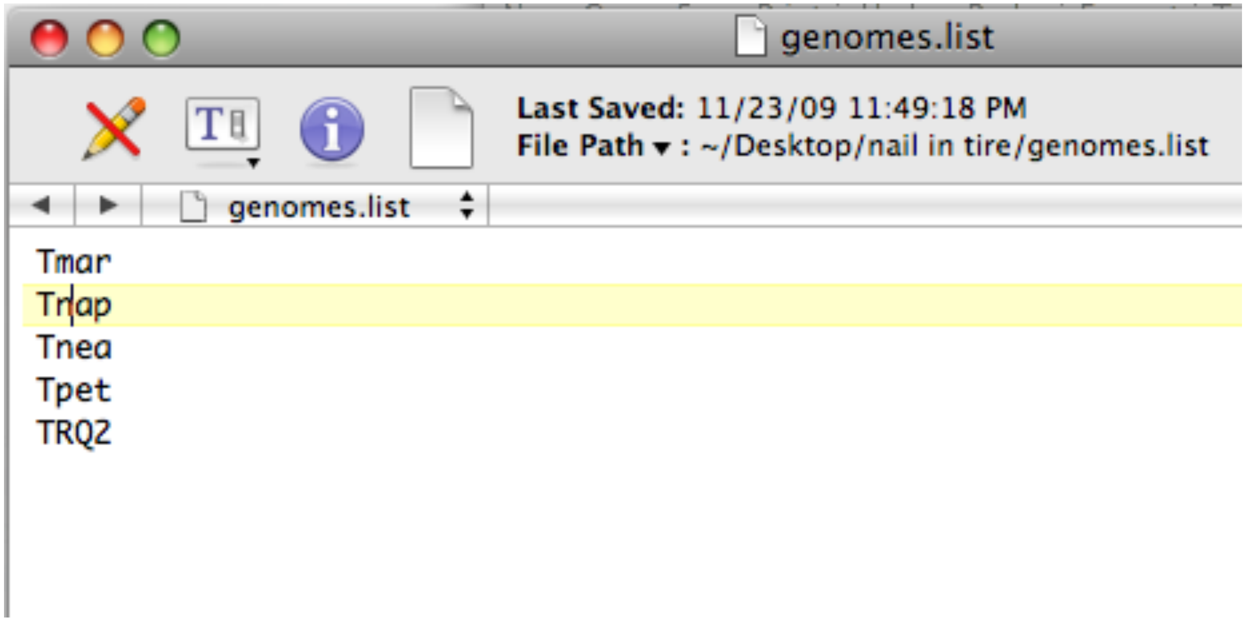
The resulting trees can be analyzed by decomposition and supertree approaches.

The Quartet Decomposition Server

http://csbl1.bmb.uga.edu/QD/phytree.php

Input A):

a file listing the names of genomes: E.g.:



Last Saved: 11/23/09 11:49:18 PM
File Path ▾ : ~/Desktop/nail in tire/genomes.list

