MCB 5472

Assembly of Gene Families

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

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.

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

For the same dataset to estimate the dN/dS ratios for individual branches, you could use this file codeml.hv1.branches.ctl as control file.

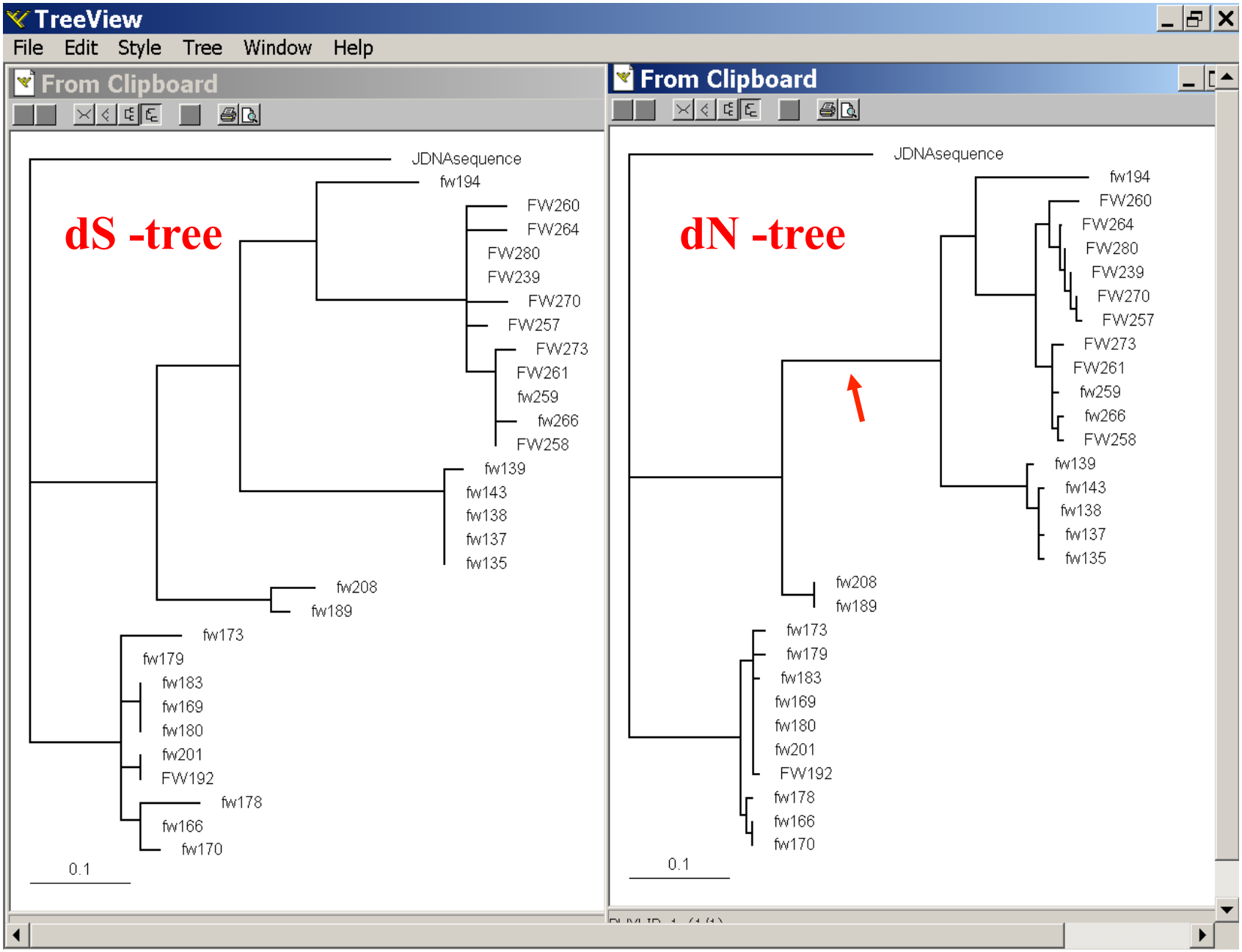
The output is written, as directed by the control file, into a file called Hv1.branch.codeml_out

A good way to check for episodes with plenty of non-synonymous substitutions is to compare the dn and ds trees.

Also, it might be a good idea to repeat the analyses on parts of the sequence (using the same tree). In this case the sequences encode a family of spider toxins that include the mature toxin, a propeptide and a signal sequence (see here for more information).

Bottom line: one needs plenty of sequences to detect positive selection.

PAML – codeml – branch model



where to get help

read the manuals and help files
check out the discussion boards at http://www.rannala.org/phpBB2/

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?

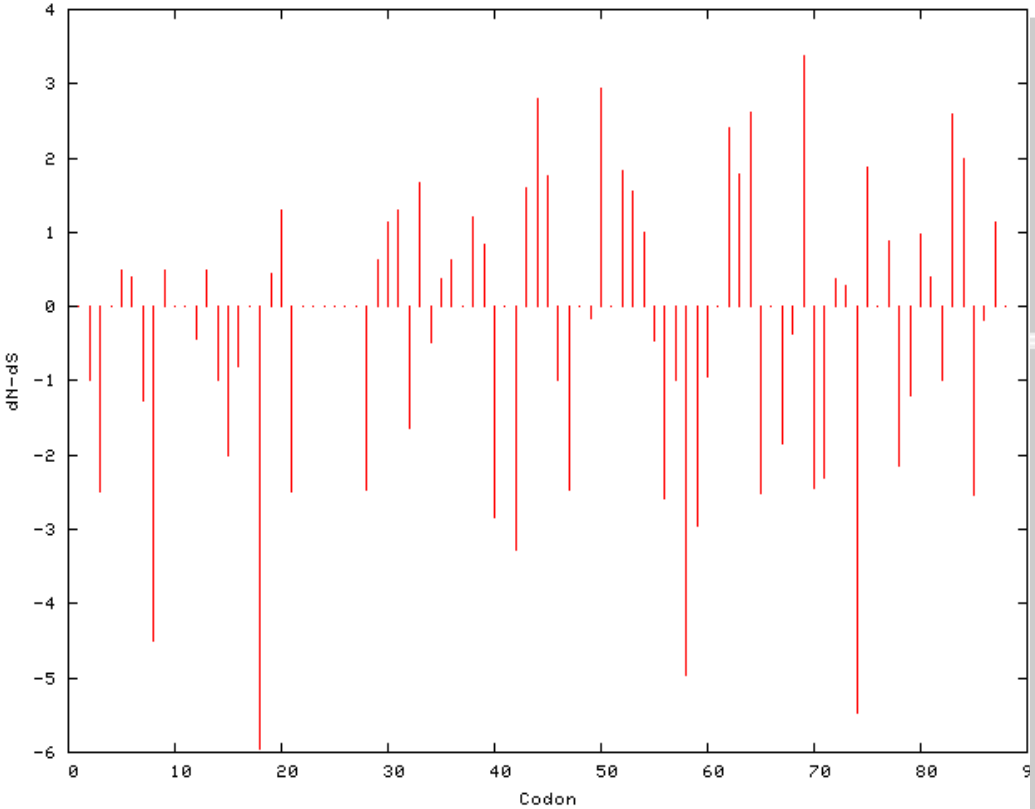
Selective sweep -> fewer alleles present in population

Repeated episodes of positive selection -> high dN

If time discuss http://online.itp.ucsb.edu/online/infobio01/fitch1/

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

more output might still be here

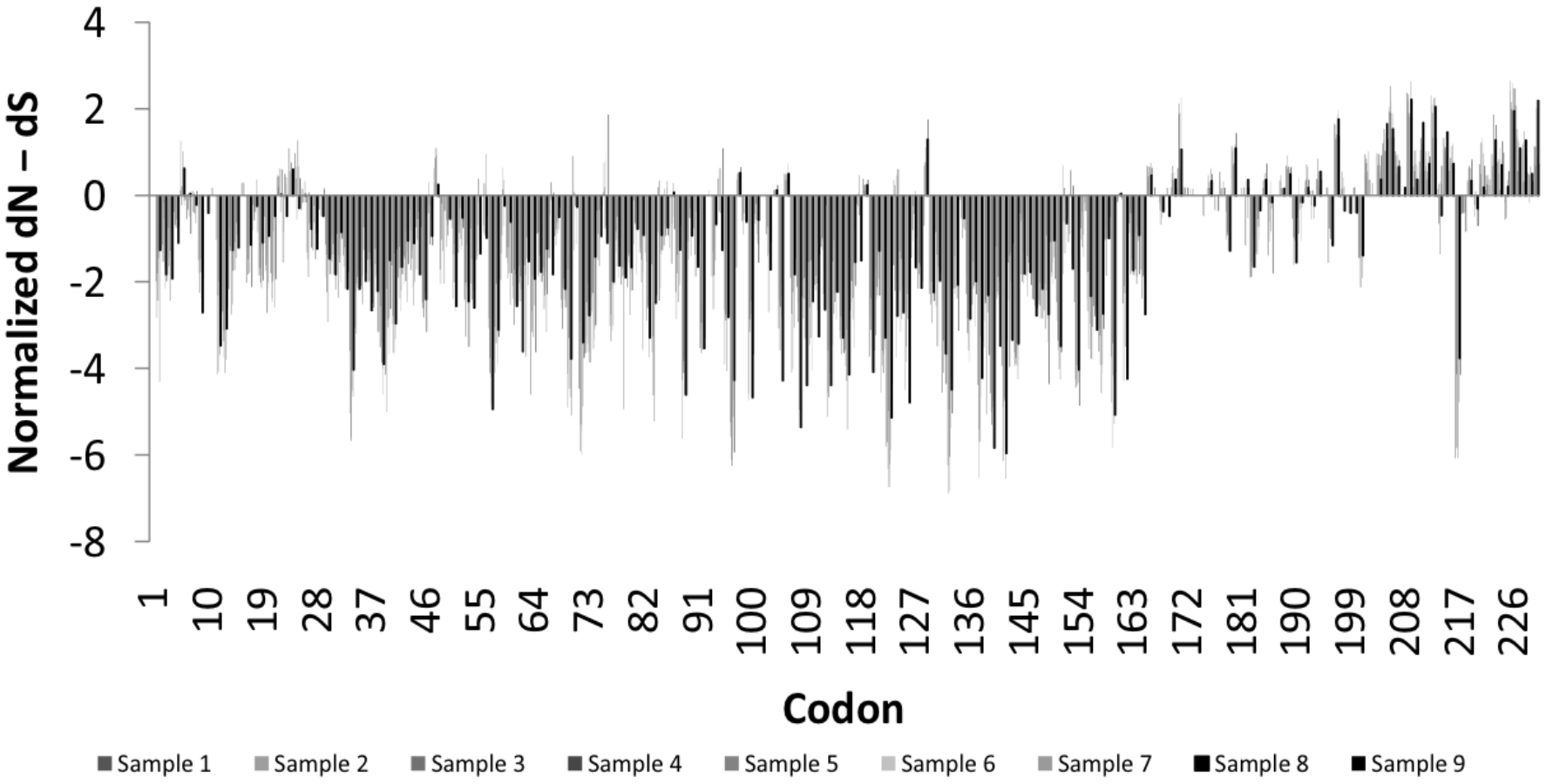


Fig 1. Patterns of substitutions: Bars represent dN > dS (positive) or dN < dS (negative) in random samples of 148 – 150 sequences (A) and the whole dataset of 1312 viruses (B). Included in B are regions of mapped activity and 3D structures of the RNA-binding domain (RBD, panel I) [21] and Effector domain (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.

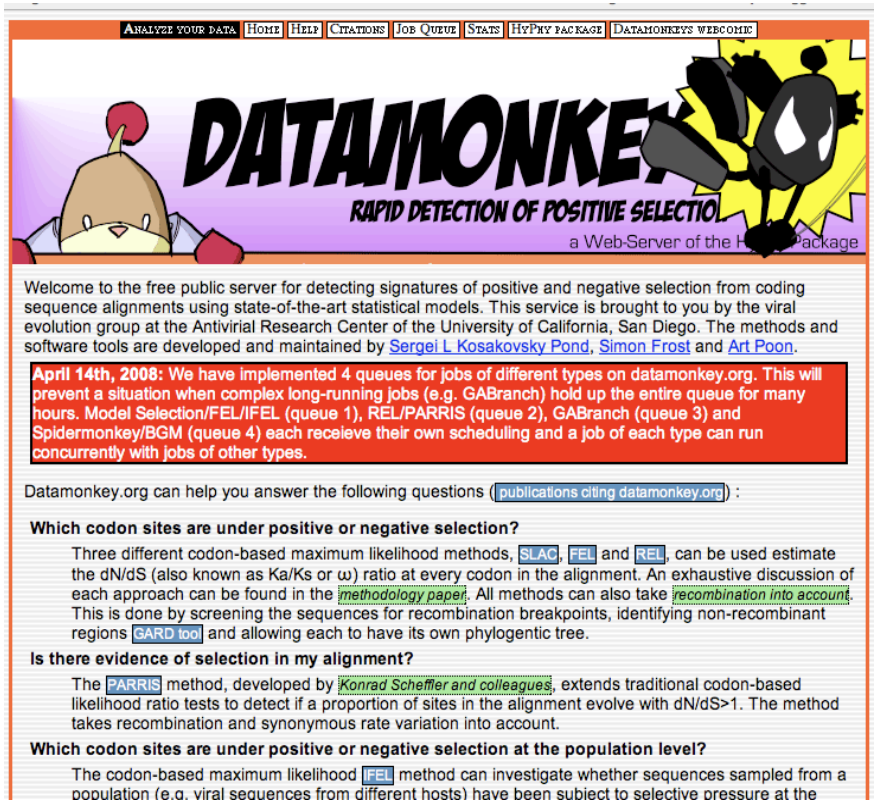
Hy-Phy -

Hypothesis Testing using Phylogenies.

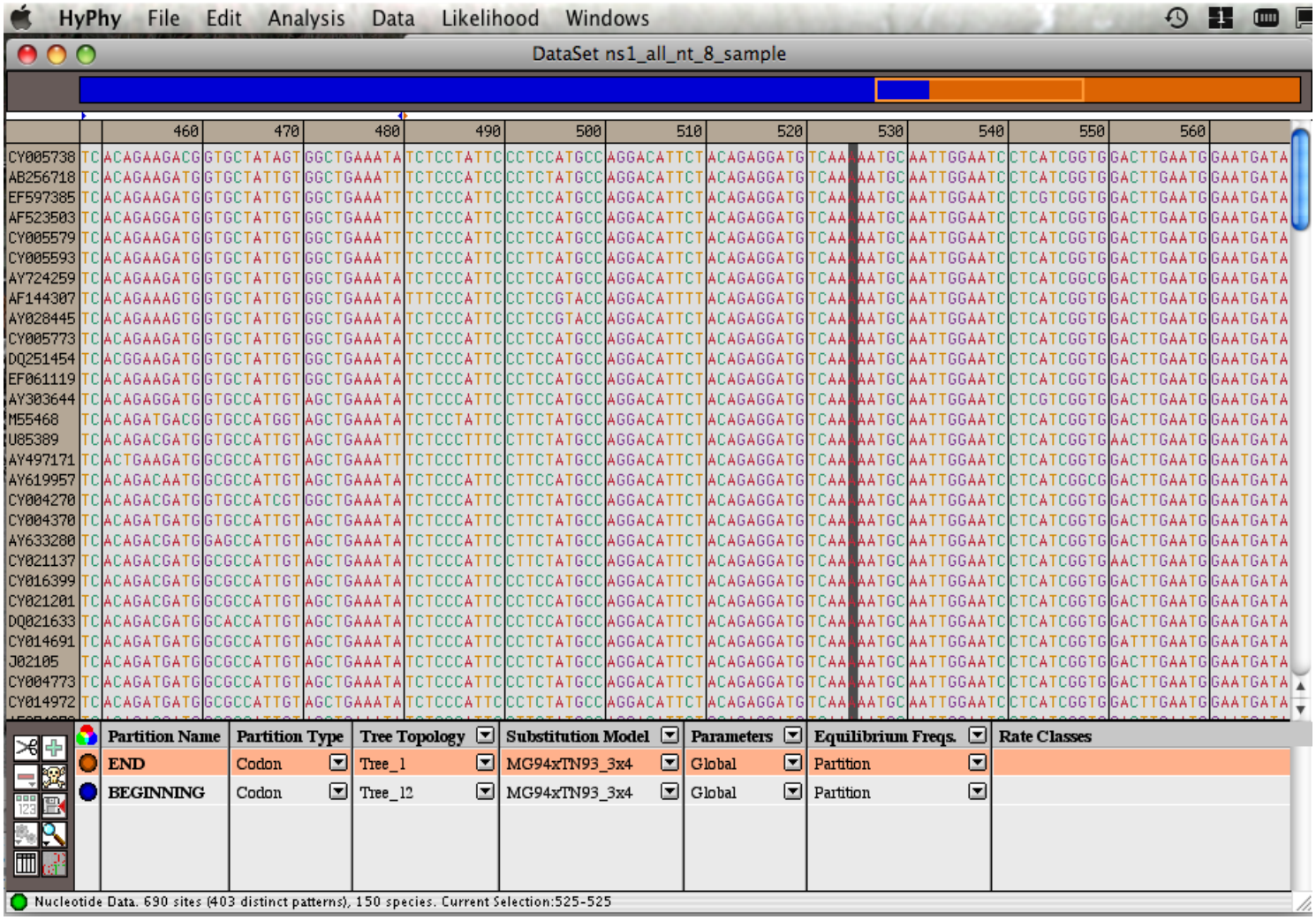
Using Batchfiles or GUI

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

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



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



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_...

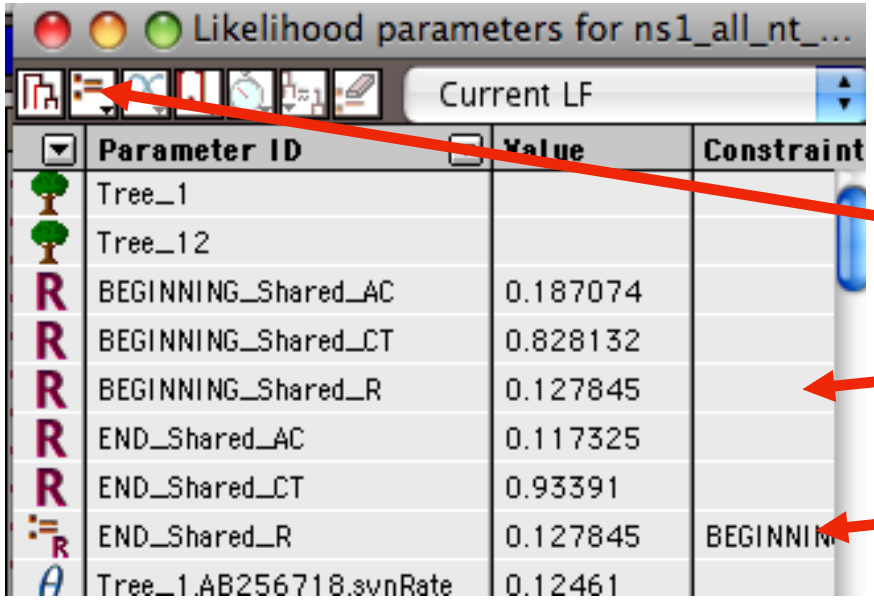
Current LF

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
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	0.946316	
Tree_1.AB256718.synRate	0.12461	
Tree_1.AF001672.synRate	0.016737	
Tree_1.AF009898.synRate	0	
Tree_1.AF055424.synRate	0.017357	
Tree_1.AF074267.synRate	0	
Tree_1.AF074279.synRate	0.0527182	
Tree_1.AF084286.synRate	0.0176037	
Tree_1.AF144307.synRate	0.0528252	
Tree_1.AF256183.synRate	0	
Tree_1.AF256188.synRate	0.0174124	
Tree_1.AF523503.synRate	0.0527042	
Tree_1.AJ344036.synRate	0	
Tree_1.AJ410594.synRate	0.0350104	
Tree_1.AJ410598.synRate	0.0174538	
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

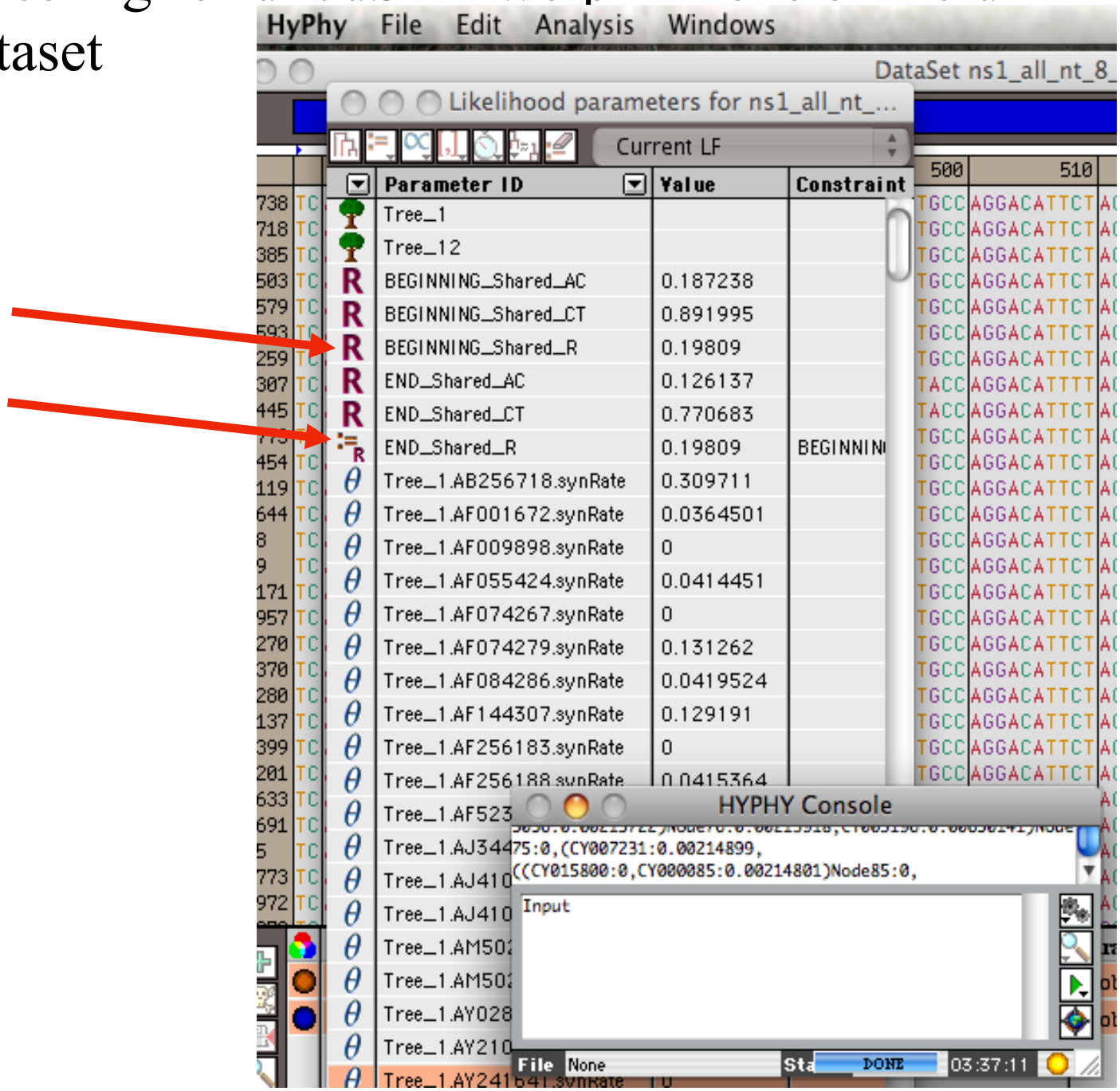


Set up null hypothesis, i.e.:

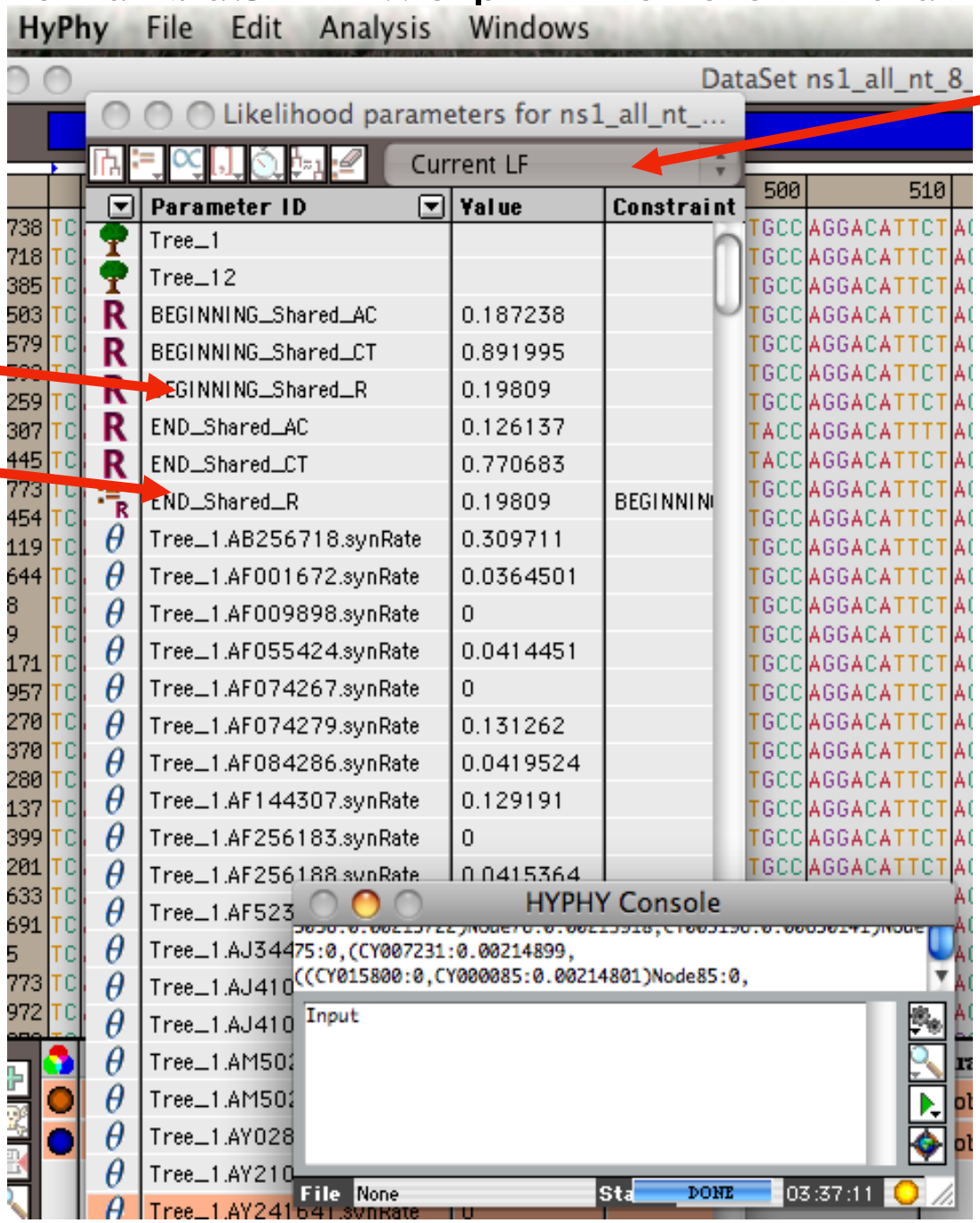
The two dN/dS are equal

(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

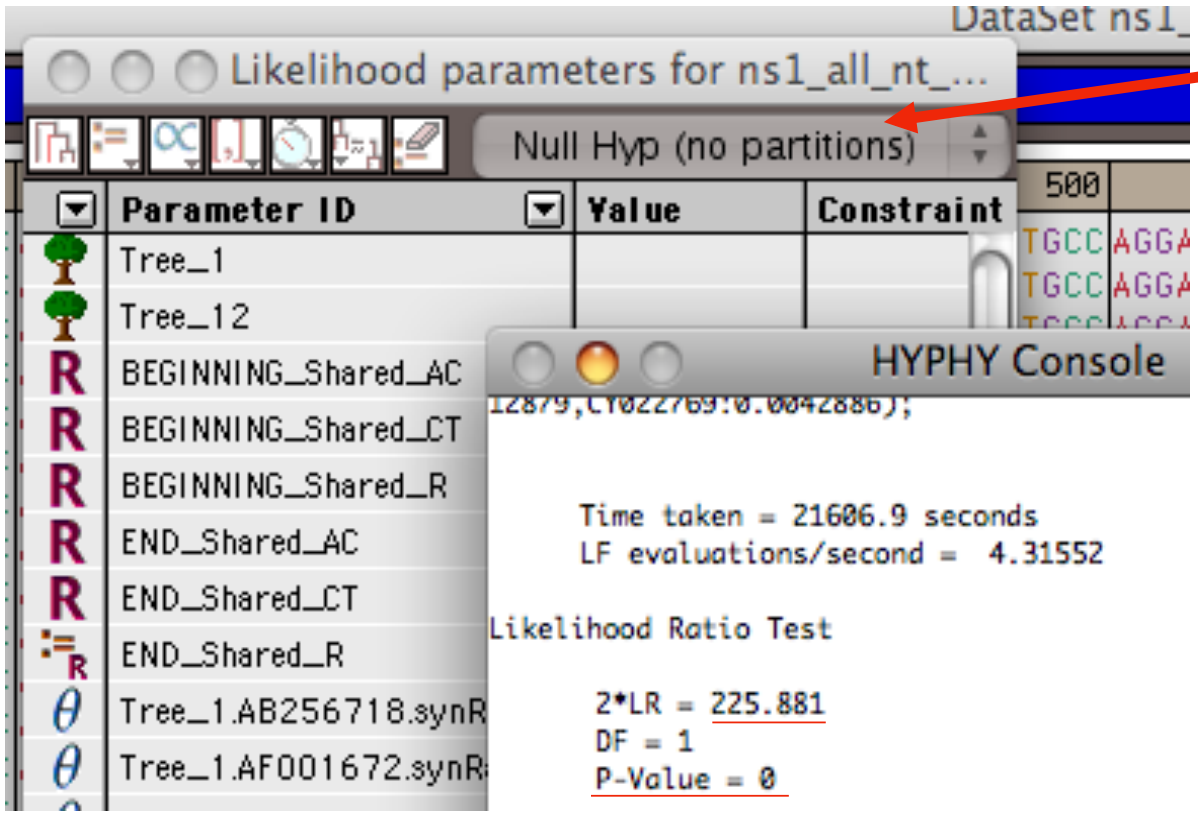


Example testing for dN/dS in two partitions of the data -- John's dataset



Name and save as Null-hyp.

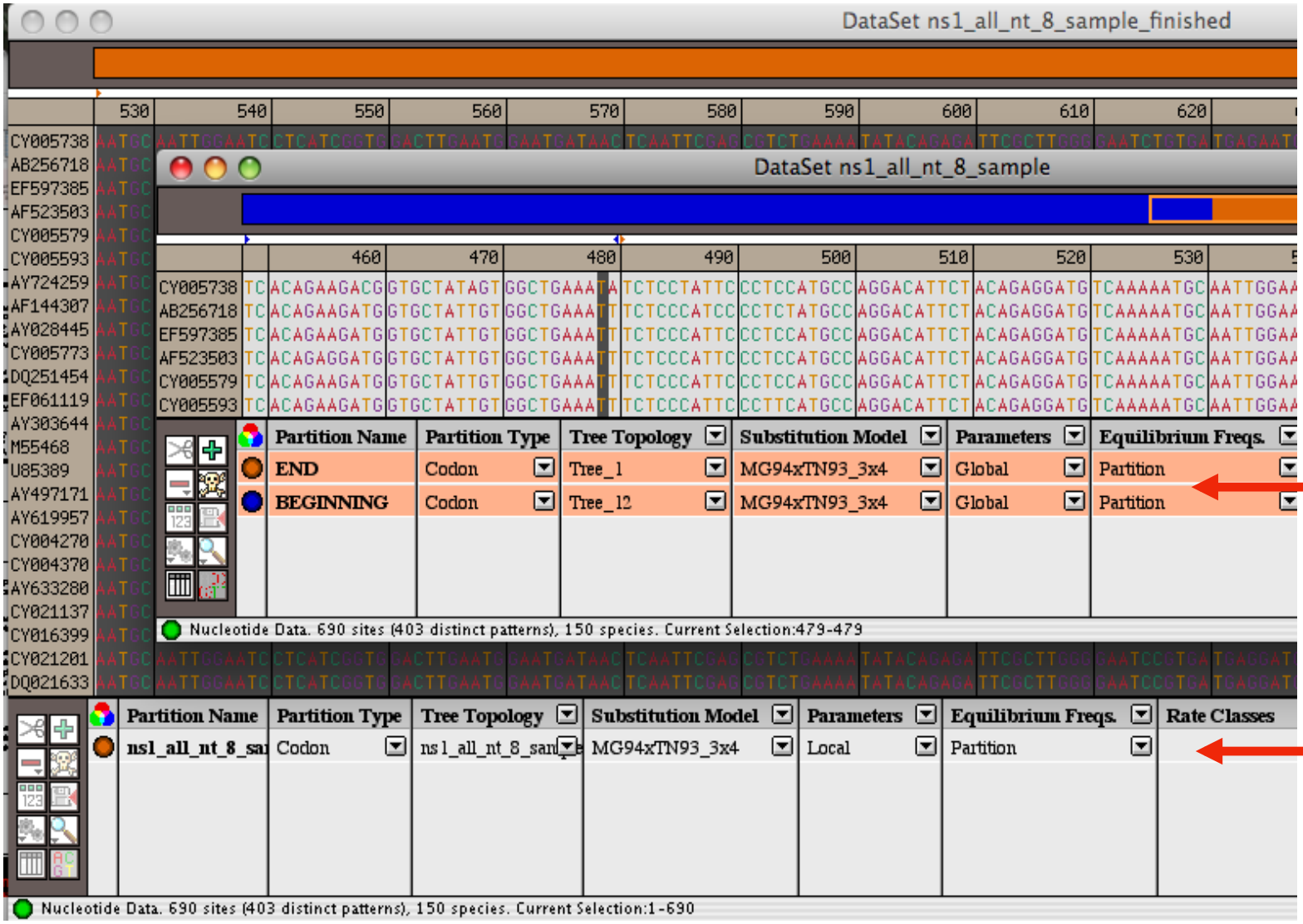
Example testing for dN/dS in two partitions of the data -- John's dataset



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:

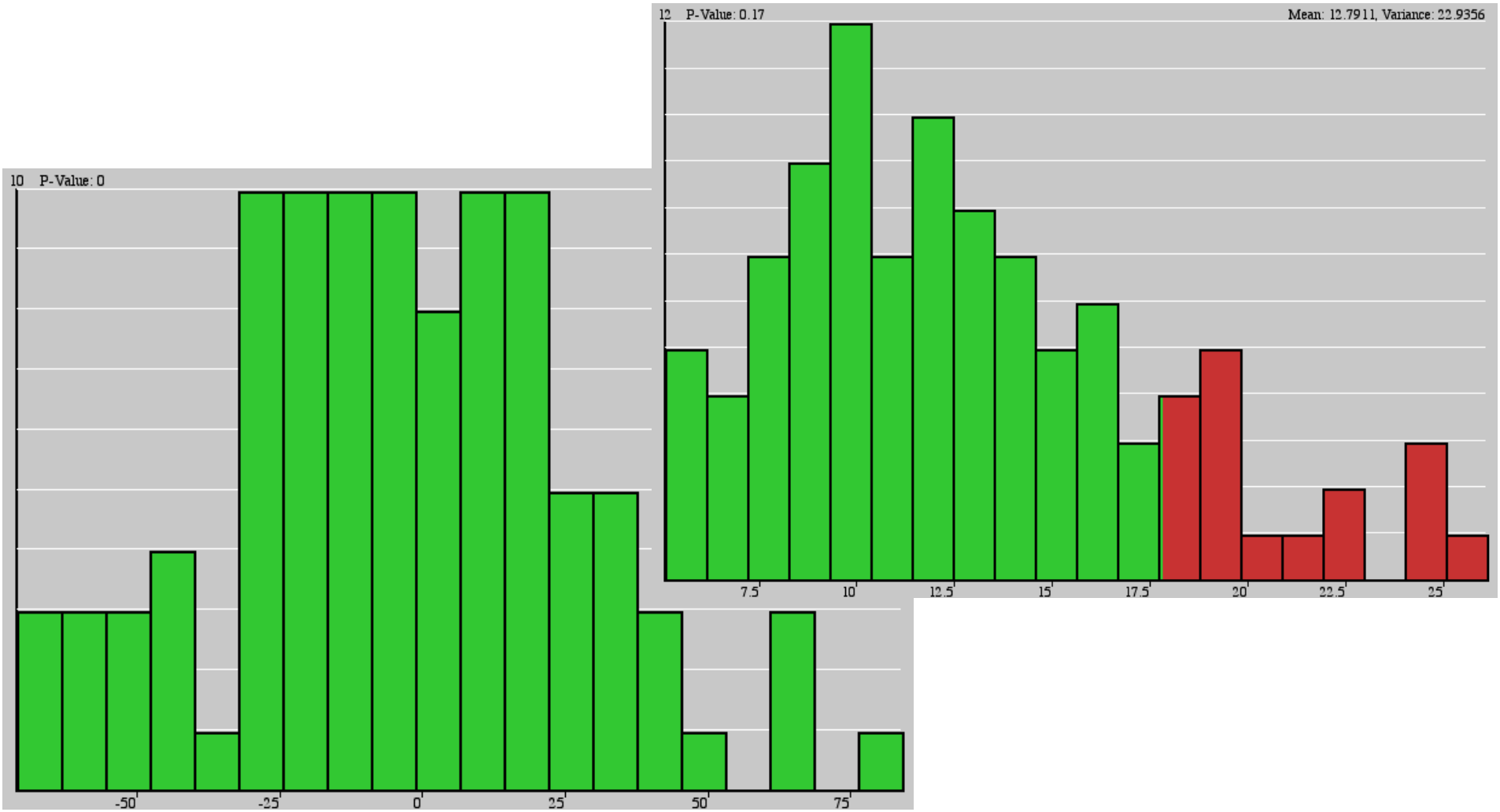


Model 1

Model 2

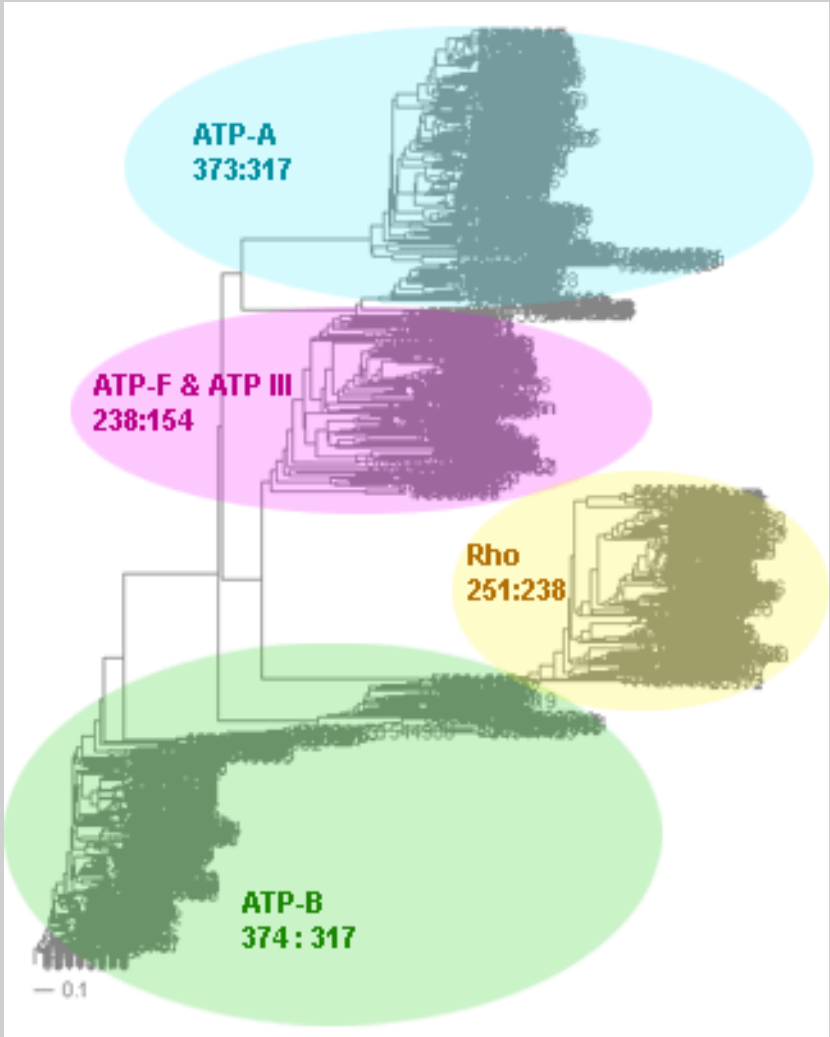
Example testing for dN/dS in two partitions of the data -- John's dataset

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



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

Workshop at Te Aviv University, November 29th, 2009.