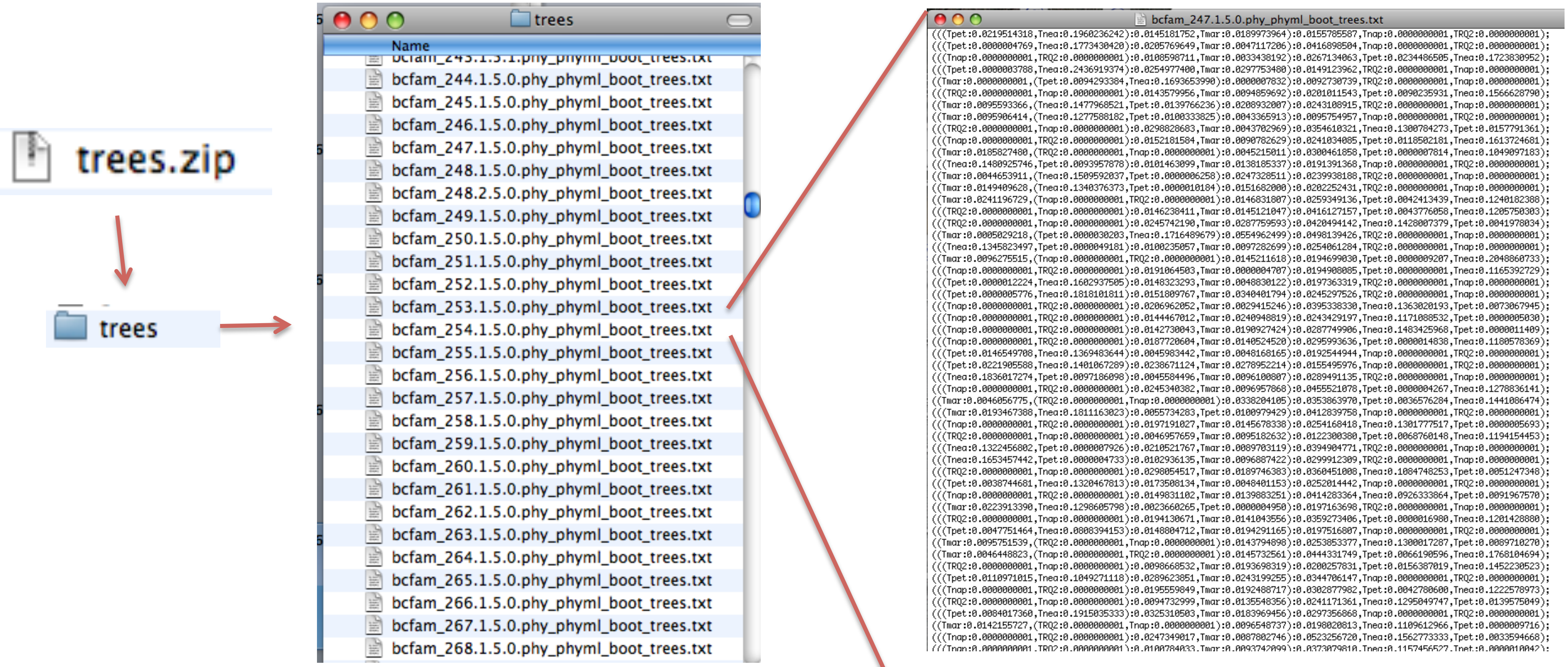
```
Tmar
Tnap
Tnea
Tpet
TRQ2
```

The Quartet Decomposition Server

http://csbl1.bmb.uga.edu/QD/phytree.php

Input B):

An Archive of files where every file contains all the trees that resulted from a bootstrap analysis of one gene family:



trees

Name

```
bcfam_243.1.5.1.phy_phyml_boot_trees.txt
bcfam_244.1.5.0.phy_phyml_boot_trees.txt
bcfam_245.1.5.0.phy_phyml_boot_trees.txt
bcfam_246.1.5.0.phy_phyml_boot_trees.txt
bcfam_247.1.5.0.phy_phyml_boot_trees.txt
bcfam_248.1.5.0.phy_phyml_boot_trees.txt
bcfam_248.2.5.0.phy_phyml_boot_trees.txt
bcfam_249.1.5.0.phy_phyml_boot_trees.txt
bcfam_250.1.5.0.phy_phyml_boot_trees.txt
bcfam_251.1.5.0.phy_phyml_boot_trees.txt
bcfam_252.1.5.0.phy_phyml_boot_trees.txt
bcfam_253.1.5.0.phy_phyml_boot_trees.txt
bcfam_254.1.5.0.phy_phyml_boot_trees.txt
bcfam_255.1.5.0.phy_phyml_boot_trees.txt
bcfam_256.1.5.0.phy_phyml_boot_trees.txt
bcfam_257.1.5.0.phy_phyml_boot_trees.txt
bcfam_258.1.5.0.phy_phyml_boot_trees.txt
bcfam_259.1.5.0.phy_phyml_boot_trees.txt
bcfam_260.1.5.0.phy_phyml_boot_trees.txt
bcfam_261.1.5.0.phy_phyml_boot_trees.txt
bcfam_262.1.5.0.phy_phyml_boot_trees.txt
bcfam_263.1.5.0.phy_phyml_boot_trees.txt
bcfam_264.1.5.0.phy_phyml_boot_trees.txt
bcfam_265.1.5.0.phy_phyml_boot_trees.txt
bcfam_266.1.5.0.phy_phyml_boot_trees.txt
bcfam_267.1.5.0.phy_phyml_boot_trees.txt
bcfam_268.1.5.0.phy_phyml_boot_trees.txt
```

bcfam_247.1.5.0.phy_phyml_boot_trees.txt

```
(((Tpet:0.0219514318,Tnea:0.1960236242):0.0145181752,Tmar:0.0189973964):0.0155785587,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tpet:0.0000004769,Tnea:0.1773430420):0.0205769649,Tmar:0.0047117206):0.0416898504,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0108598711,Tmar:0.0033438192):0.0267134063,Tpet:0.0234486505,Tnea:0.1723830952);
(((Tpet:0.0000003788,Tnea:0.2436919374):0.0254977400,Tmar:0.0297753480):0.0149123962,TRQ2:0.0000000001,Tnap:0.0000000001);
((Tmar:0.0000000001,(Tpet:0.0094293384,Tnea:0.1693653990):0.0000007832):0.0092730739,TRQ2:0.0000000001,Tnap:0.0000000001);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0143579956,Tmar:0.0094859692):0.0201011543,Tpet:0.0090235931,Tnea:0.1566628790);
((Tmar:0.0095593366,(Tnea:0.1477968521,Tpet:0.0139766236):0.0208932007):0.0243108915,TRQ2:0.0000000001,Tnap:0.0000000001);
((Tmar:0.0095906414,(Tnea:0.1277588182,Tpet:0.0100333825):0.0043365913):0.0095754957,Tnap:0.0000000001,TRQ2:0.0000000001);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0298828683,Tmar:0.0043702969):0.0354610321,Tnea:0.1300784273,Tpet:0.0157791361);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0152181584,Tmar:0.0090782629):0.0241034085,Tpet:0.0118502181,Tnea:0.1613724681);
((Tmar:0.0185827480,(TRQ2:0.0000000001,Tnap:0.0000000001):0.0045215011):0.0300461858,Tpet:0.0000007814,Tnea:0.1049097183);
(((Tnea:0.1480925746,Tpet:0.0093957878):0.0101463099,Tmar:0.0138185337):0.0191391368,Tnap:0.0000000001,TRQ2:0.0000000001);
((Tmar:0.0044653911,(Tnea:0.1509592037,Tpet:0.0000006258):0.0247328511):0.0239938188,TRQ2:0.0000000001,Tnap:0.0000000001);
((Tmar:0.0149409628,(Tnea:0.1340376373,Tpet:0.0000010184):0.0151682000):0.0202252431,TRQ2:0.0000000001,Tnap:0.0000000001);
((Tmar:0.0241196729,(Tnap:0.0000000001,TRQ2:0.0000000001):0.0146831807):0.0259349136,Tpet:0.0042413439,Tnea:0.1240182388);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0146238411,Tmar:0.0145121047):0.0416127157,Tpet:0.0043776058,Tnea:0.1205750303);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0245742190,Tmar:0.0287759593):0.0420494142,Tnea:0.1428007379,Tpet:0.0041978034);
((Tmar:0.0005029218,(Tpet:0.0000030203,Tnea:0.1716489679):0.0554962499):0.0498139426,TRQ2:0.0000000001,Tnap:0.0000000001);
(((Tnea:0.1345823497,Tpet:0.0000049181):0.0100235057,Tmar:0.0097282699):0.0254061284,TRQ2:0.0000000001,Tnap:0.0000000001);
((Tmar:0.0096275515,(Tnap:0.0000000001,TRQ2:0.0000000001):0.0145211618):0.0194699030,Tpet:0.0000009207,Tnea:0.2048860733);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0191064503,Tmar:0.0000004707):0.0194908085,Tpet:0.0000000001,Tnea:0.1165392729);
(((Tpet:0.0000012224,Tnea:0.1602937505):0.0148323293,Tmar:0.0048830122):0.0197363319,TRQ2:0.0000000001,Tnap:0.0000000001);
(((Tpet:0.0000005776,Tnea:0.1818101811):0.0151809767,Tmar:0.0340401794):0.0245297526,TRQ2:0.0000000001,Tnap:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0206962052,Tmar:0.0029415246):0.0395338330,Tnea:0.1363020193,Tpet:0.0073067945);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0144467012,Tmar:0.0240948819):0.0243429197,Tnea:0.1171088532,Tpet:0.0000005030);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0142730043,Tmar:0.0190927424):0.0287749906,Tnea:0.1483425968,Tpet:0.0000011409);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0187720604,Tmar:0.0140524520):0.0295993636,Tpet:0.0000014838,Tnea:0.1180578369);
(((Tpet:0.0146549708,Tnea:0.1369483644):0.0045983442,Tmar:0.0048168165):0.0192544944,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tpet:0.0221905588,Tnea:0.1401067289):0.0238671124,Tmar:0.0278952214):0.0155495976,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tnea:0.1836017274,Tpet:0.0097186098):0.0045584496,Tmar:0.0096100807):0.0289491135,TRQ2:0.0000000001,Tnap:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0245340382,Tmar:0.0096957868):0.0455521078,Tpet:0.0000004267,Tnea:0.1278836141);
((Tmar:0.0046056775,(TRQ2:0.0000000001,Tnap:0.0000000001):0.0338204105):0.0353863970,Tpet:0.0036576284,Tnea:0.1441086474);
(((Tmar:0.0193467388,Tnea:0.1811163023):0.0055734283,Tpet:0.0100979429):0.0412839758,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0197191027,Tmar:0.0145678338):0.0254168418,Tnea:0.1301777517,Tpet:0.0000005693);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0046957659,Tmar:0.0095182632):0.0122300380,Tpet:0.0068760148,Tnea:0.1194154453);
(((Tnea:0.1322456802,Tpet:0.0000007926):0.0210521767,Tmar:0.0089703119):0.0394904771,TRQ2:0.0000000001,Tnap:0.0000000001);
(((Tnea:0.1653457442,Tpet:0.0000004733):0.0102936135,Tmar:0.0096887422):0.0299912309,TRQ2:0.0000000001,Tnap:0.0000000001);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0298054517,Tmar:0.0189746383):0.0360451008,Tnea:0.1084748253,Tpet:0.0051247348);
(((Tpet:0.0038744681,Tnea:0.1320467813):0.0173508134,Tmar:0.0048401153):0.0252014442,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0149831102,Tmar:0.0139883251):0.0414283364,Tnea:0.0926333864,Tpet:0.0091967570);
(((Tmar:0.0223913390,Tnea:0.1298605798):0.0023660265,Tpet:0.0000004950):0.0197163698,TRQ2:0.0000000001,Tnap:0.0000000001);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0194130671,Tmar:0.0141043556):0.0359273406,Tpet:0.0000016980,Tnea:0.1201428880);
(((Tpet:0.0047751464,Tnea:0.0808394153):0.0148804712,Tmar:0.0194291165):0.0197516807,Tnap:0.0000000001,TRQ2:0.0000000001);
((Tmar:0.0095751539,(TRQ2:0.0000000001,Tnap:0.0000000001):0.0143794898):0.0253853377,Tnea:0.1300017287,Tpet:0.0089710270);
((Tmar:0.0046448823,(Tnap:0.0000000001,TRQ2:0.0000000001):0.0145732561):0.0444331749,Tpet:0.0066190596,Tnea:0.1768104694);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0098668532,Tmar:0.0193698319):0.0200257831,Tpet:0.0156387019,Tnea:0.1452230523);
(((Tpet:0.0110971015,Tnea:0.1049271118):0.0289623851,Tmar:0.0243199255):0.0344706147,Tnap:0.0000000001,TRQ2:0.0000000001);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0195559849,Tmar:0.0192488717):0.0302877982,Tpet:0.0042780600,Tnea:0.1222578973);
(((TRQ2:0.0000000001,Tnap:0.0000000001):0.0094732999,Tmar:0.0135548356):0.0241171361,Tnea:0.1295049747,Tpet:0.0139575049);
(((Tpet:0.0084017360,Tnea:0.1915035333):0.0325310503,Tmar:0.0183969456):0.0297356868,Tnap:0.0000000001,TRQ2:0.0000000001);
((Tmar:0.0142155727,(TRQ2:0.0000000001,Tnap:0.0000000001):0.0096548737):0.0198020813,Tnea:0.1109612966,Tpet:0.0000009716);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0247349017,Tmar:0.0087802746):0.0523256720,Tnea:0.1562773333,Tpet:0.0033594668);
(((Tnap:0.0000000001,TRQ2:0.0000000001):0.0100784033,Tmar:0.0093742099):0.0373079810,Tnea:0.1157456527,Tpet:0.0000010042);
```