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

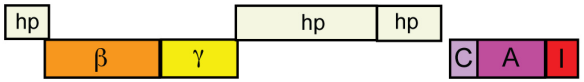
Number of families with 1 missing gene

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***

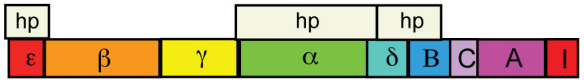
Example of missed ORFs

ATP synthase operon

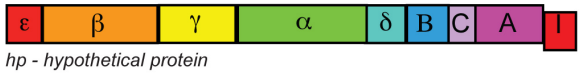
4 missing genes in Escherichia coli APEC O1



Escherichia coli UTI89



Escherichia coli CFT073



hp - hypothetical protein

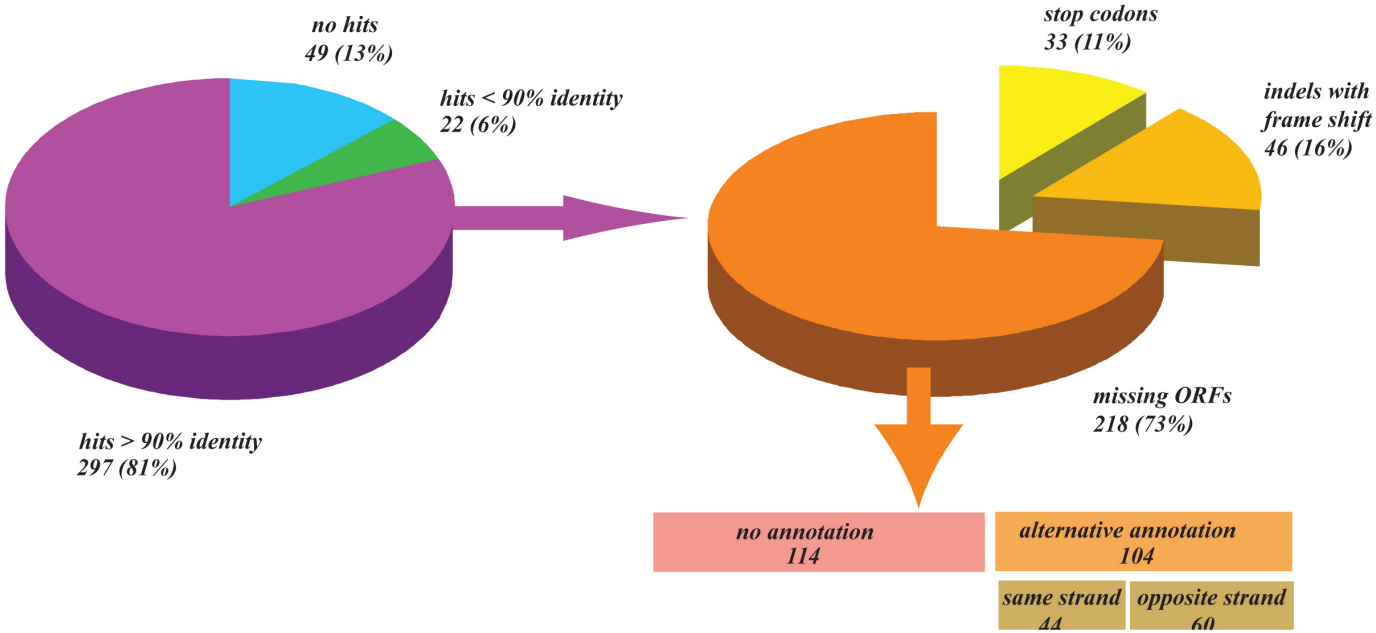
ε, β, γ, α, δ, B, C, A, I - ATP synthase subunits

Analysis of 368 missing orthologs with blastn

An ortholog from a family with 1 missing gene was used as a query against nucletide 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).

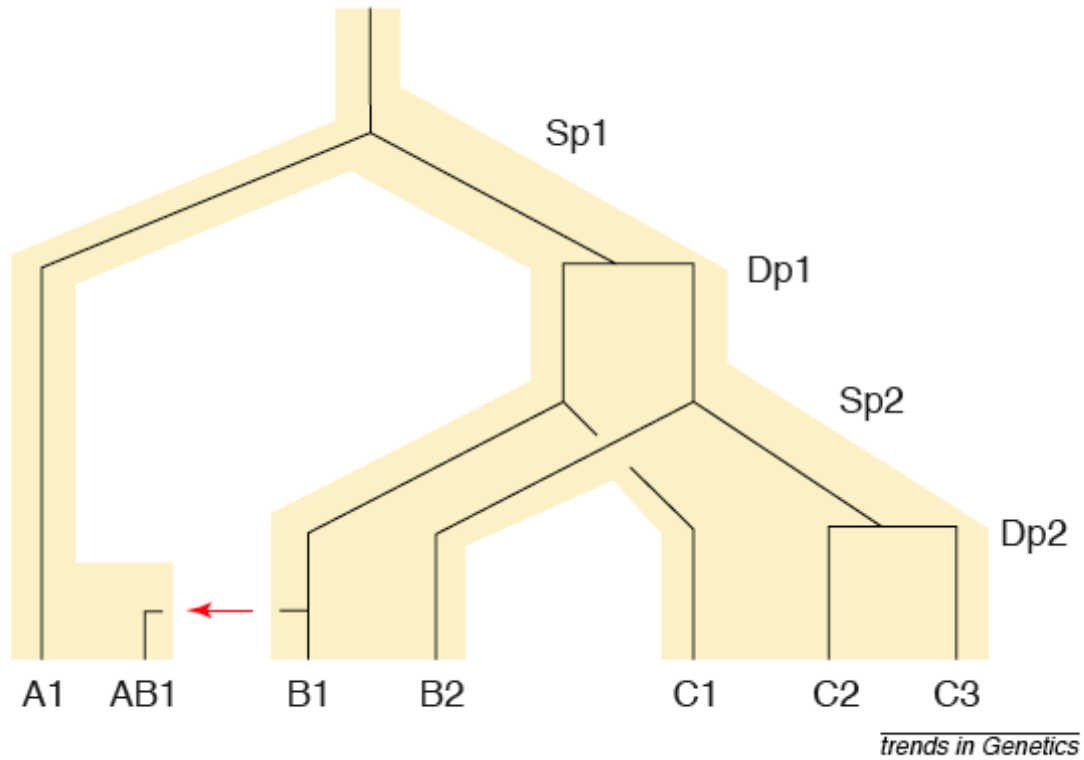


Homologs, orthologs, and paralogs

- Homologous structures or characters evolved from the same ancestral structure or character that *existed in some organism in the past*.
- Orthologous characters present in two organism (A and B) are homologs that are derived from a structure *that existed in the most recent common ancestor* (MRCA) of A and B (orthologs often have the same function, but this is NOT part of the definition; e.g. human arms, wings or birds and bats).
- Paralogous characters in the same or in two different organisms are homologs that are not derived from the same character in the MRCA, rather they are *related* (at their deepest node) *by a gene duplication event*.

Examples

FIGURE 1. Orthology, paralogy and xenology

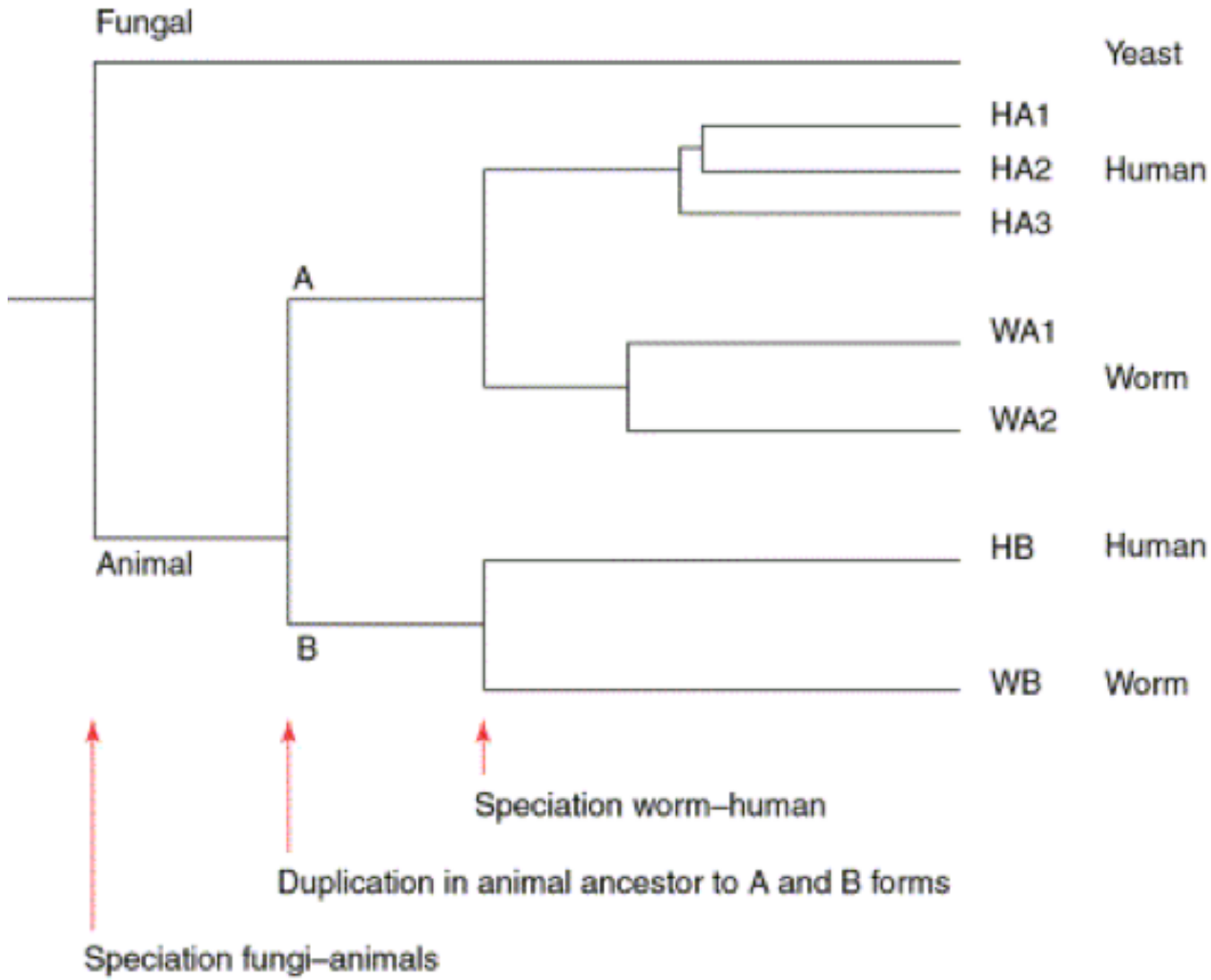


B1 is an ortholog to C1 and to A1
C2 is a paralog to C3 and to B1;
BUT
A1 is an ortholog to both B1, B2,and to C1, C2, and C3

From: Walter Fitch (2000): *Homology: a personal view on some of the problems*, TIG 16 (5) 227-231

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

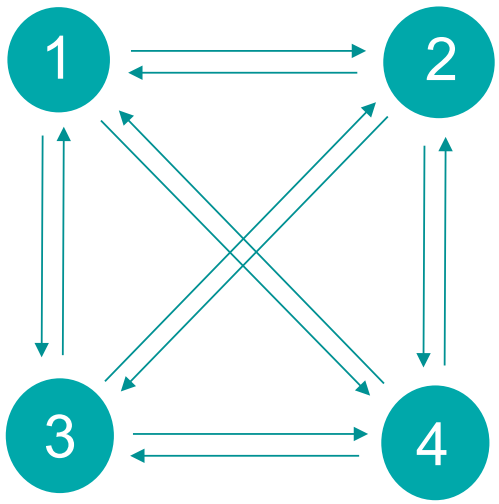
(SCOP database)

Similarity complemented by HMM-profile analysis

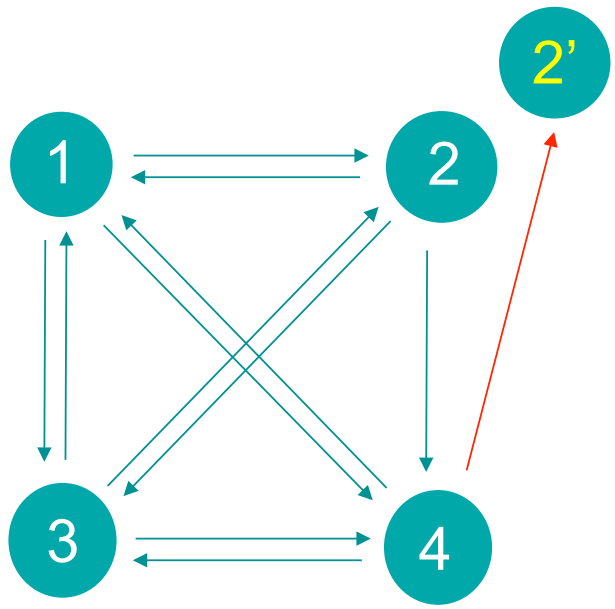
Pfam database

Reciprocal BLAST hit method

Strict Reciprocal BLAST Hit Method



1 gene family



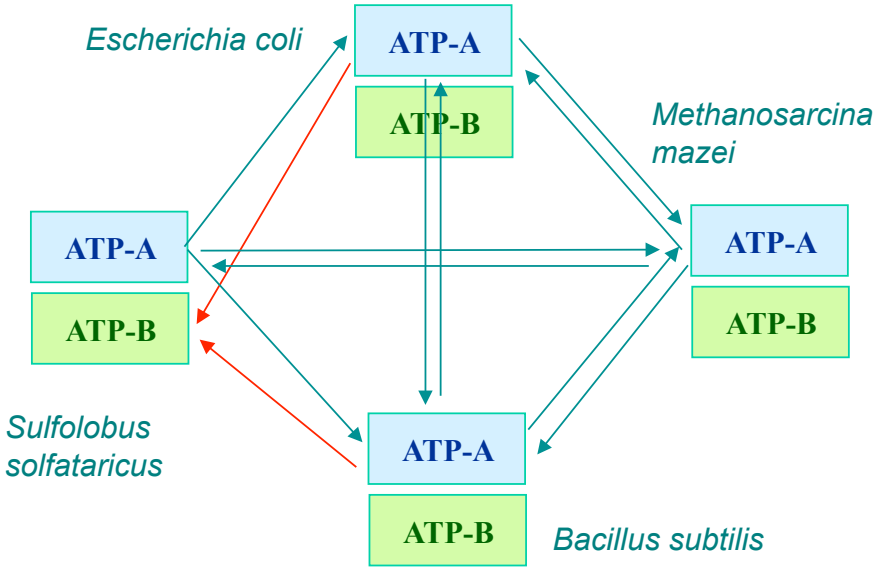
0 gene family

often fails in the presence of paralogs

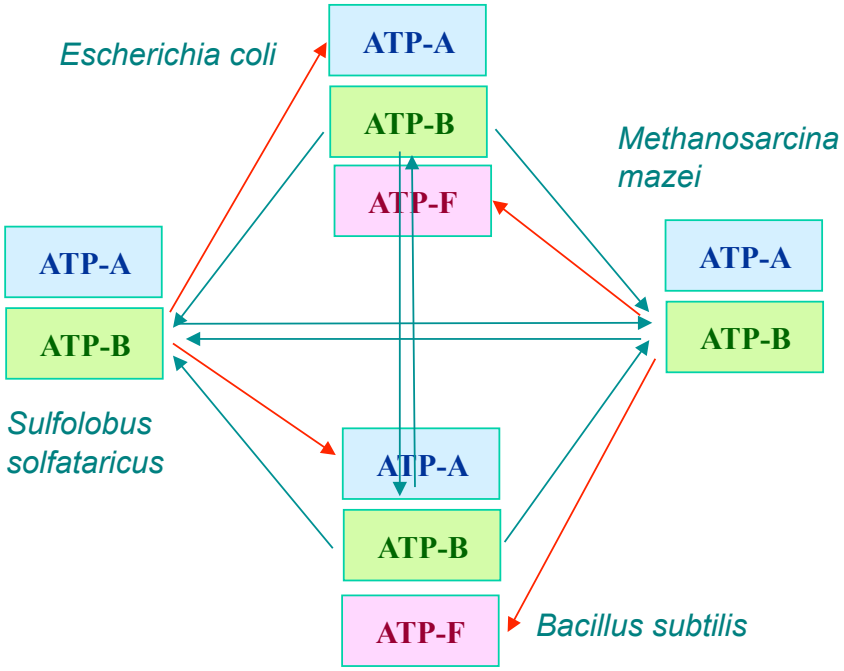
Families of ATP-synthases

Case of 2 bacteria and 2 archaea species

ATP-A (catalytic subunit)



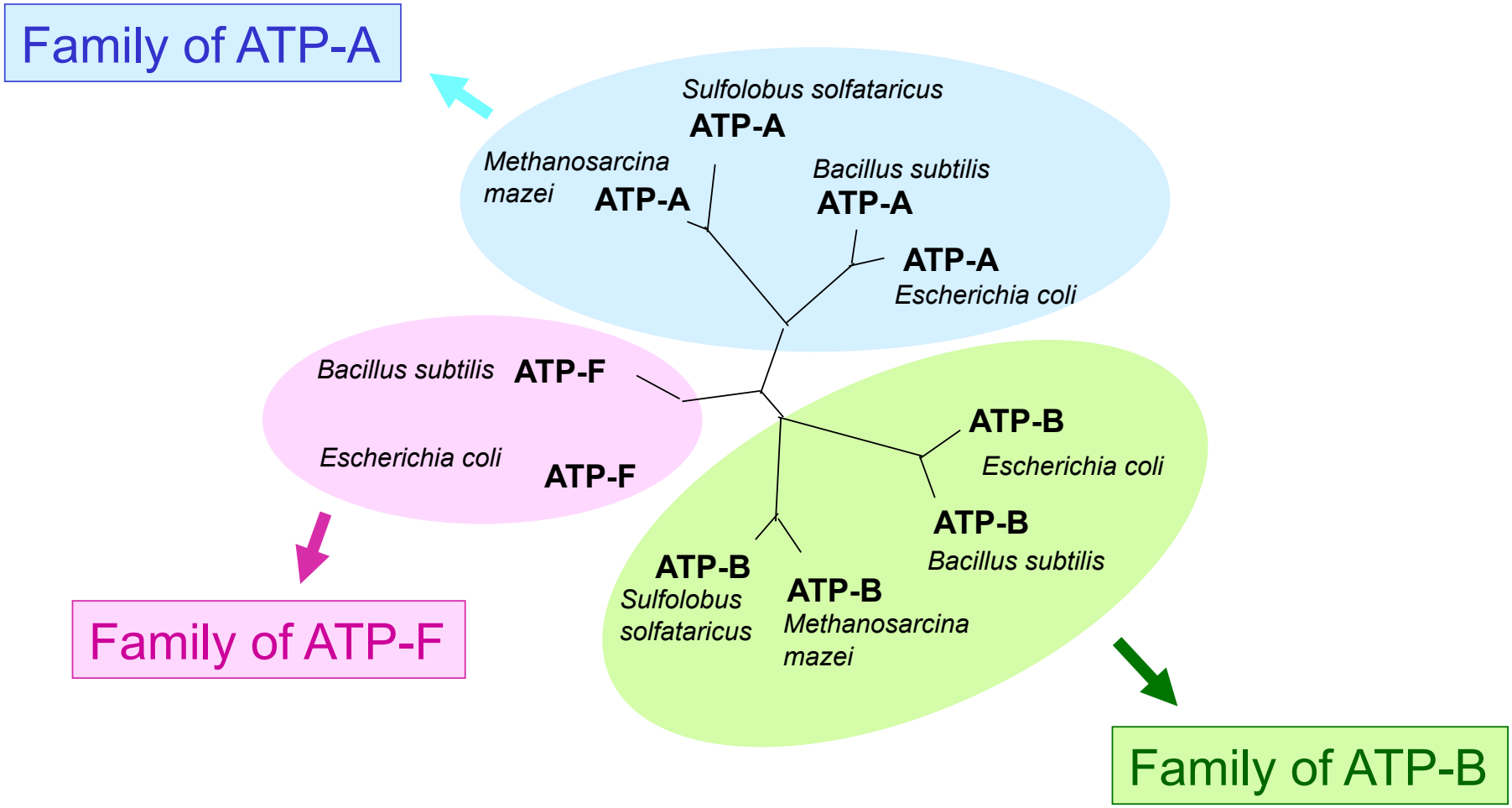
ATP-B (non-catalytic subunit)



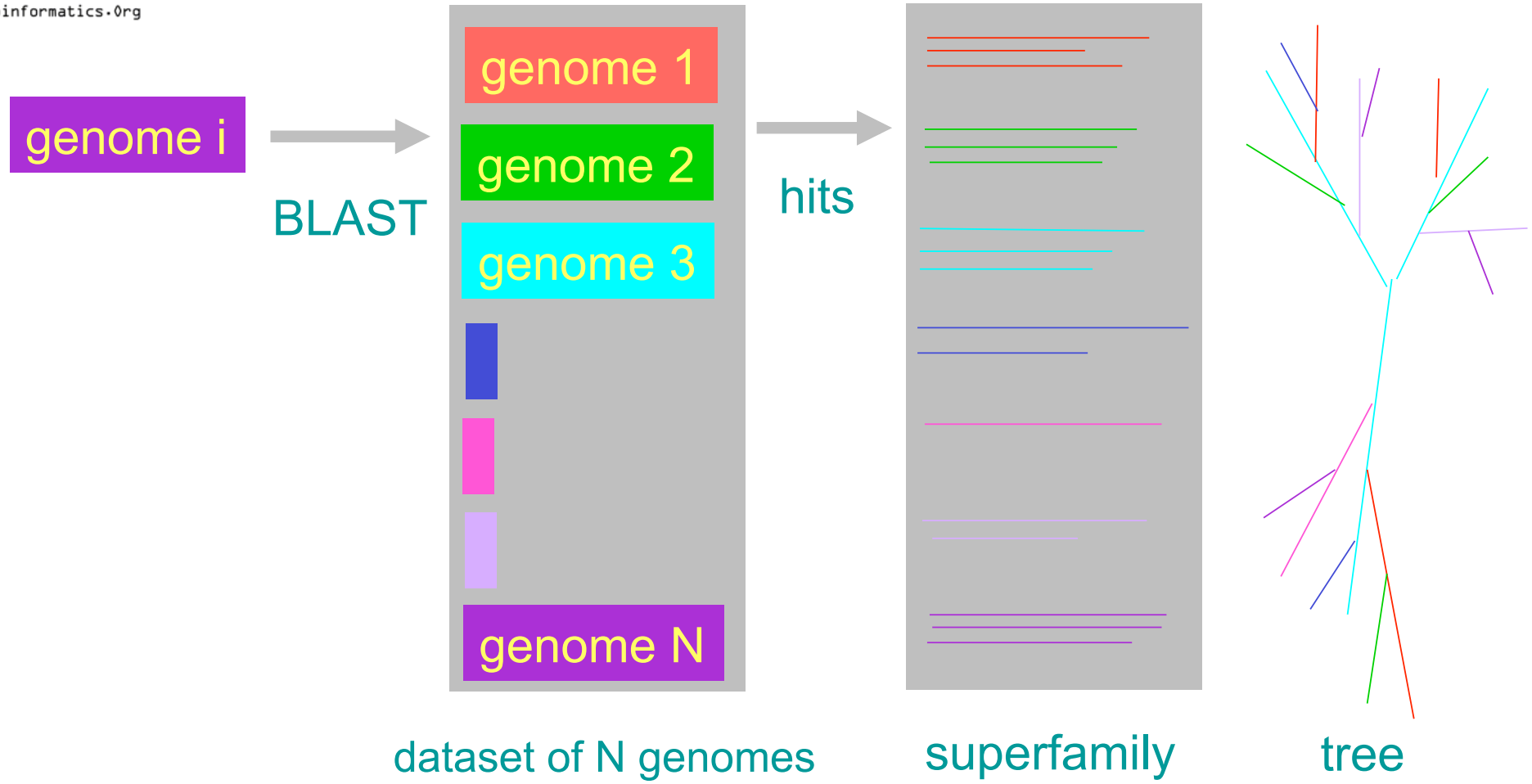
Neither ATP-A nor ATB-B is selected by RBH method

Families of ATP-synthases

Phylogenetic Tree

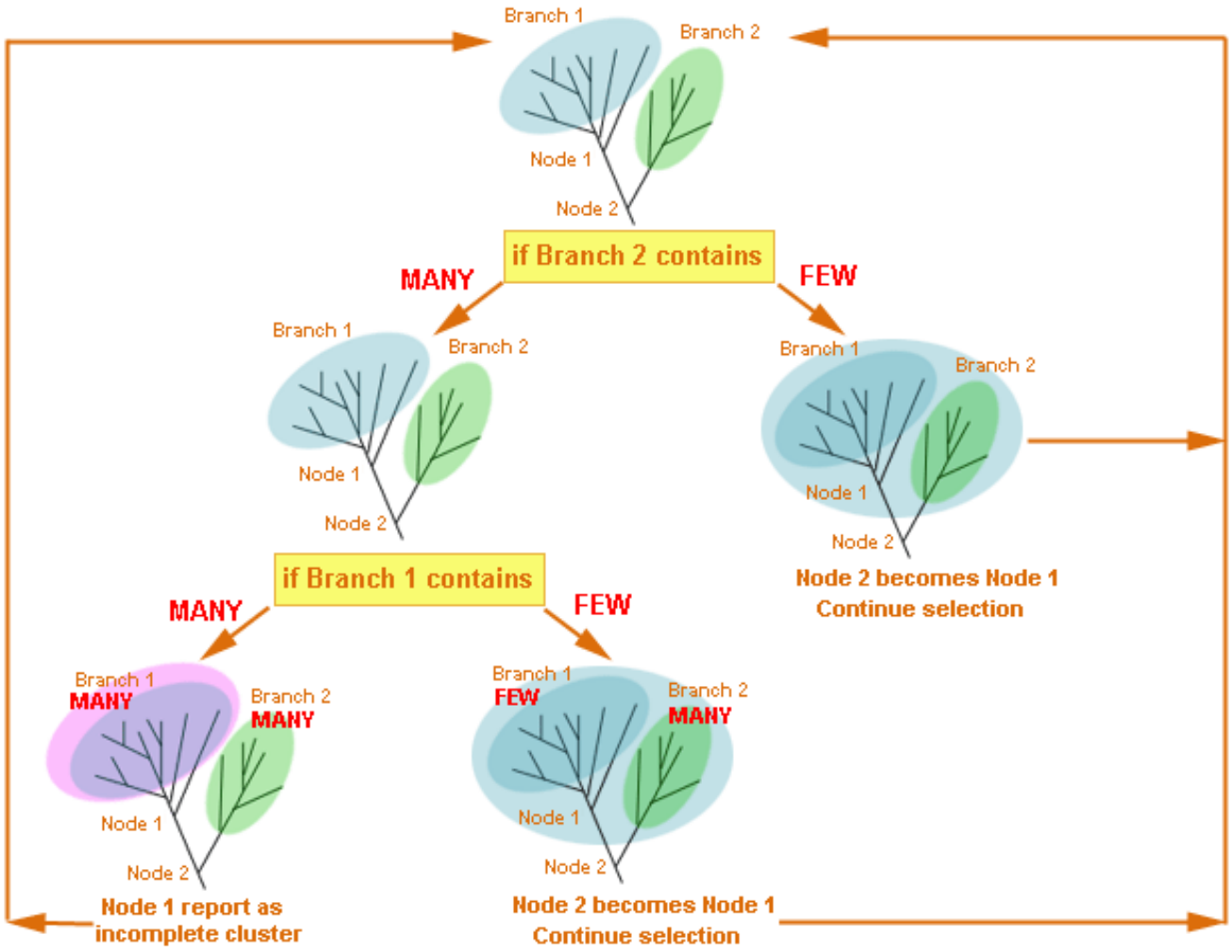


BranchClust Algorithm



BranchClust Algorithm

Bioinformatics.Org

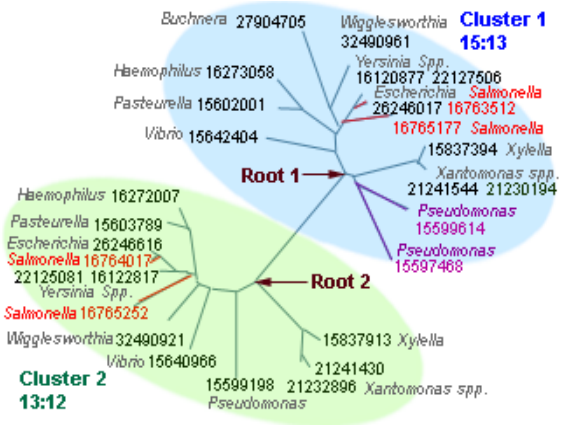


BranchClust Algorithm

Root positions

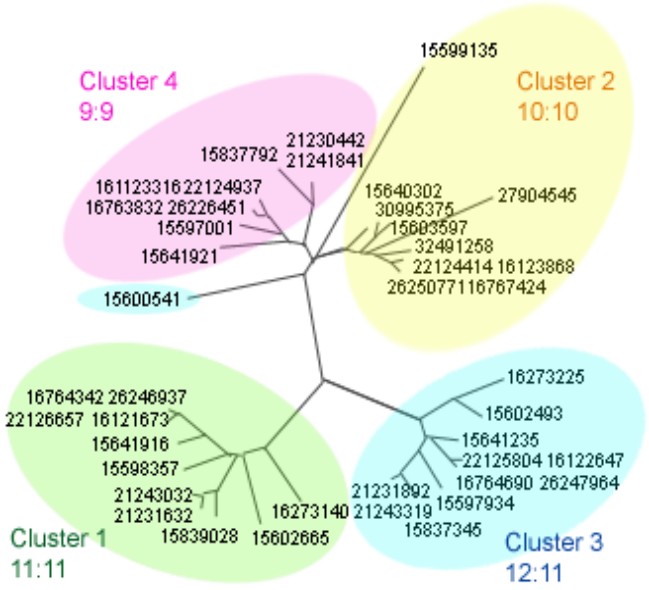
Superfamily of penicillin-binding protein

13 gamma proteo bacteria



Superfamily of DNA-binding protein

13 gamma proteo bacteria



Bioinformatics.Org

BranchClust Algorithm

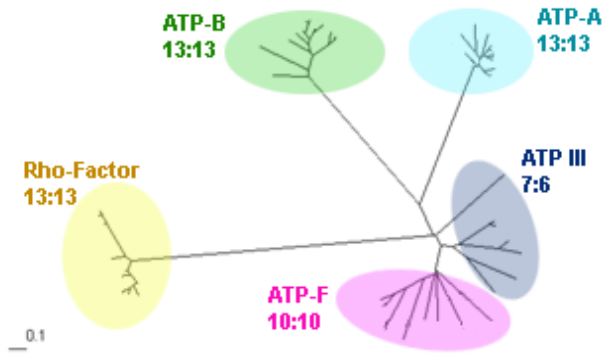
Comparison of the best BLAST hit method and BranchClust algorithm

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).

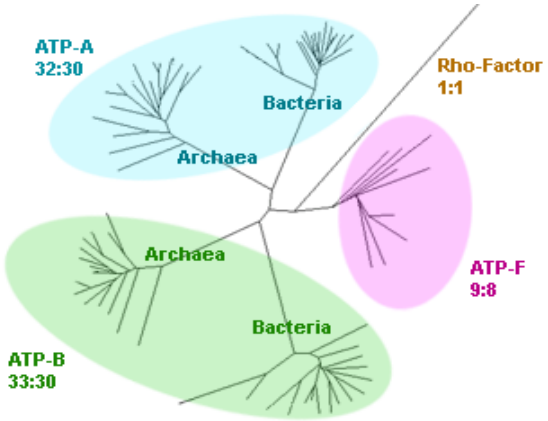
BranchClust Algorithm

ATP-synthases: Examples of Clustering

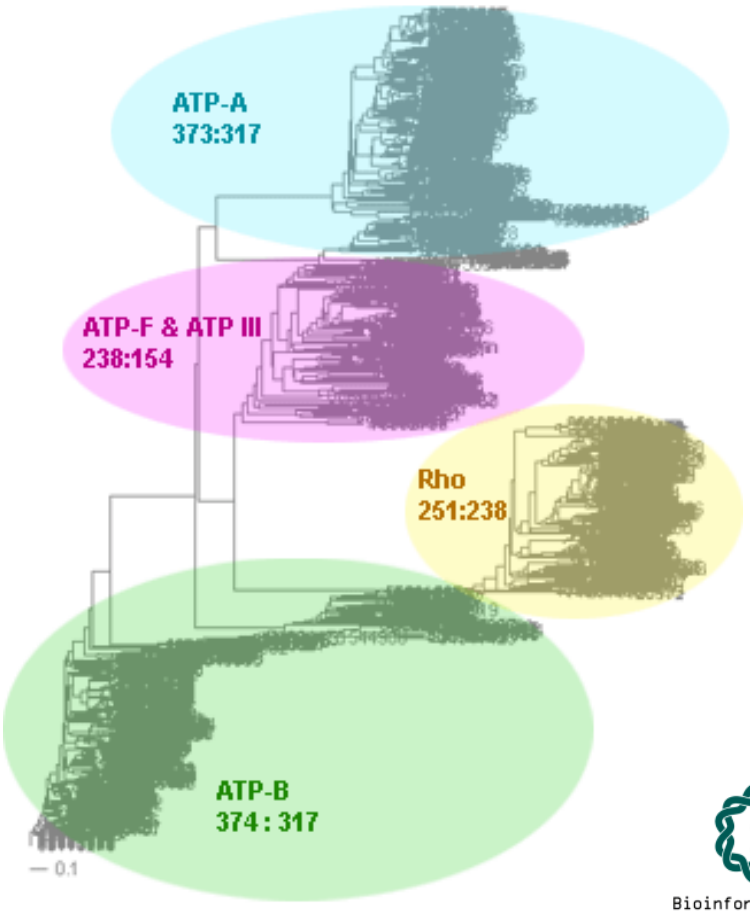
13 gamma proteobacteria



30 taxa: 16 bacteria and 14 archaea



317 bacteria and archaea

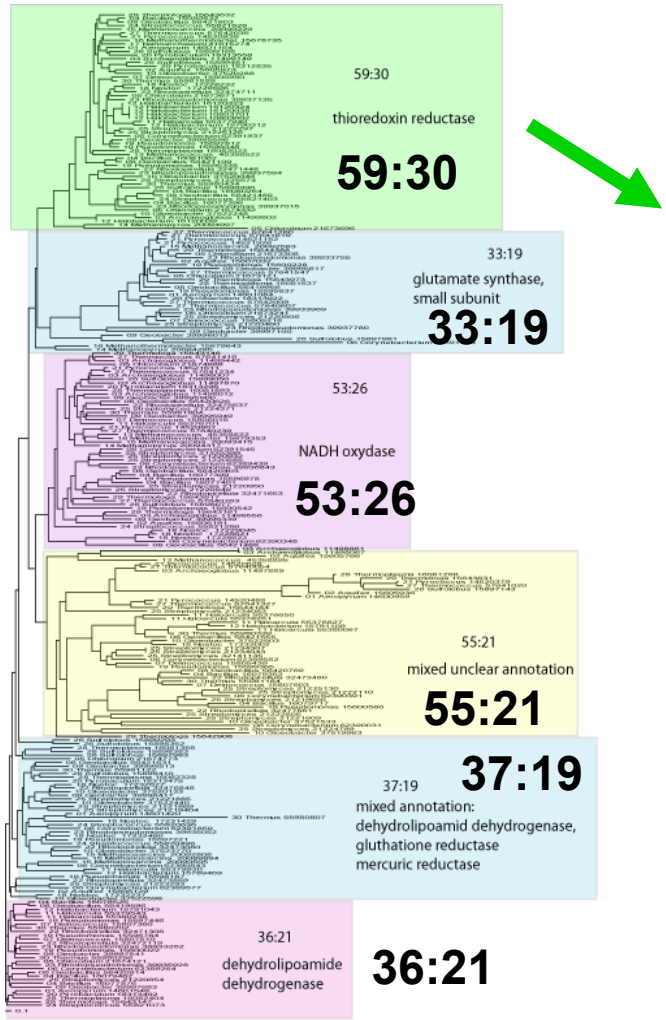


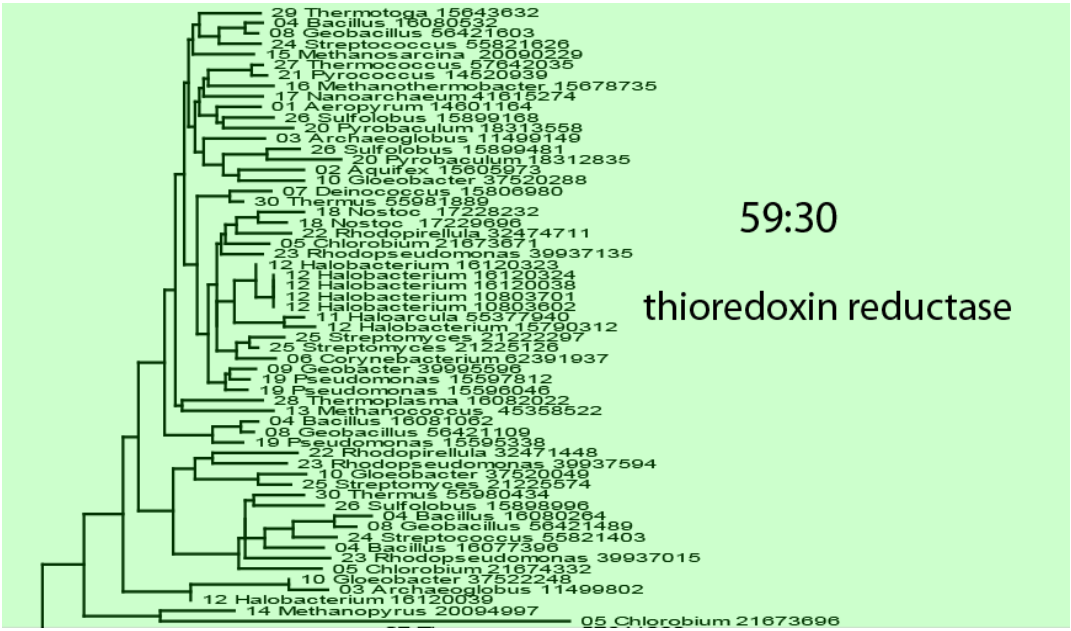
Bioinformatics.Org

www.bioinformatics.org/branchclust

BranchClust Algorithm

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





Bioinformatics.Org

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

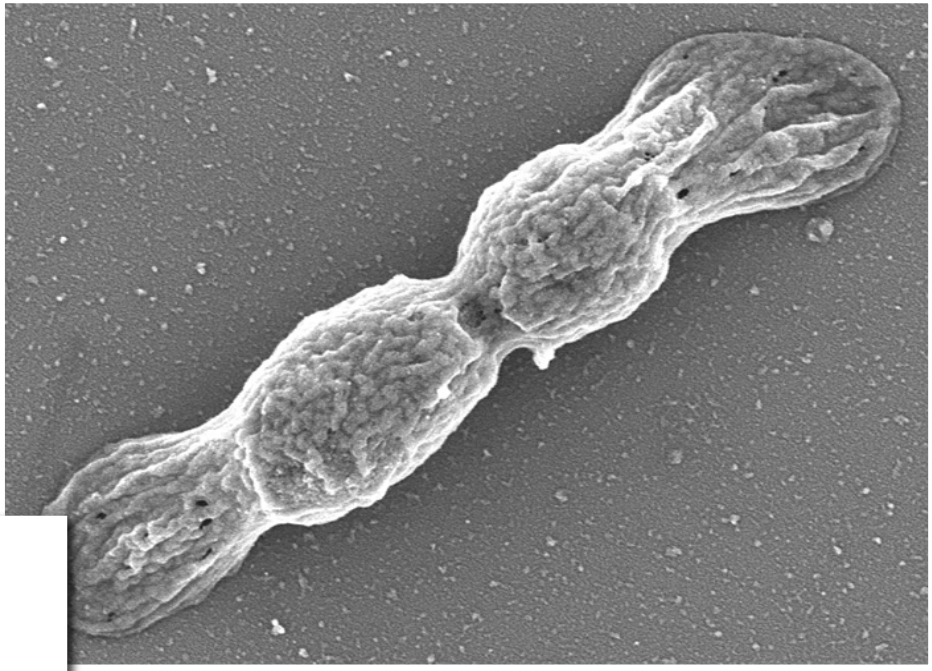
The BranchClust algorithm is implemented in Perl with the use of the BioPerl module for parsing trees and is freely available at http://bioinformatics.org/branchclust

Required:

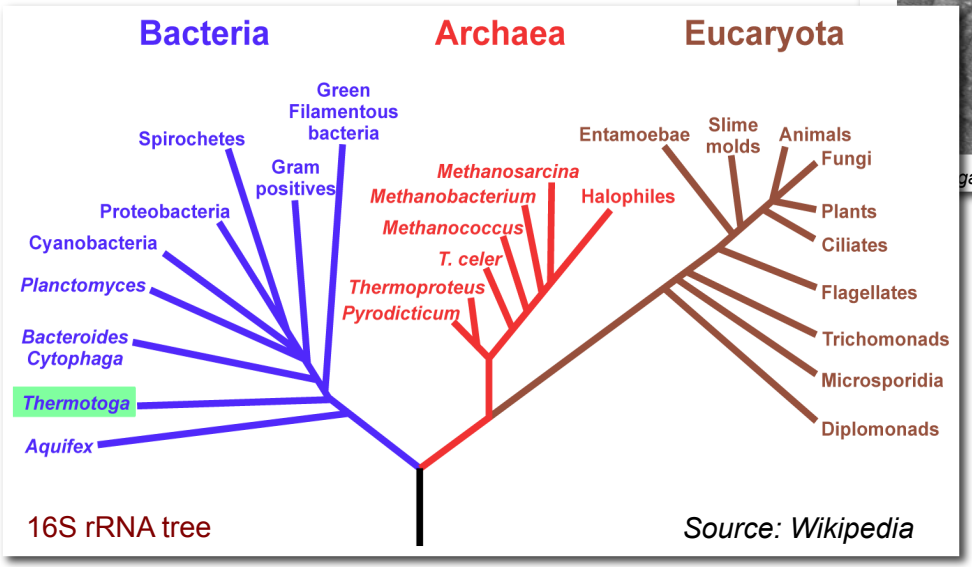
1.Bioperl module for parsing trees Bio::TreeIO
2. Taxa recognition file gi_numbers.out must be present in the current directory.
For information on how to create this file, read the Taxa recognition file section on the web-site.
3. Blastall from NCB needs to be installed.