One file per family

100 trees per file

The Quartet Decomposition Server

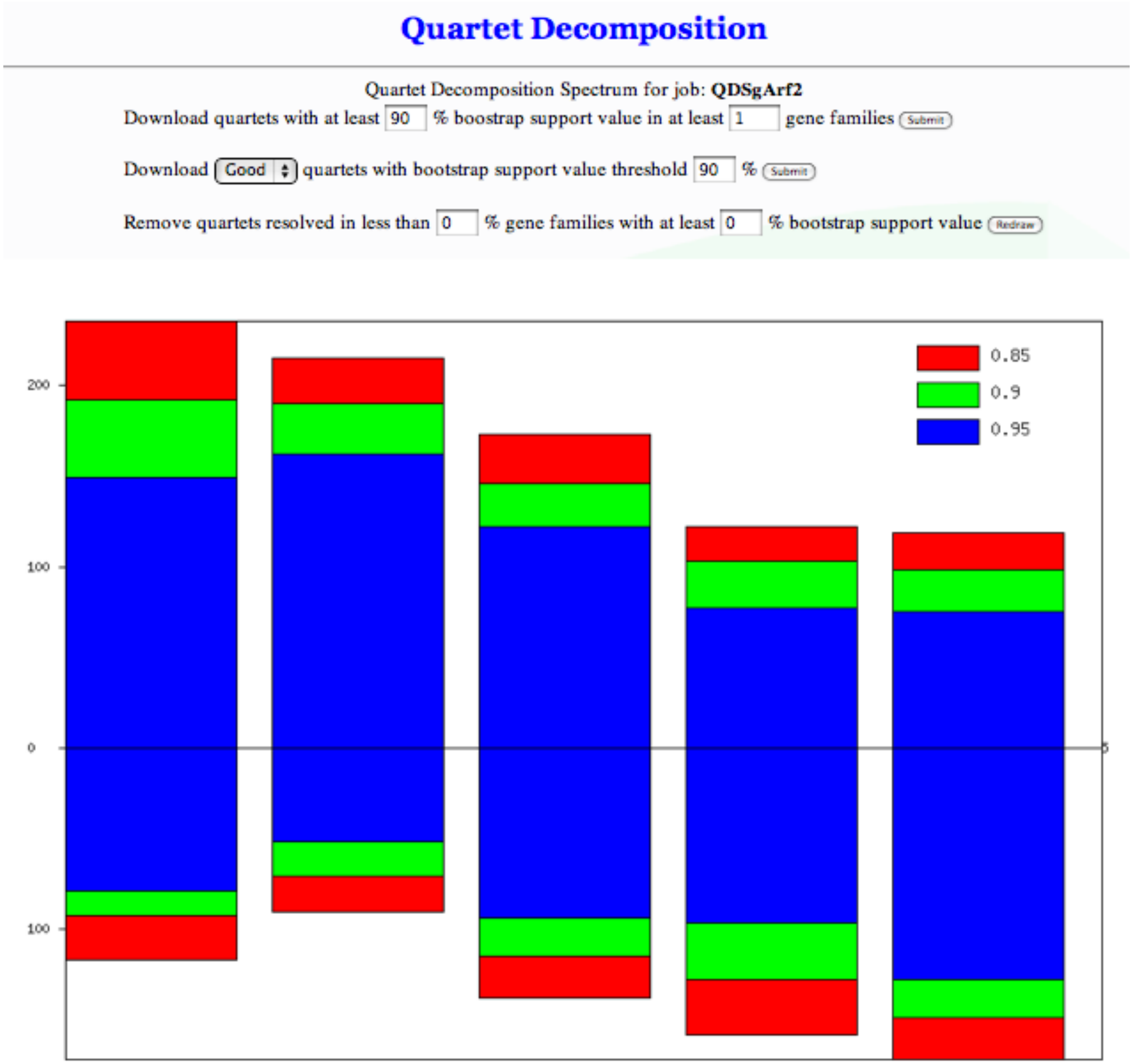
http://csbl1.bmb.uga.edu/QD/phytree.php

Trees from the bootstrap samples should contain branch lengths, but the name for each sequence should be translated to the genome name, using the names in the genome list. See the following three trees in Newick notation for an example:

```
(((Tnea:0.1559823230,Tpet:0.0072068797):
0.0287486818,Tmar:0.0046676053):0.0407339037,Tnap:
0.0000000001,TRQ2:0.0000000001);
(((Tpet:0.0219514318,Tnea:0.1960236242):
0.0145181752,Tmar:0.0189973964):0.0155785587,Tnap:
0.0000000001,TRQ2:0.0000000001);
(((Tpet:0.0000004769,Tnea:0.1773430420):
0.0205769649,Tmar:0.0047117206):0.0416898504,Tnap:
0.0000000001,TRQ2:0.0000000001);
```