Bioinformatics.Org

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



ga olearia. Courtesy of Kenneth Noll, UConn



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

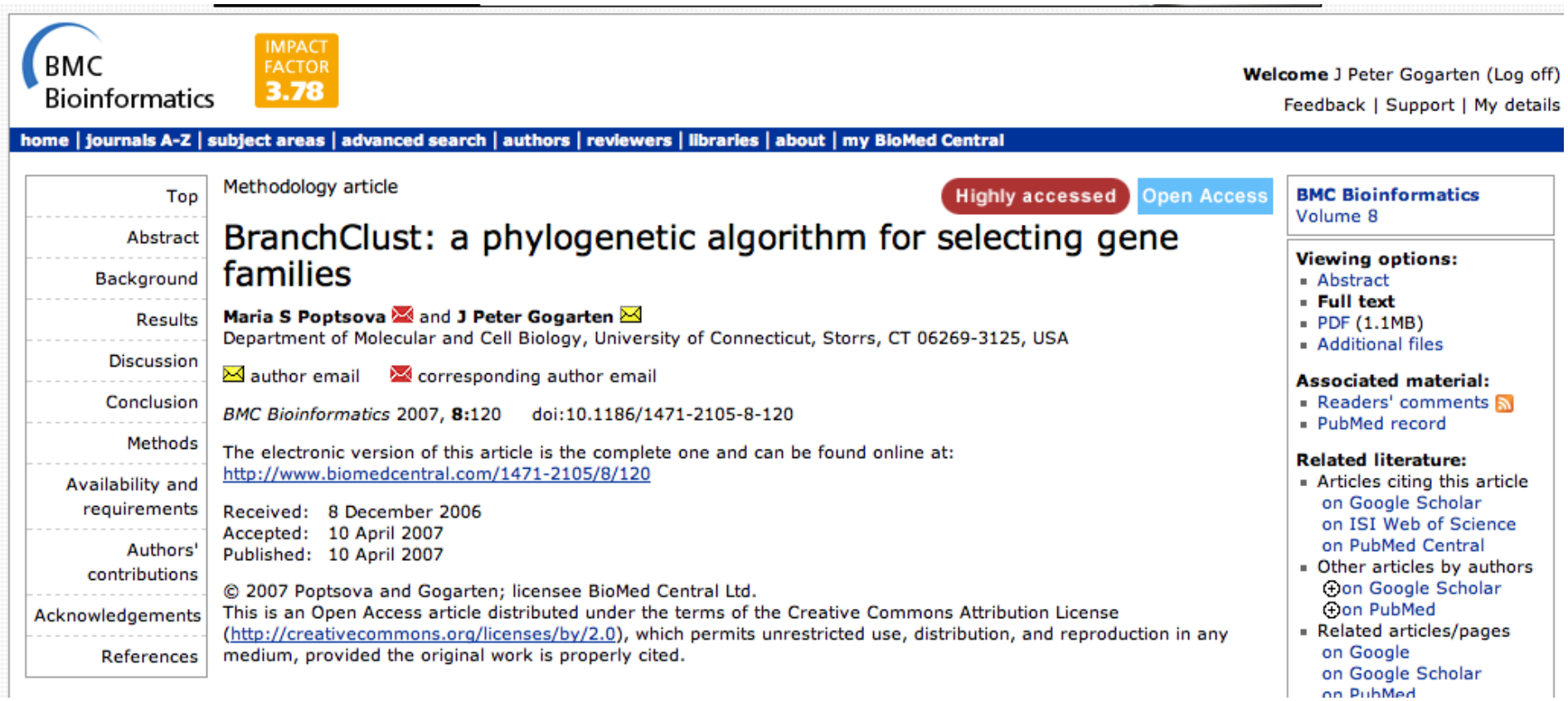
```
Burkholderia phage Bcep781 |     2375....     4783....     1179.....
Enterobacteria phage K1F | 7711....
Enterobacteria phage N4 |  1199.....
Enterobacteria phage P22 | 5123....     9635...      1271....
       193433..
Enterobacteria phage RB43 |      6639....
Enterobacteria phage T1 |  4568....
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.
- 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

Methodology article

Highly accessed | Open Access

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

BMC Bioinformatics
Volume 8

Viewing options:
- Abstract
- **Full text**
- PDF (1.1MB)
- Additional files

Associated material:
- Readers' comments
- PubMed record

Related literature:
- Articles citing this article
  on Google Scholar
  on ISI Web of Science
  on PubMed Central
- Other articles by authors
  on Google Scholar
  on PubMed
- Related articles/pages
  on Google
  on Google Scholar
  on PubMed

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

detailed_summary.out

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

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.

superfamily_##	ily_ ##	he_fa mily	nu_of_p 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

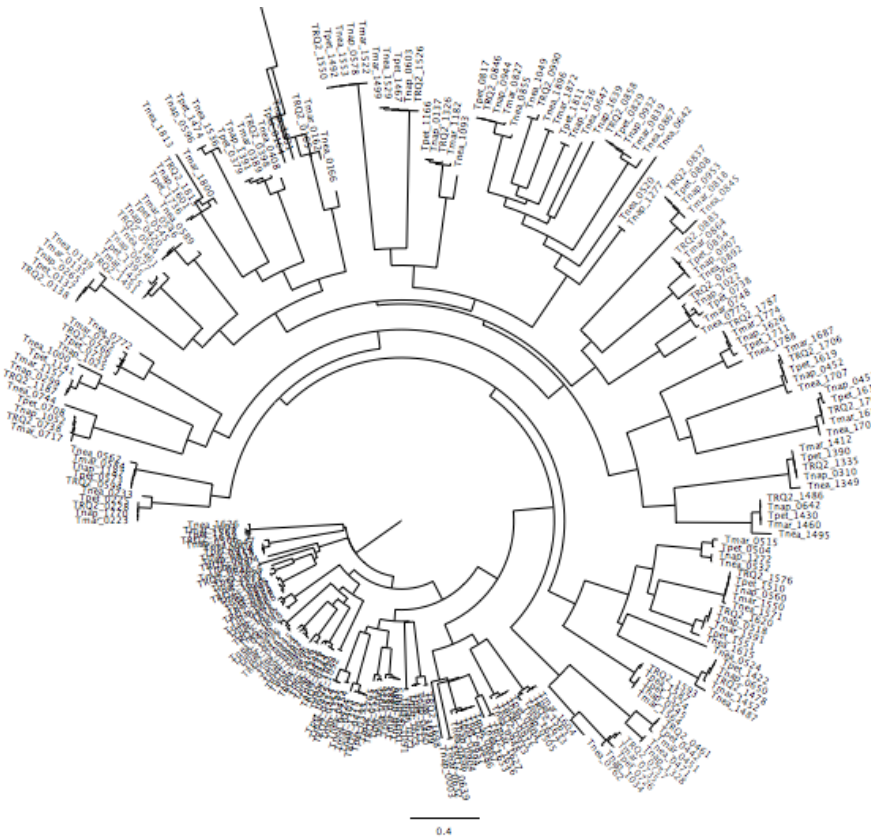
clusters/clusters_NNN.out.names

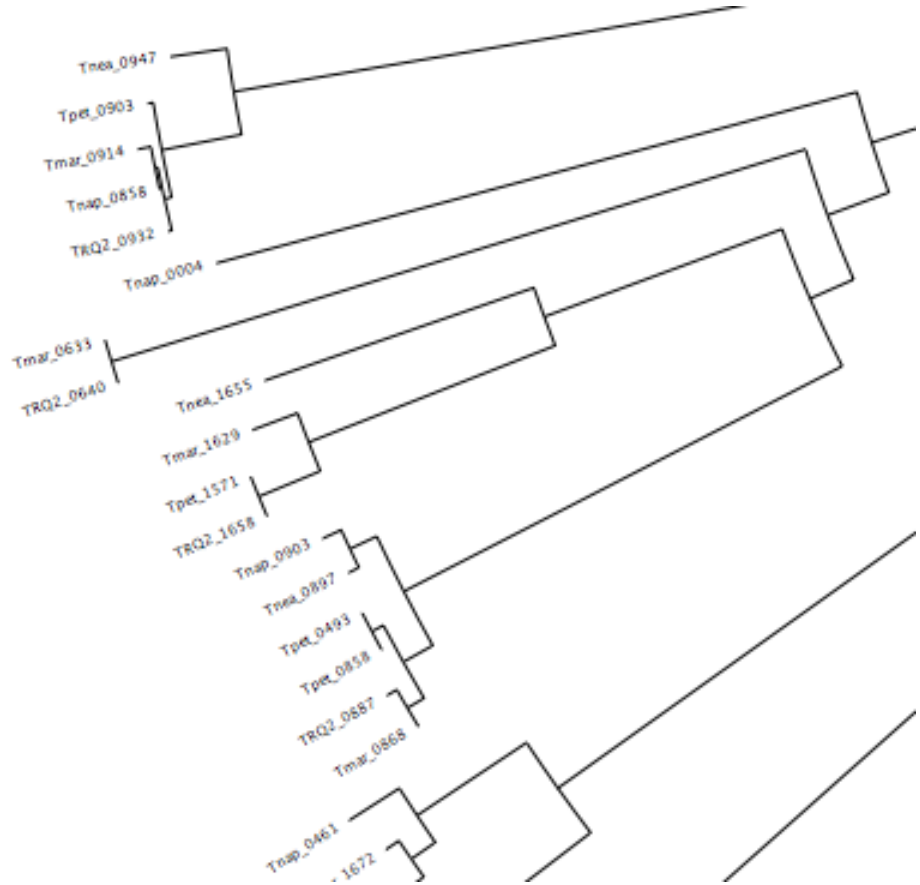
- Check a superfamily of your choice.
  Within a family, are all the annotation lines uniform?
- Within this report, if there are inparalogs, one is listed as a family member, the other one as inparalog. This is an arbitrary choice, both inparalogs from the same genome should be considered as being part 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





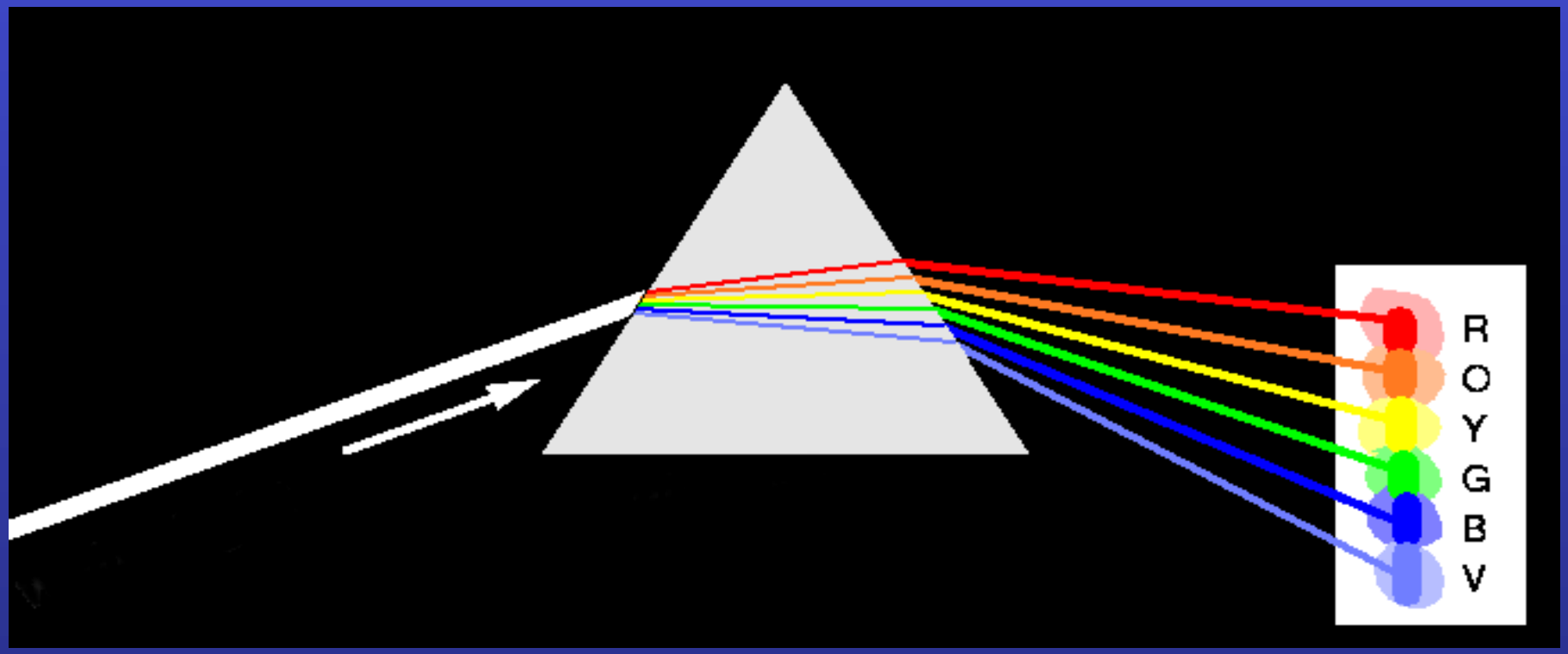
`prepare_bcfam_thermotoga.pl families_inpar_5_4_0.list`

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.

Decomposition of Phylogenetic Data



Phylogenetic information present in genomes

Break information into small quanta of information (bipartitions or embedded quartets)

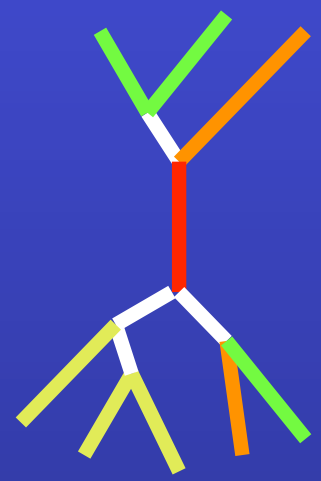
Analyze spectra to detect transferred genes and plurality consensus.

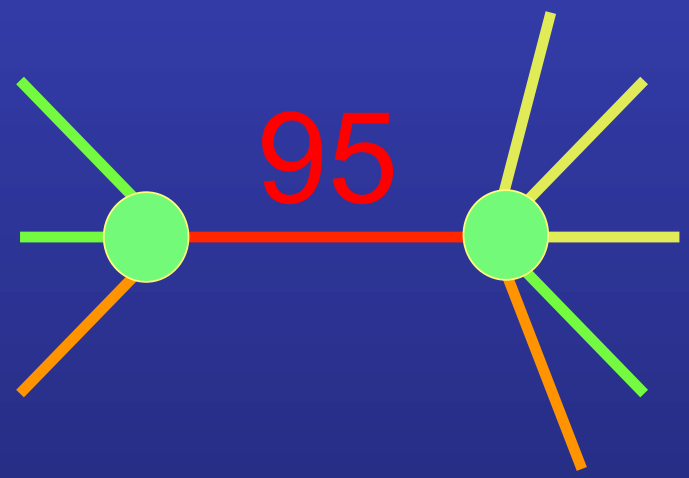
TOOLS TO ANALYZE PHYLOGENETIC INFORMATION FROM MULTIPLE GENES IN GENOMES:

Bipartition Spectra (Lento Plots)

BIPARTITION OF A PHYLOGENETIC TREE

Bipartition (or split) – a division of a phylogenetic tree into two parts that are connected by a single branch.
It divides a dataset into two groups, but it does not consider the relationships within each of the two groups.





* * * . . . . .

Yellow *vs* Rest

* * * . . . * *

compatible to illustrated bipartition

Orange vs Rest

. . * . . . . *

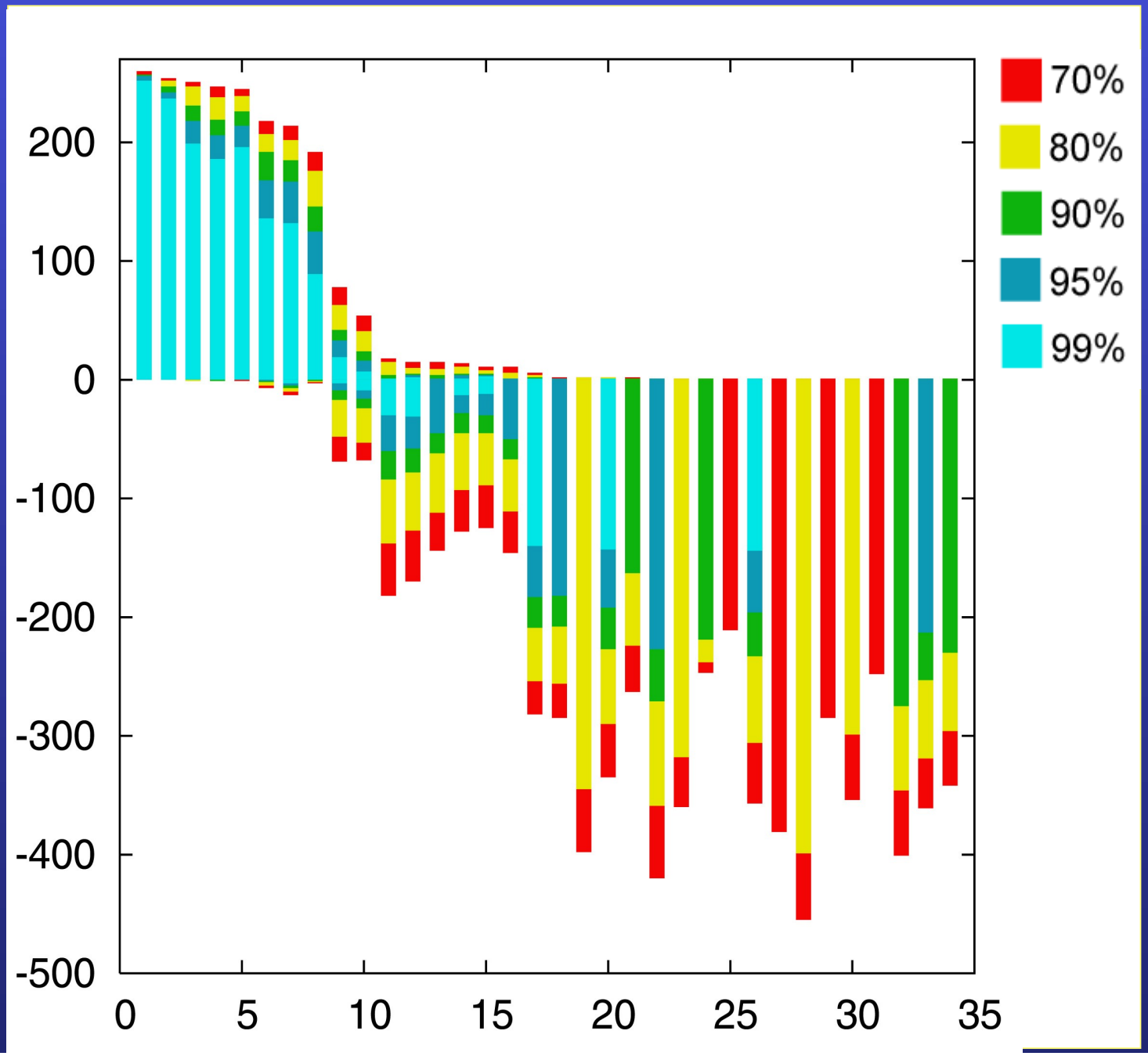
incompatible to illustrated bipartition

“Lento”-plot of 34 supported bipartitions (out of 4082 possible)

13 gamma-proteobacterial genomes (258 putative orthologs):

- E.coli
- Buchnera
- Haemophilus
- Pasteurella
- Salmonella
- Yersinia pestis (2 strains)
- Vibrio
- Xanthomonas (2 sp.)
- Pseudomonas
- Wigglesworthia

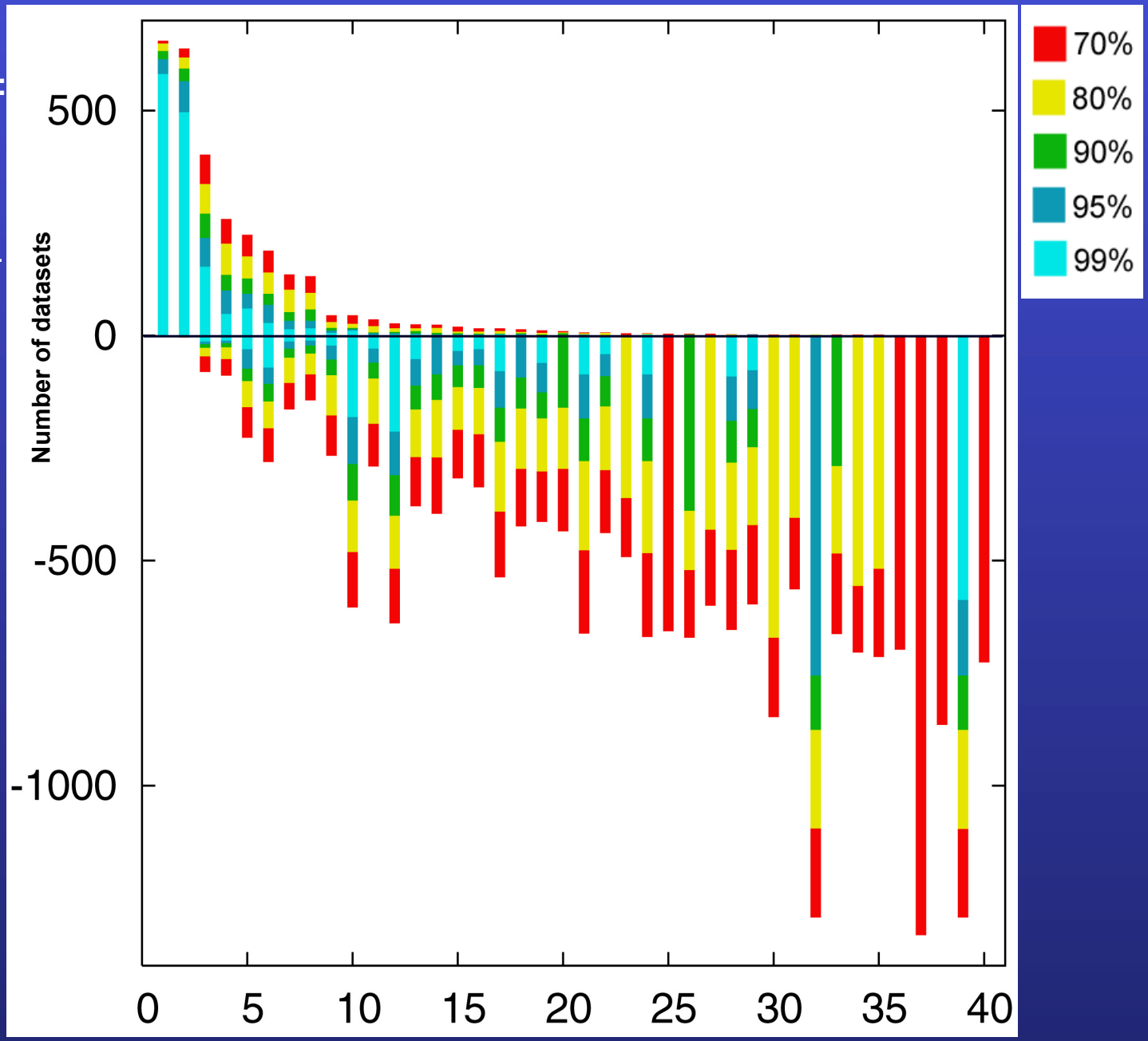
There are 13,749,310,575 possible unrooted tree topologies for 13 genomes



“Lento”-plot of supported bipartitions (out of 501 possible)

10 cyanobacteria:

- *Anabaena*
- *Trichodesmium*
- *Synechocystis* sp.
- *Prochlorococcus marinus* (3 strains)
- Marine *Synechococcus*
- *Thermo-synechococcus elongatus*
- *Gloeobacter*
- *Nostoc punctioforme*

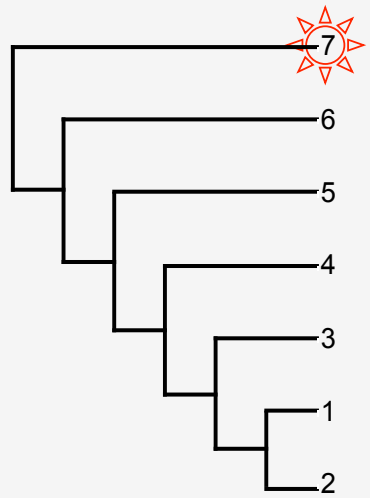


Based on 678 sets of orthologous genes

Zhaxybayeva, Lapierre and Gogarten, *Trends in Genetics*, 2004, **20(5)**: 254-260.

PROBLEMS WITH BIPARTITIONS (CONT.)

Q_1={
4 5 6 7
1 5 6 7
2 5 6 7
3 5 6 7
3 4 6 7
1 4 6 7
2 4 6 7
2 3 6 7
1 3 6 7
1 2 6 7
1 2 3 7
1 2 4 7
1 3 4 7
2 3 4 7
2 3 5 7
1 3 5 7
1 2 5 7
1 4 5 7
2 4 5 7
3 4 5 7
3 4 5 6
1 4 5 6
2 4 5 6
2 3 5 6
1 3 5 6
1 2 5 6
1 2 3 6
1 2 4 6
1 3 4 6
2 3 4 6
2 3 4 5
1 3 4 5
1 2 4 5
1 2 3 5
1 2 3 4}



B_1={
**.....,
***....,
****...,
*****..
}

bipartitions

embedded quartets

A single rogue sequence that moves from one end of a Hennigian comb to the other changes all bipartition

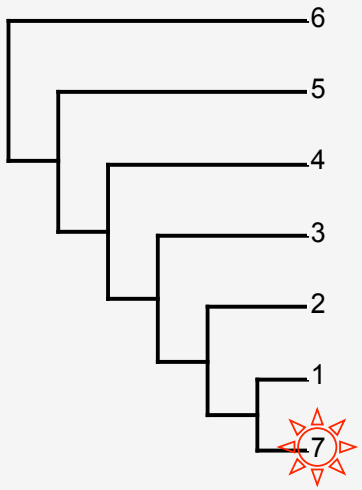
supported quartets

$Q1 \cap Q2 =$

{3 4 5 6, 1 4 5 6, 2 4 5 6, 2 3 5 6,
1 3 5 6, 1 2 5 6, 1 2 3 6, 1 2 4 6,
1 3 4 6, 2 3 4 6, 2 3 4 5, 1 3 4 5,
1 2 4 5, 1 2 3 5, 1 2 3 4}

supported bipartitions:

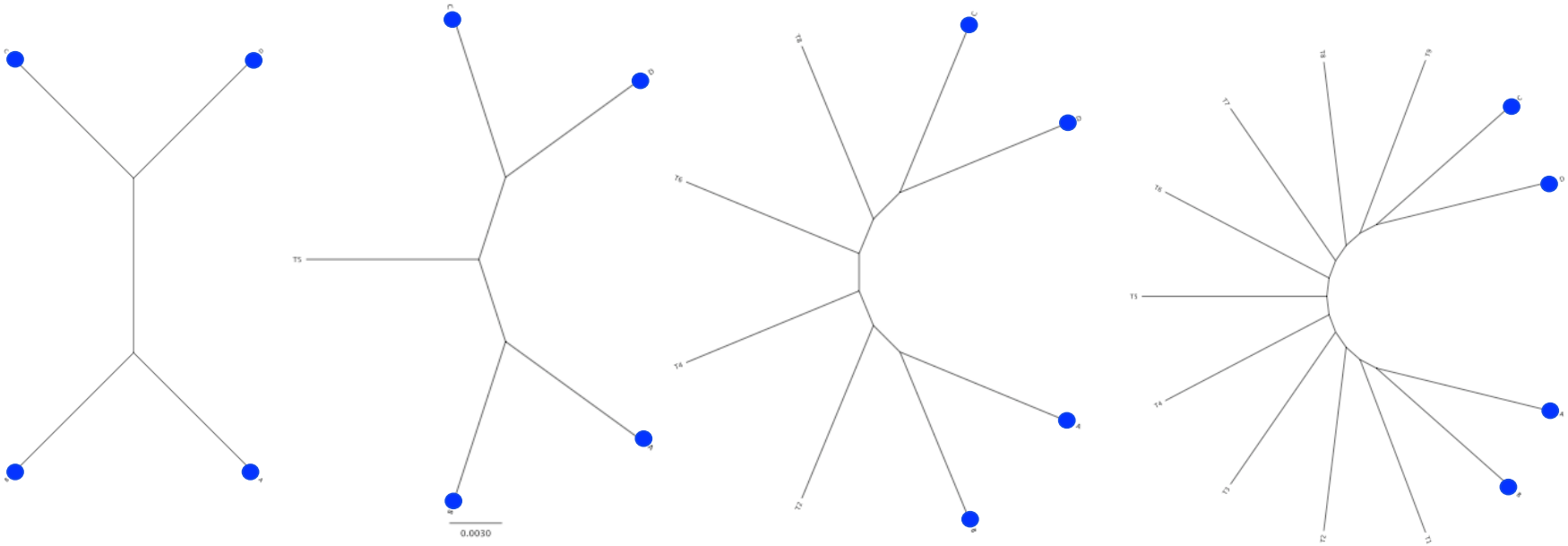
$B1 \cap B2 = \varnothing$



B_2={
.....,
**.....*,
***...*,
****..*
}

Q_2={
3 4 5 6
1 4 5 6
7 4 5 6
2 4 5 6
2 3 5 6
1 3 5 6
7 3 5 6
7 2 5 6
1 2 5 6
1 7 5 6
1 7 2 6
1 7 3 6
1 2 3 6
7 2 3 6
7 2 4 6
1 2 4 6
1 7 4 6
1 3 4 6
7 3 4 6
2 3 4 6
2 3 4 5
1 3 4 5
7 3 4 5
7 2 4 5
1 2 4 5
1 7 4 5
1 7 2 5
1 7 3 5
1 2 3 5
7 2 3 5
7 2 3 4
1 2 3 4
1 7 3 4
1 7 2 4
1 7 2 3}

Decay of bipartition support with number of OTUs



Phylogenies used for simulation

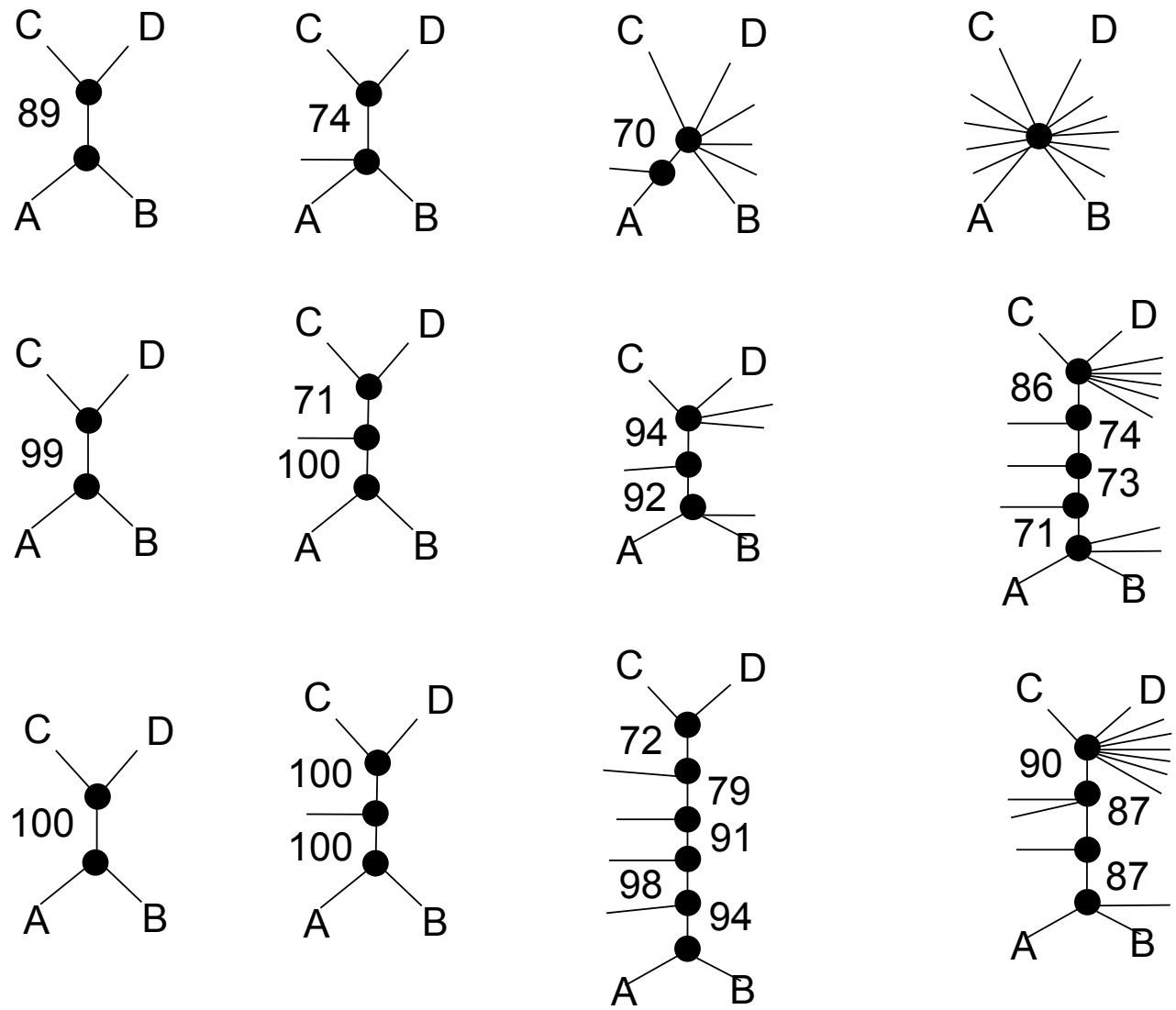
Example for decay of bipartition support with number of OTUs

Sequence lengths

200

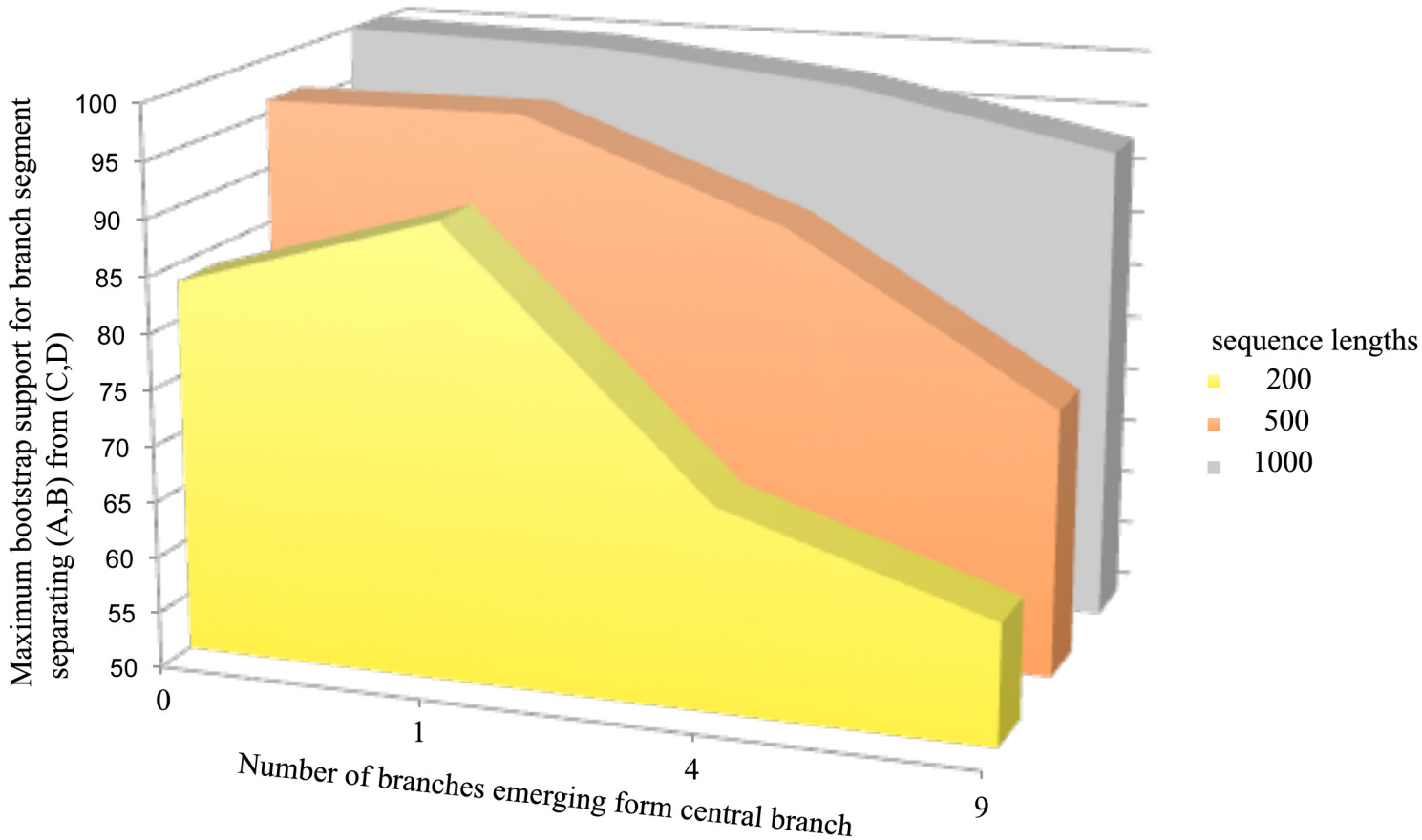
500

1000



Only branches with better than 70% bootstrap support are shown

Decay of bipartition support with number of OTUs



Each value is the average of 10 simulations using seq-gen.
Simulated sequences were evaluated using PHYML.
Model for simulation and evaluation WAG + Γ(α=1, 4 rate categories)

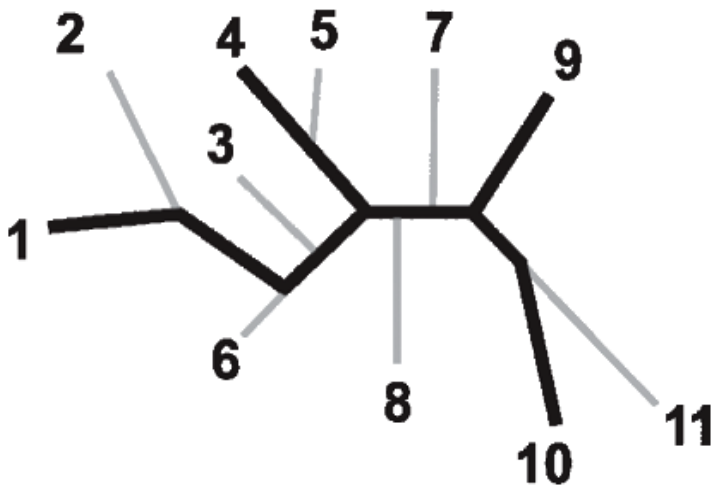
Bipartition Paradox:

- The more sequences are added, the lower the support for bipartitions that include all sequences. The more data one uses, the lower the bootstrap support values become.
- This paradox disappears when only embedded splits for 4 sequences are considered.

TOOLS TO ANALYZE PHYLOGENETIC INFORMATION FROM MULTIPLE GENES IN GENOMES:

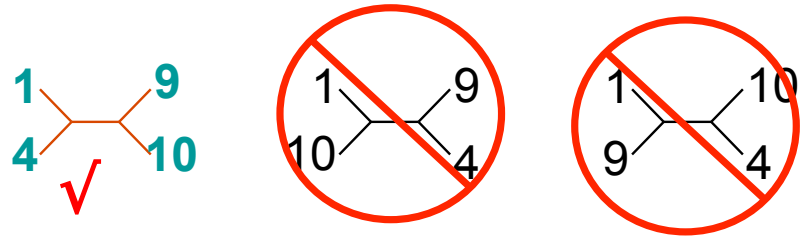
QUARTET DECOMPOSITION

Bootstrap support values for embedded quartets



+ : tree calculated from one pseudo-sample generated by bootstraping from an alignment of one gene family present in 11 genomes

: embedded quartet for genomes 1, 4, 9, and 10 .
This bootstrap sample supports the topology ((1,4),9,10).

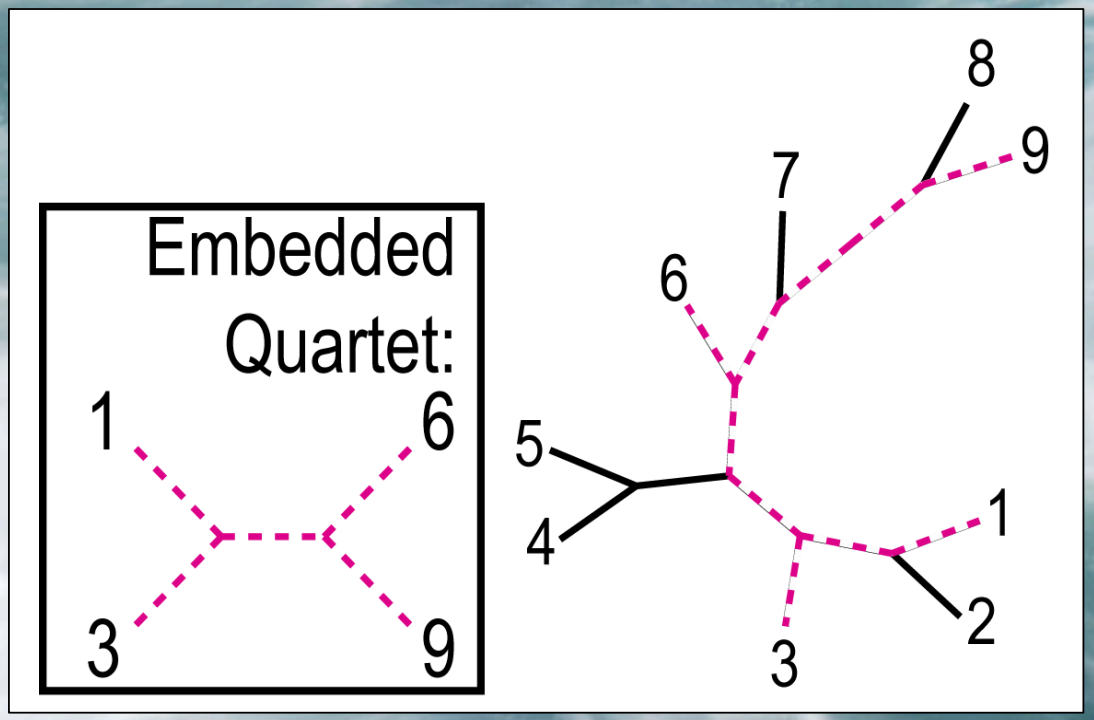


Quartet spectral analyses of genomes iterates over three loops:

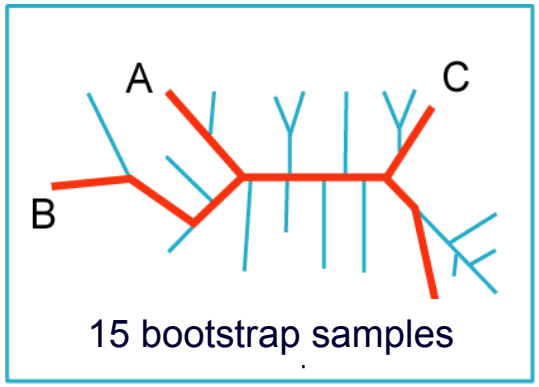
- **Repeat for all bootstrap samples.**
- **Repeat for all possible embedded quartets.**
- **Repeat for all gene families.**

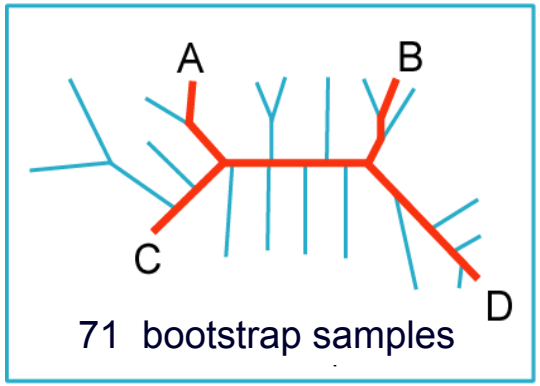
Zhaxybayeva et al. 2006, Genome Research, **16**(9):1099-108

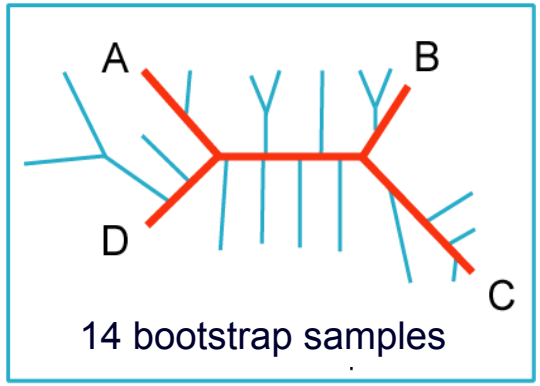
QUARTET DECOMPOSITION METHOD



- Quartet is a smallest unit of phylogenetic information