The spectrum

http://csbl1.bmb.uga.edu/QD/jobstatus.php?jobid=QDSgArf2&source=0&resolve=0&support=0



Quartet Decomposition

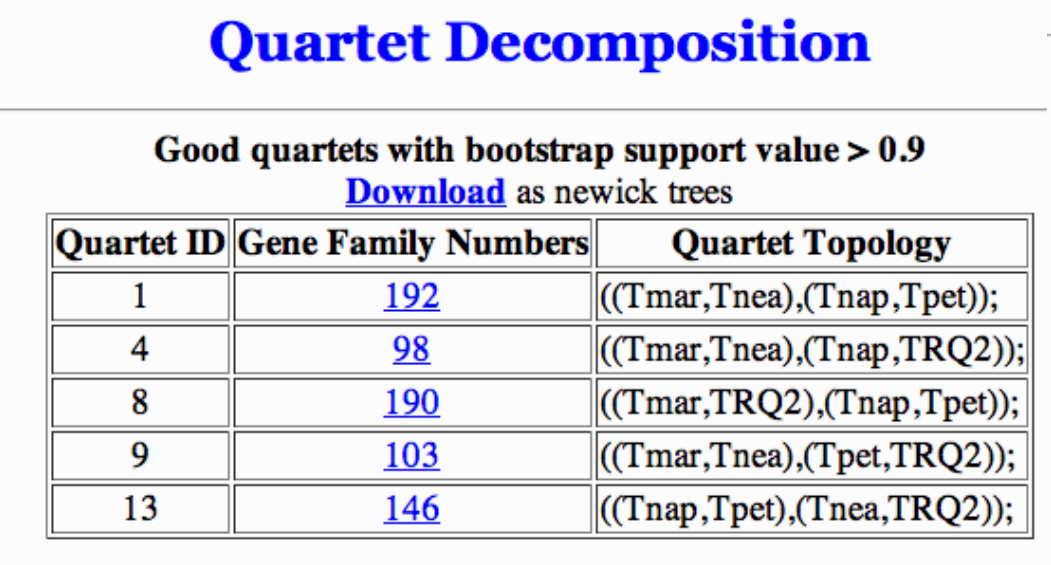
Quartet Decomposition Spectrum for job: **QDSgArf2**

Download quartets with at least 90 % boostrap support value in at least 1 gene families Submit

Download Good quartets with bootstrap support value threshold 90 % Submit

Remove quartets resolved in less than 0 % gene families with at least 0 % bootstrap support value Redraw

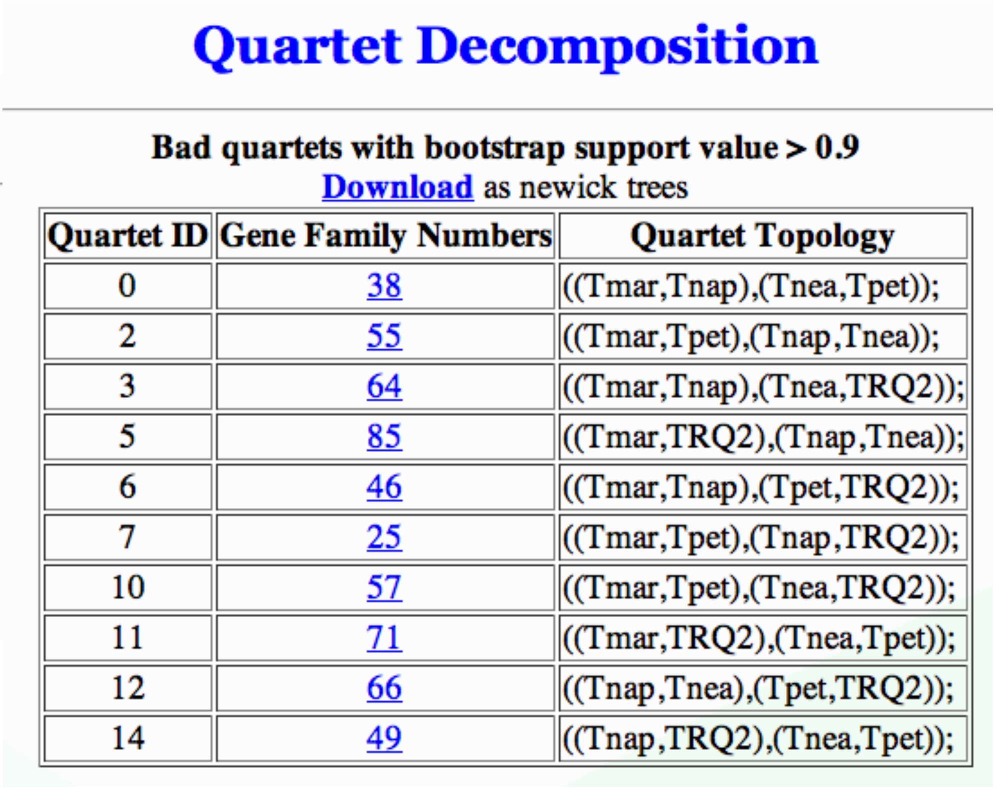
good and bad quartets



Quartet Decomposition

Good quartets with bootstrap support value > 0.9
Download as newick trees

Quartet ID	Gene Family Numbers	Quartet Topology
1	192	((Tmar,Tnea),(Tnap,Tpet));
4	98	((Tmar,Tnea),(Tnap,TRQ2));
8	190	((Tmar,TRQ2),(Tnap,Tpet));
9	103	((Tmar,Tnea),(Tpet,TRQ2));
13	146	((Tnap,Tpet),(Tnea,TRQ2));