- Each quartet is associated with only three unrooted tree topologies
- Support for different quartet topologies can be summarized for all gene families







BOOTSTRAP SUPPORT VALUE VECTOR:

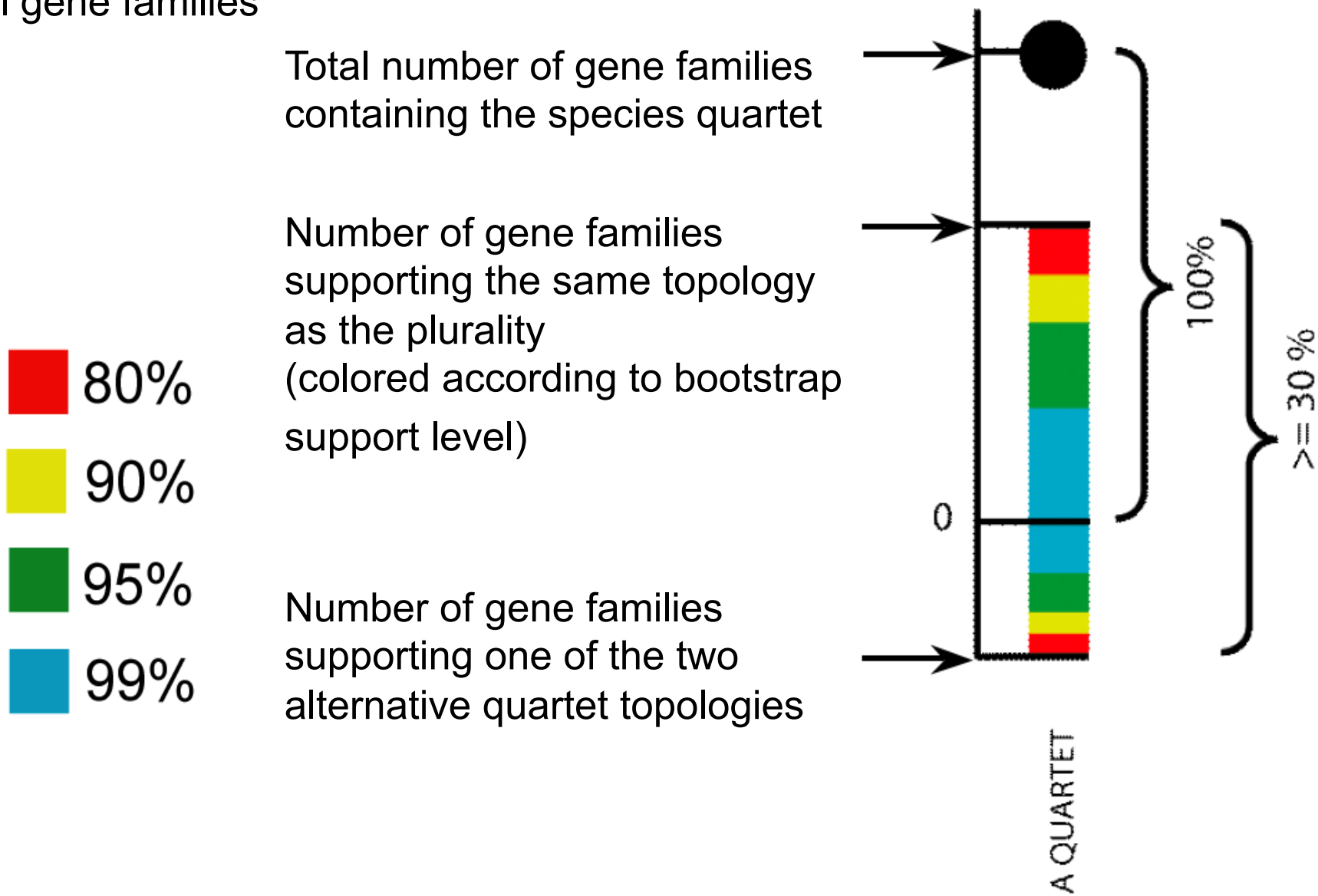
(15, 71, 14)

Illustration of one component of a quartet spectral analyses Summary of phylogenetic information for one genome quartet for all gene families

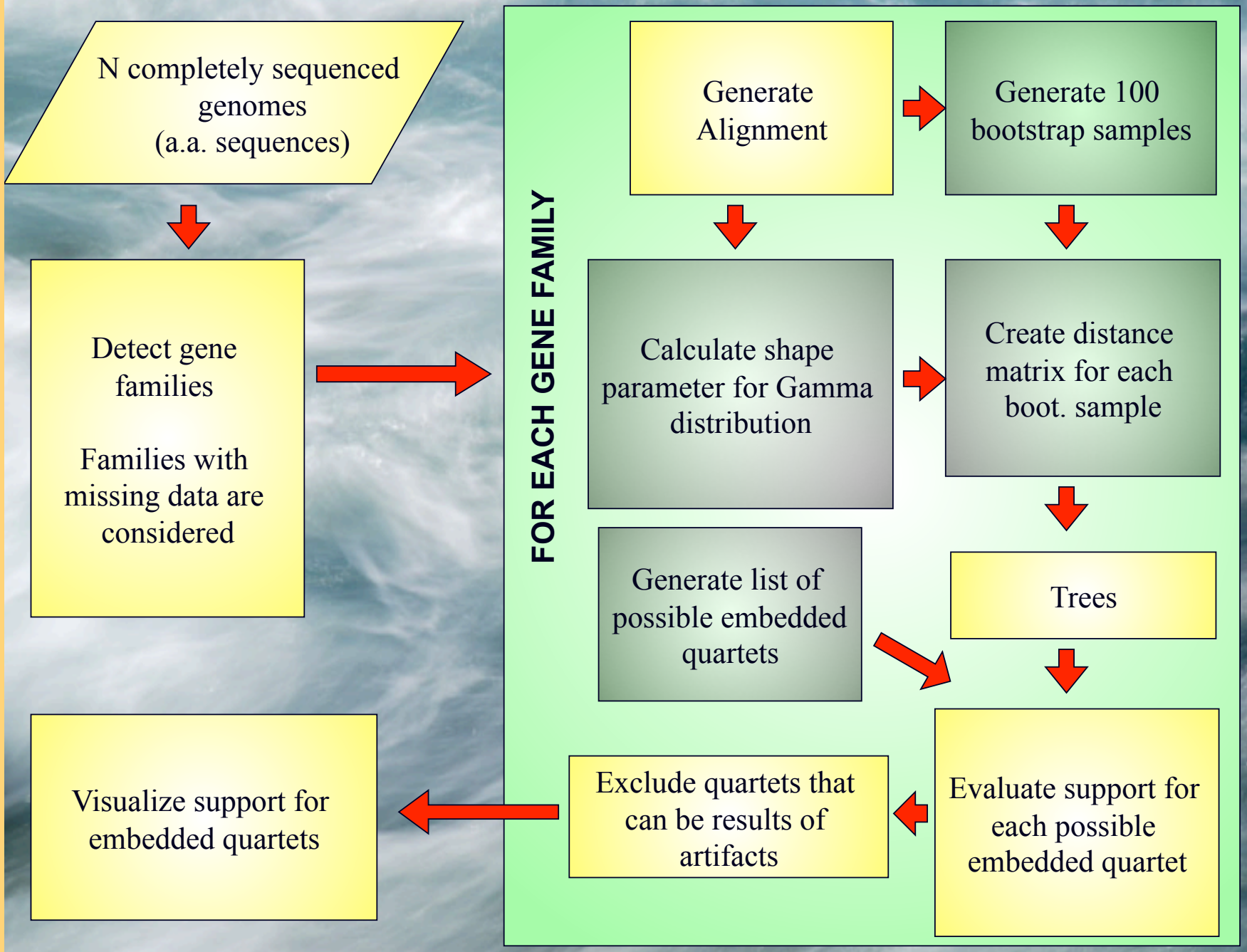
Total number of gene families containing the species quartet

Number of gene families supporting the same topology as the plurality (colored according to bootstrap support level)

Number of gene families supporting one of the two alternative quartet topologies

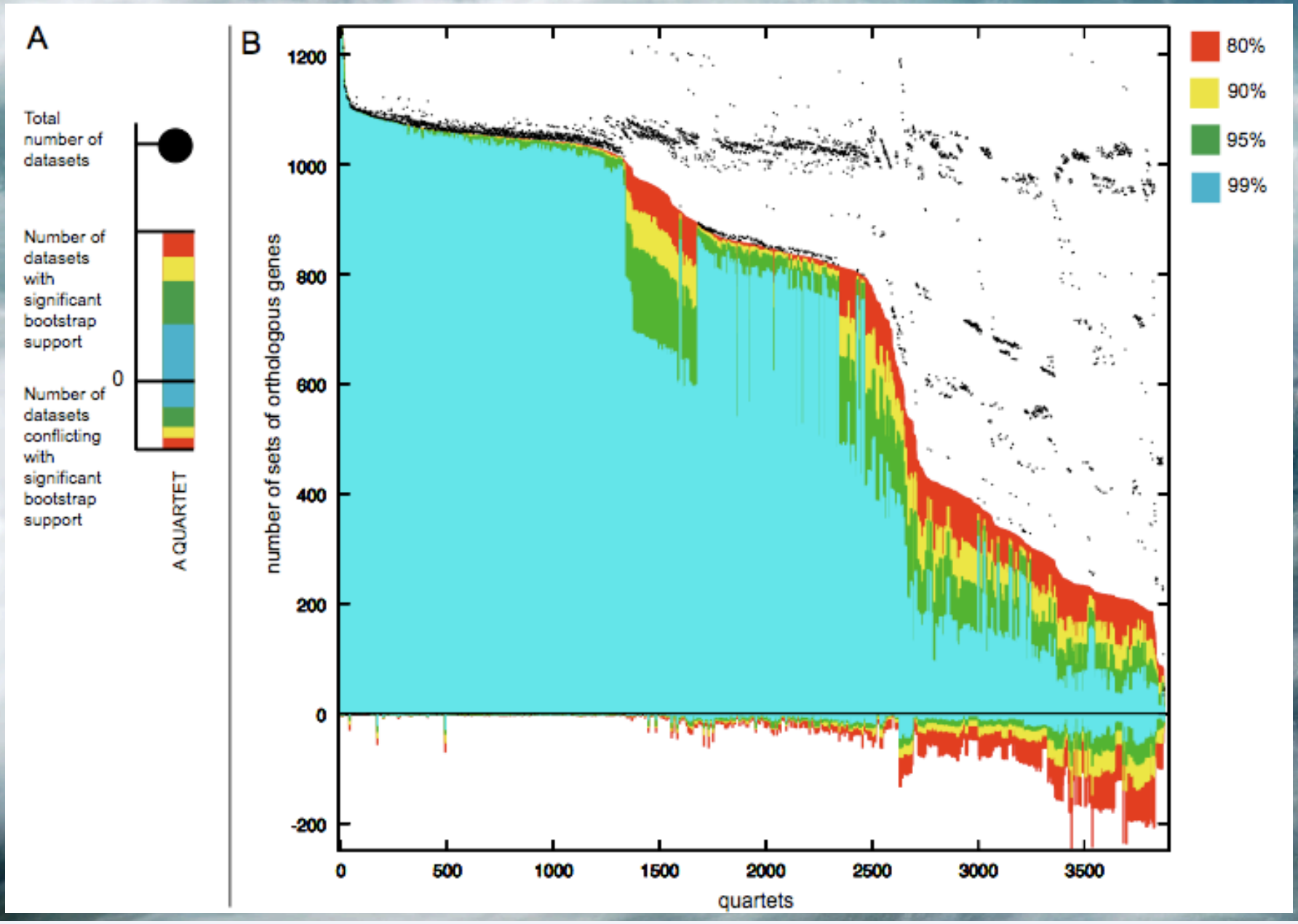


QUARTET DECOMPOSITION ANALYSES: DATA FLOW



Other POSITIVE THOUGHTS ABOUT THE METHOD

1. No assumption that all genes in a genome have the same phylogenetic history.
2. The total number of quartets is much smaller than number of tree topologies, which makes it possible to evaluate all quartets.
3. Gene families present only in few analyzed genomes can be included in the analyses
4. Phylogenetic signal can be divided into consensus supported by the plurality of gene families and the conflicting signal.
5. Allows us to partition analyzed genomes according to some scenario (e.g., grouping by ecology) and retrieve gene families that support or conflict it.



Quartet decomposition analysis of 19 *Prochlorococcus* and marine *Synechococcus* genomes. Quartets with a very short internal branch or very long external branches as well those resolved by less than 30% of gene families were excluded from the analyses to minimize artifacts of phylogenetic reconstruction.

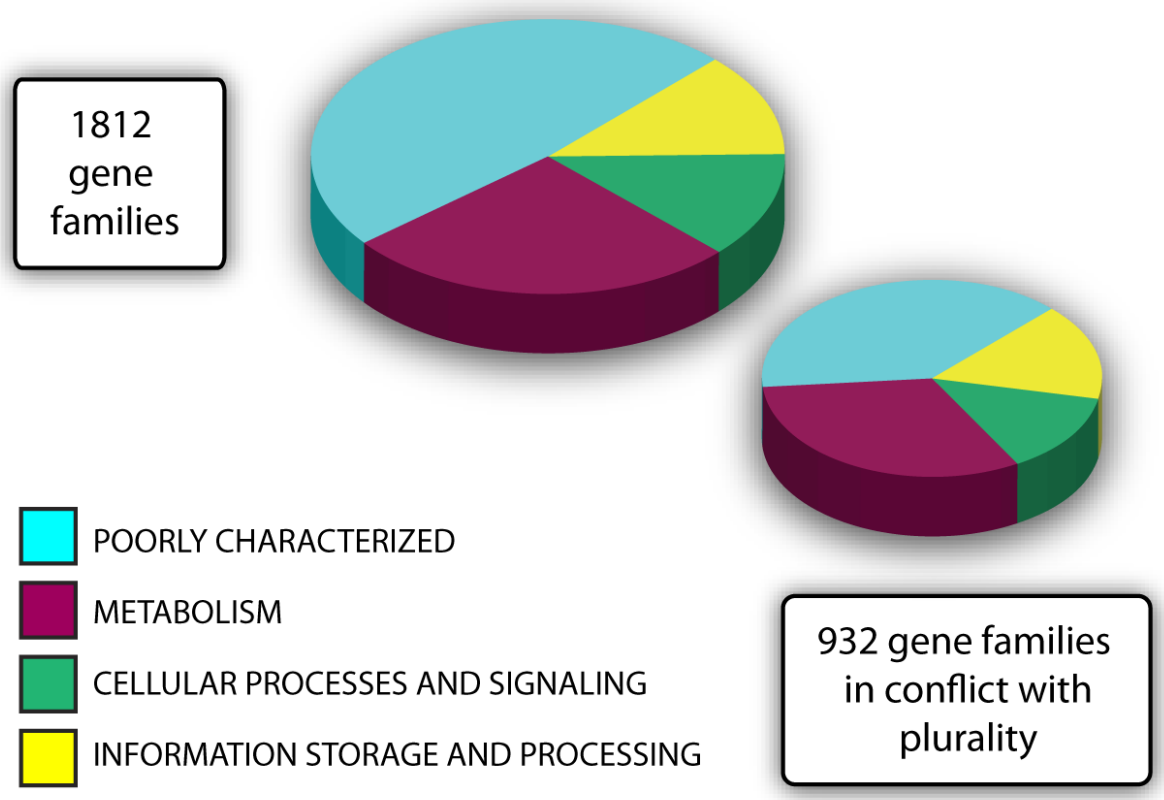
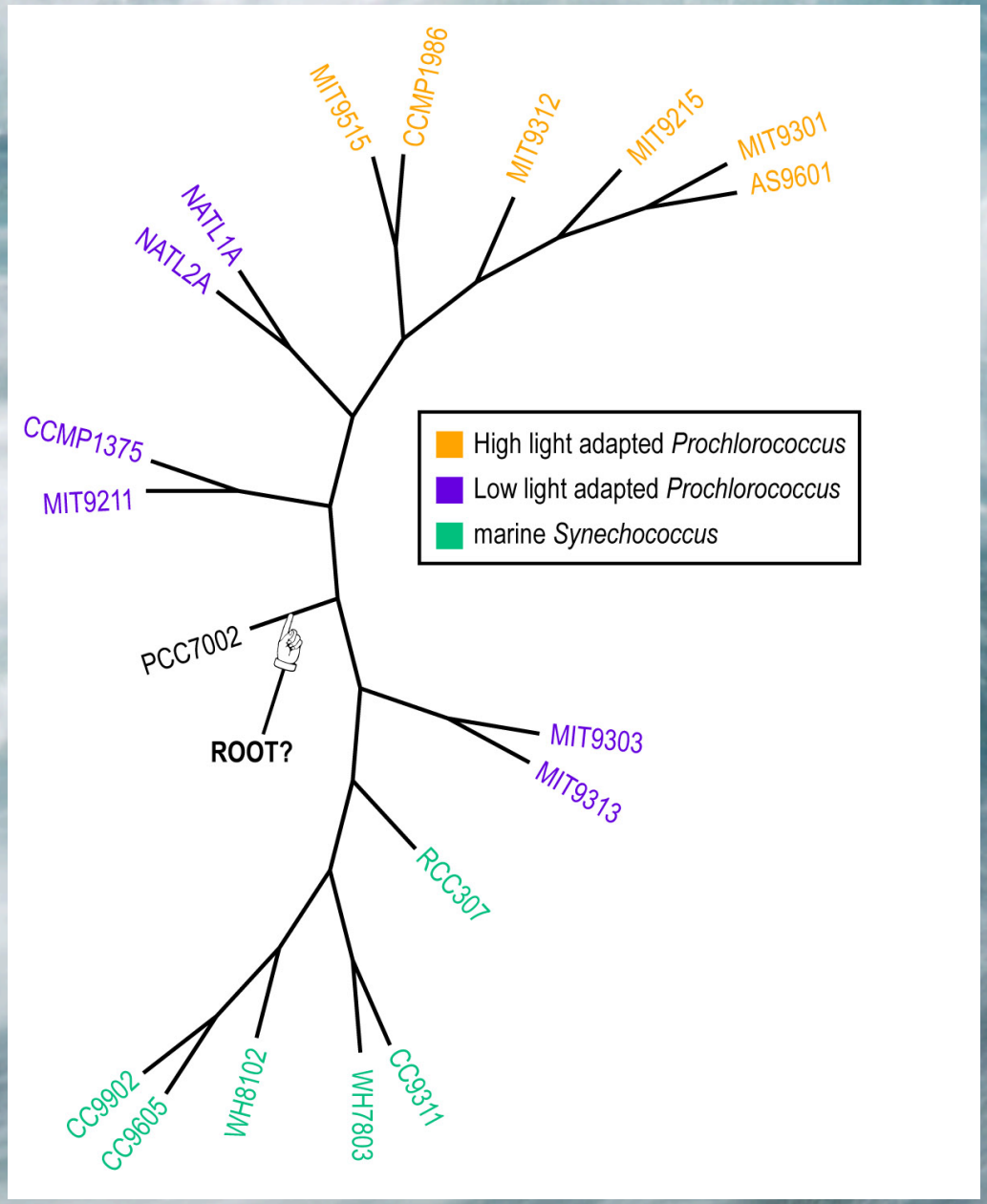
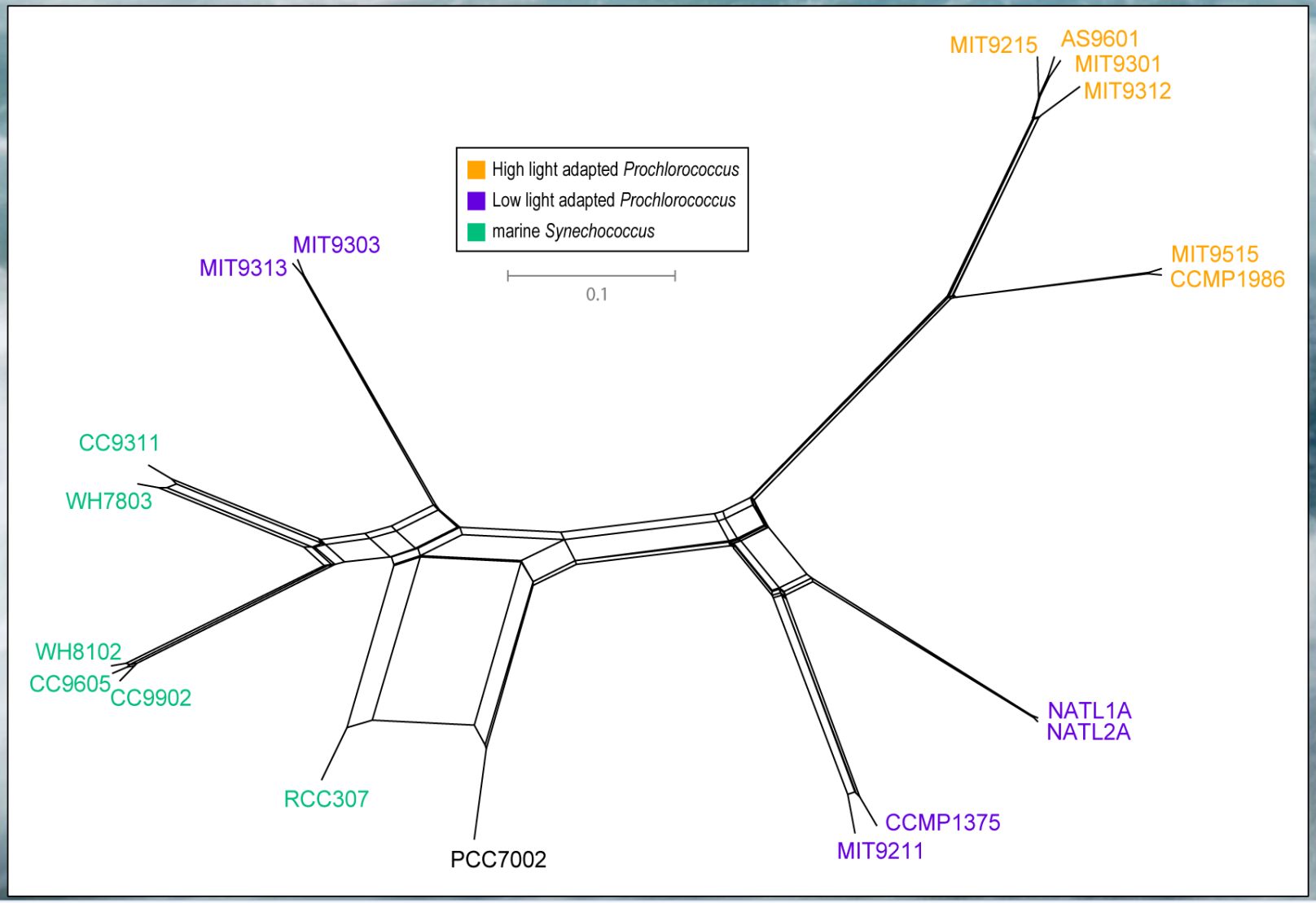


Figure 8. Distribution of gene families without paralogs across functional categories. The four super-categories are defined by COG database. Notably, genes of informational storage and processing are represented in equal proportions in genes in conflict with plurality as compared to all 1812 gene families, which contradicts complexity hypothesis {Jain, 1999 #31}. Metabolic genes appear to be overrepresented in the gene family pool which conflicts with plurality.



Plurality consensus calculated as supertree (MRP) from quartets in the plurality topology.

NeighborNet (calculated with SplitsTree 4.0)



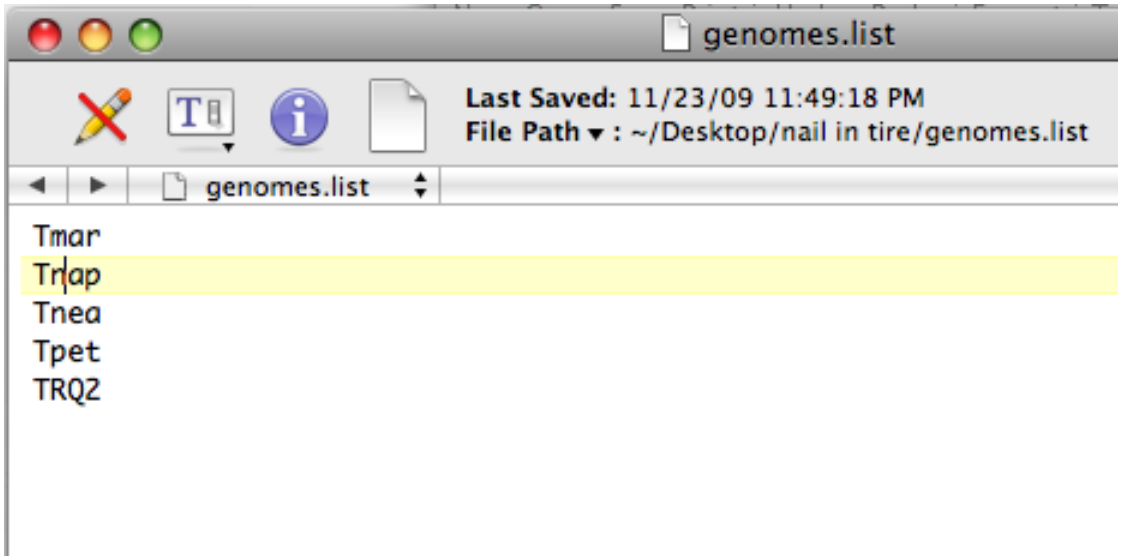
Plurality neighbor-net calculated as supertree (from the MRP matrix using SplitsTree 4.0) from all quartets significantly supported by all individual gene families (1812) without in-paralogs.

The Quartet Decomposition Server

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

Input A):

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

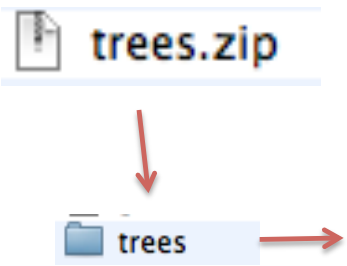


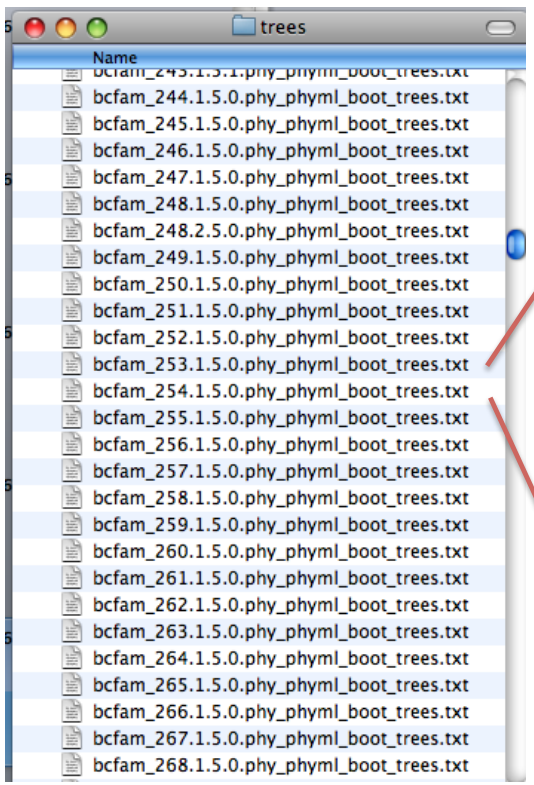
The Quartet Decomposition Server

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

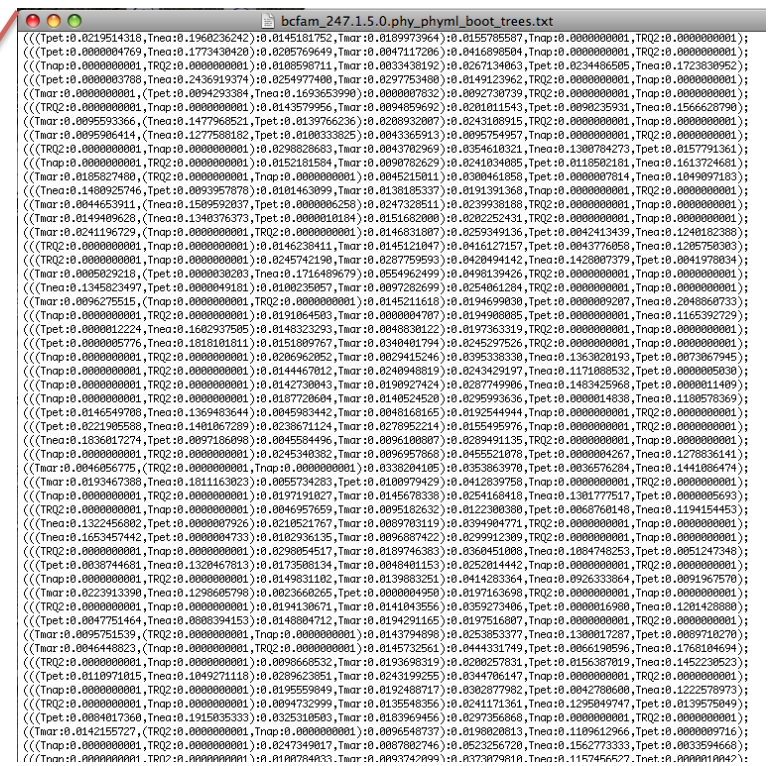
Input B):

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





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

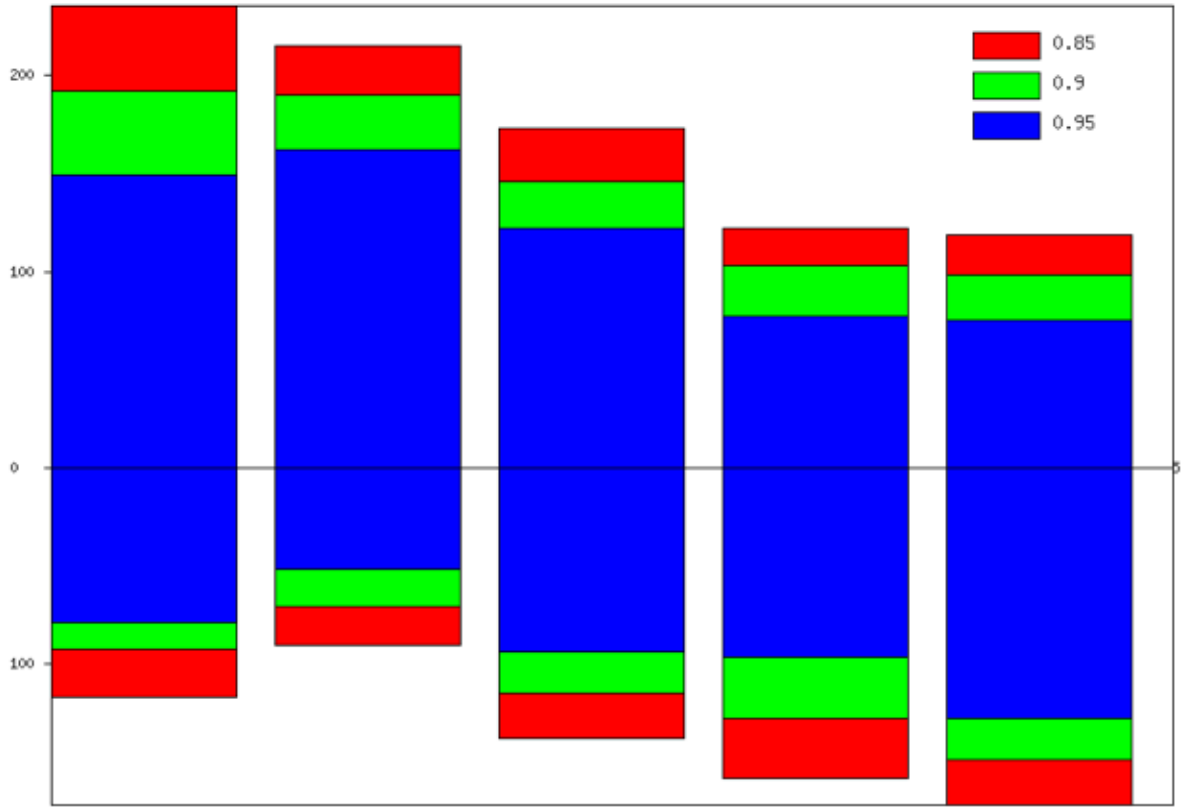
Quartet Decomposition

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));

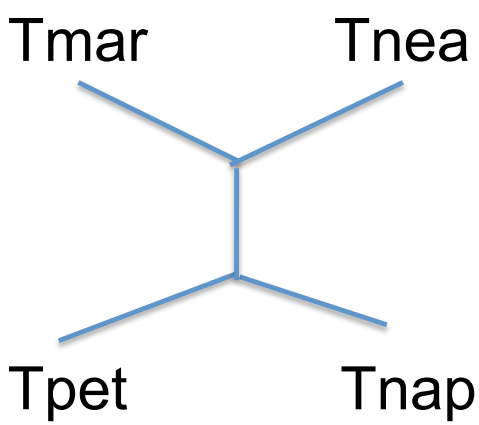
Quartet Decomposition

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
```