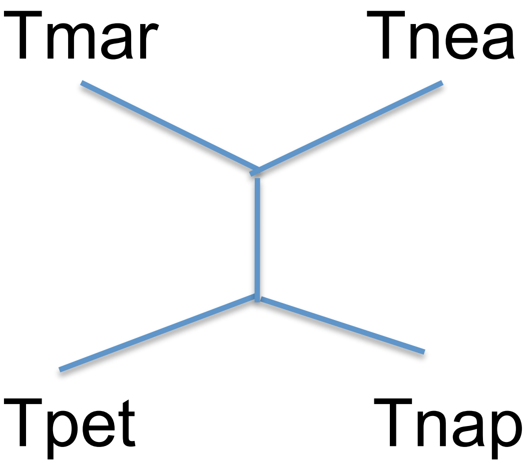


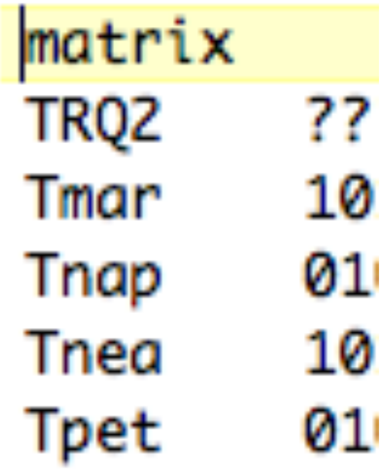
Quartet Decomposition

Bad quartets with bootstrap support value > 0.9
Download as newick trees

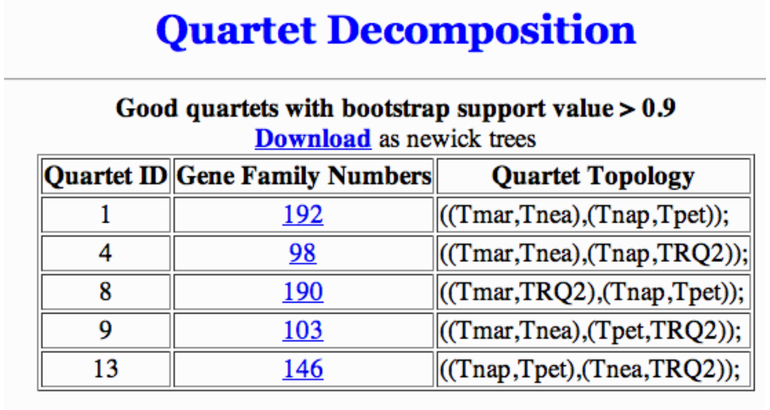
Quartet ID	Gene Family Numbers	Quartet Topology
0	38	((Tmar,Tnap),(Tnea,Tpet));
2	55	((Tmar,Tpet),(Tnap,Tnea));
3	64	((Tmar,Tnap),(Tnea,TRQ2));
5	85	((Tmar,TRQ2),(Tnap,Tnea));
6	46	((Tmar,Tnap),(Tpet,TRQ2));
7	25	((Tmar,Tpet),(Tnap,TRQ2));
10	57	((Tmar,Tpet),(Tnea,TRQ2));
11	71	((Tmar,TRQ2),(Tnea,Tpet));
12	66	((Tnap,Tnea),(Tpet,TRQ2));
14	49	((Tnap,TRQ2),(Tnea,Tpet));

Quartets -> Matrix Representation Using Parsimony





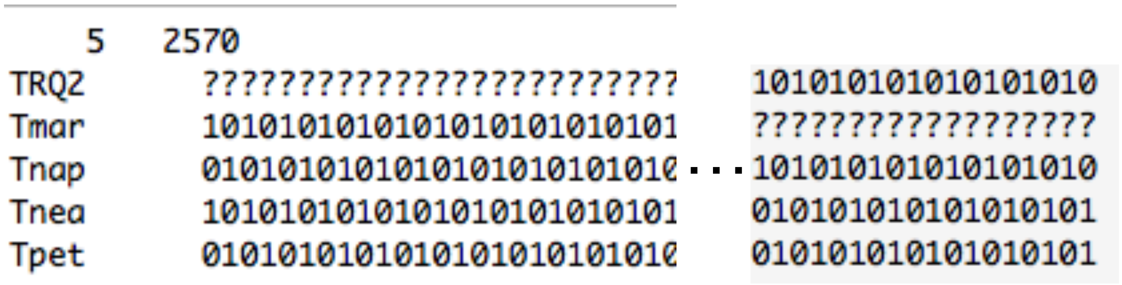
```
matrix
TRQ2      ??
Tmar      10
Tnap      01
Tnea      10
Tpet      01
```