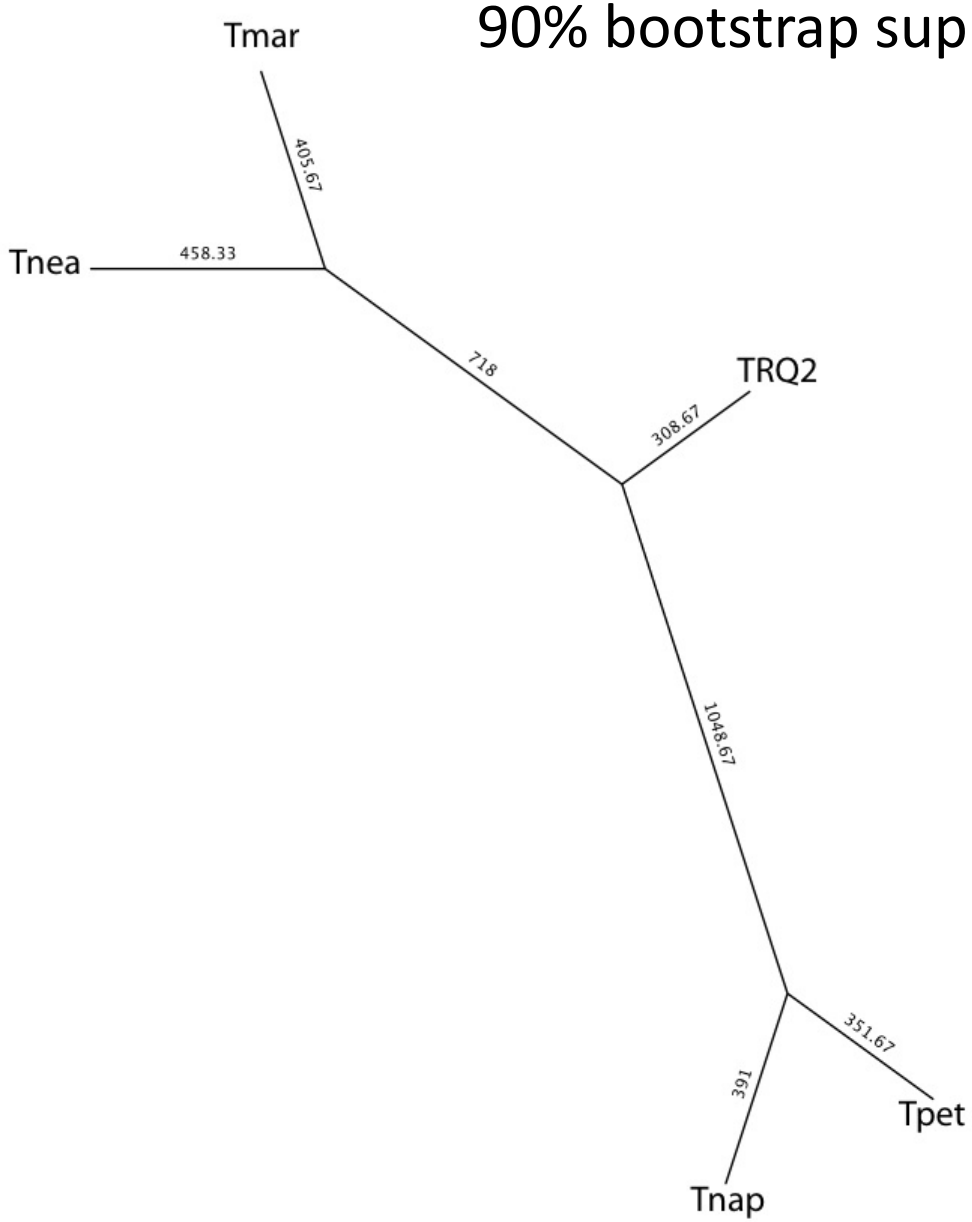
Quartet Decomposition

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

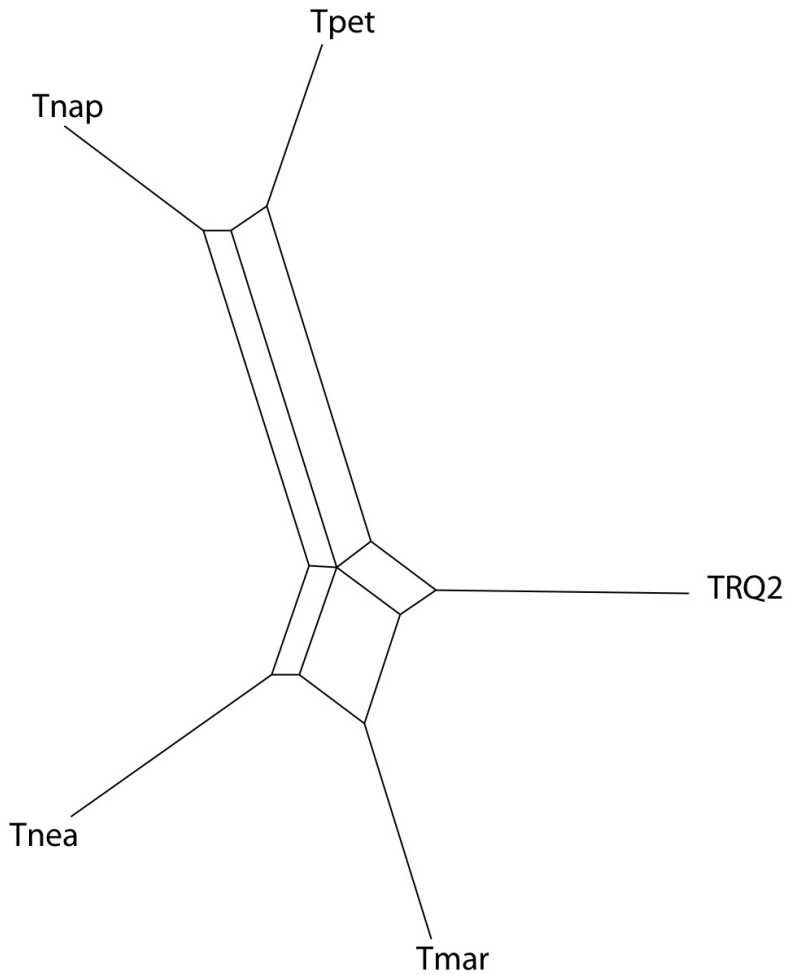
Quartet ID	Gene Family Numbers	Quartet Topology
1	192	((Tmar,Tnea),(Tnap,Tpet));
4	98	((Tmar,Tnea),(Tnap,TRQ2));
8	190	((Tmar,TRQ2),(Tnap,Tpet));
9	103	((Tmar,Tnea),(Tpet,TRQ2));
13	146	((Tnap,Tpet),(Tnea,TRQ2));

```
    5   2570
TRQ2      ??????????????????????????    101010101010101010
Tmar      101010101010101010101010101   ??????????????????
Tnap      010101010101010101010101010 ...101010101010101010
Tnea      101010101010101010101010101   010101010101010101
Tpet      010101010101010101010101010   010101010101010101
```

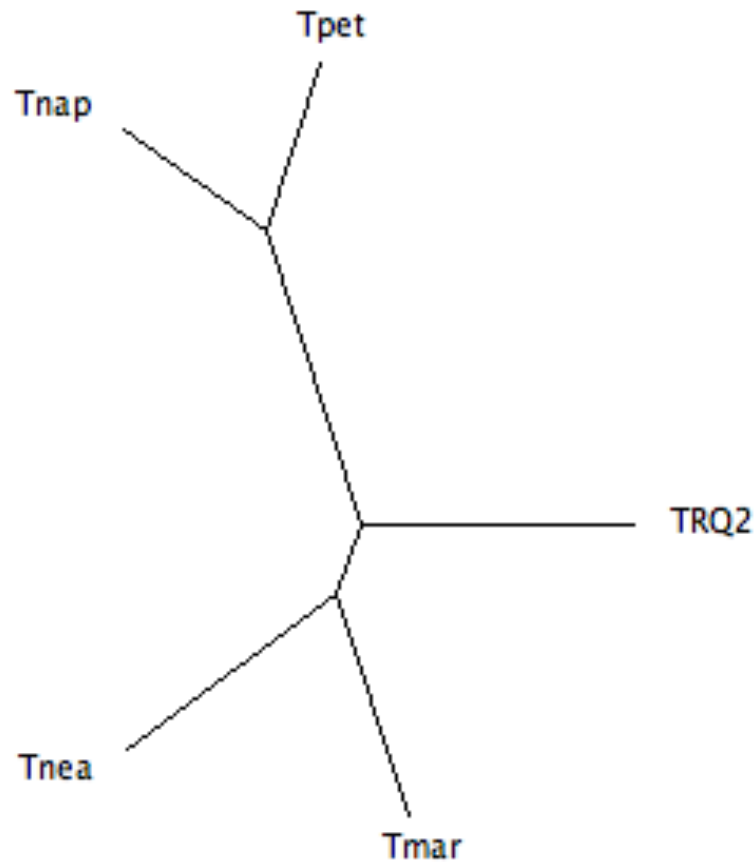
Most Parsimonious Tree (MRP)
Using all Quartets from all Gene Families that have more than 90% bootstrap support



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