Quartet Decomposition

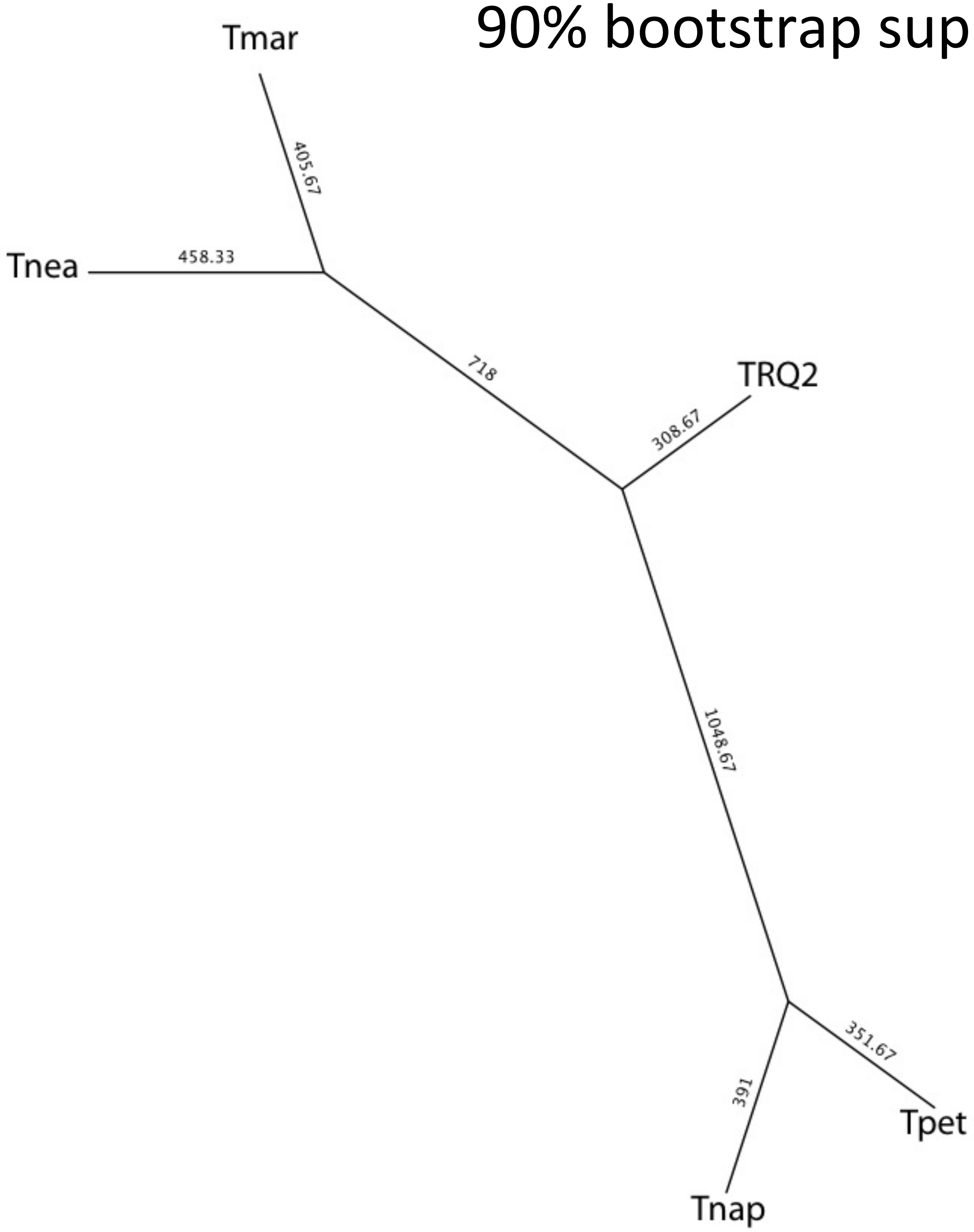
Good quartets with bootstrap support value > 0.9
Download as newick trees

Quartet ID	Gene Family Numbers	Quartet Topology
1	192	((Tmar,Tnea),(Tnap,Tpet));
4	98	((Tmar,Tnea),(Tnap,TRQ2));
8	190	((Tmar,TRQ2),(Tnap,Tpet));
9	103	((Tmar,Tnea),(Tpet,TRQ2));
13	146	((Tnap,Tpet),(Tnea,TRQ2));

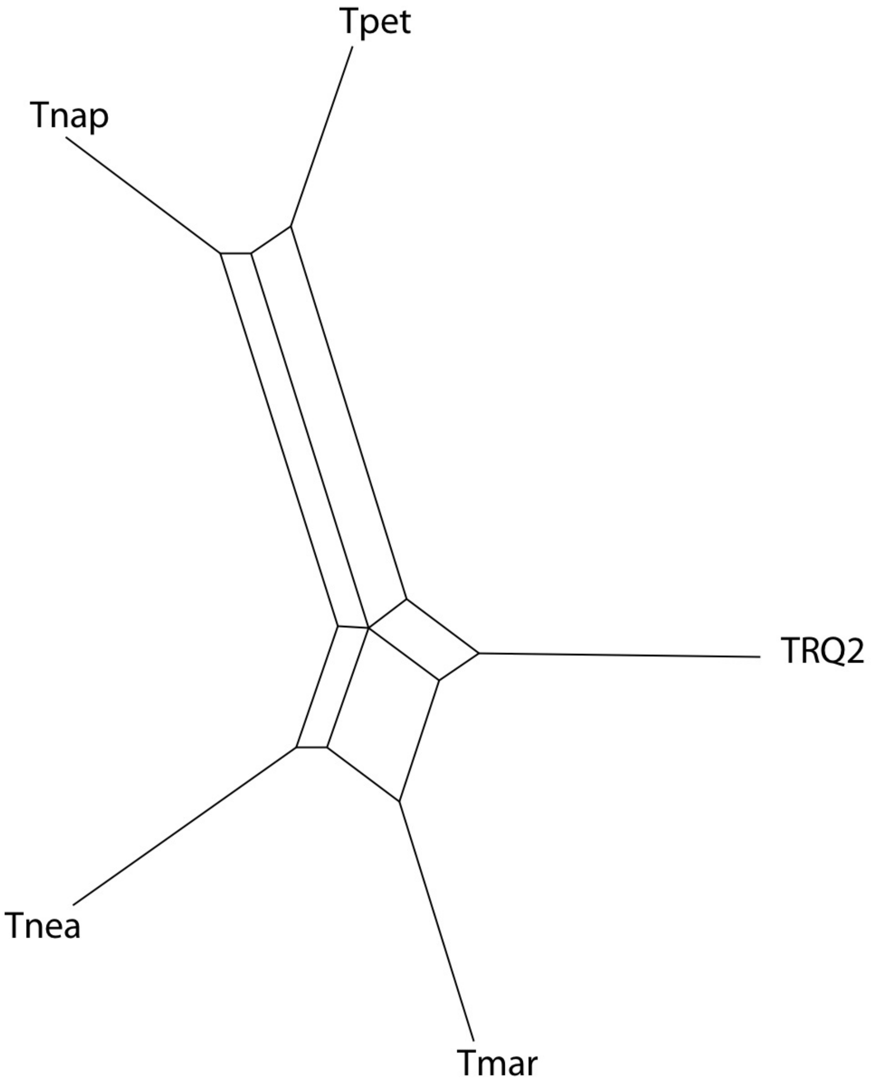


```
    5    2570
TRQ2       ?????????????????????????     101010101010101010
Tmar       1010101010101010101010101     ??????????????????
Tnap       0101010101010101010101010 . . . 101010101010101010
Tnea       1010101010101010101010101     010101010101010101
Tpet       0101010101010101010101010     010101010101010101
```

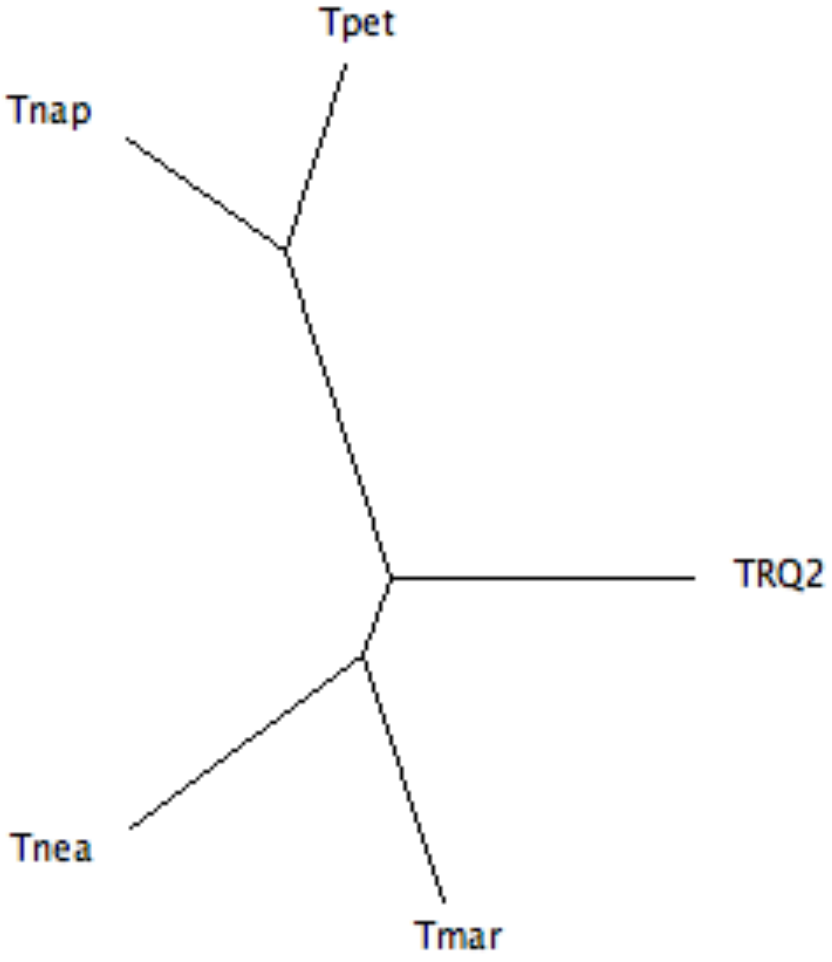
Most Parsimonious Tree (MRP) Using all Quartets from all Gene Families that have more than 90% bootstrap support



Splits Tree Representation
Using all Quartets from all Gene Families that have more than 90% bootstrap support



Split Decomposition tree
from uncorrected P distances



NJ tree
from uncorrected P distances