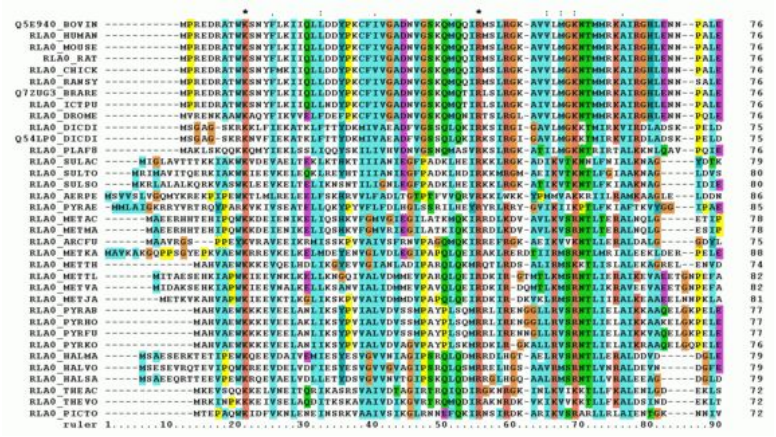
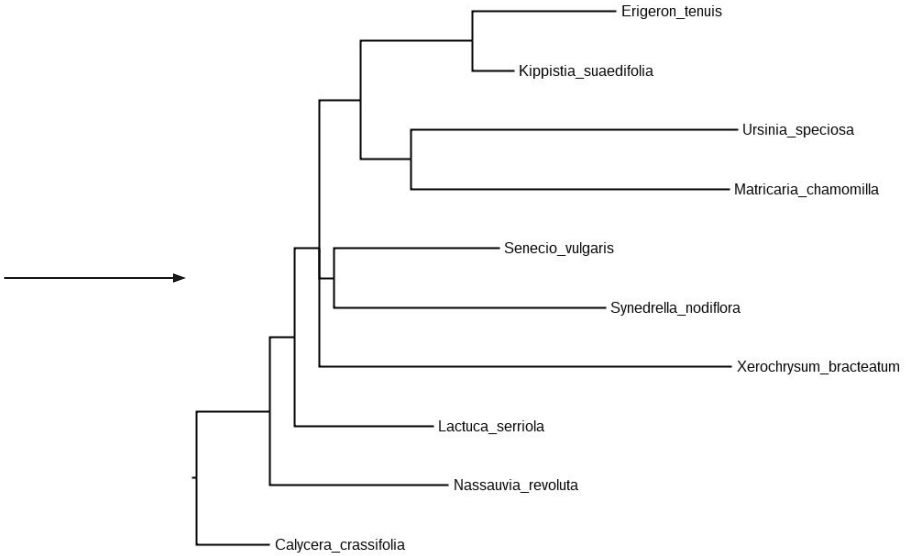
Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. a.k.a Gblock

Wei Qian
CS 581 Student Paper Presentation
March 29th. 2018

https://doi.org/10.1093/oxfordjournals.molbev.a026334

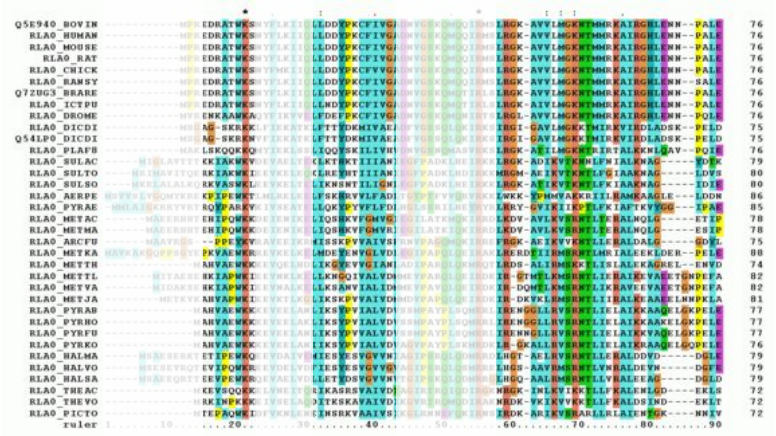
Abstract

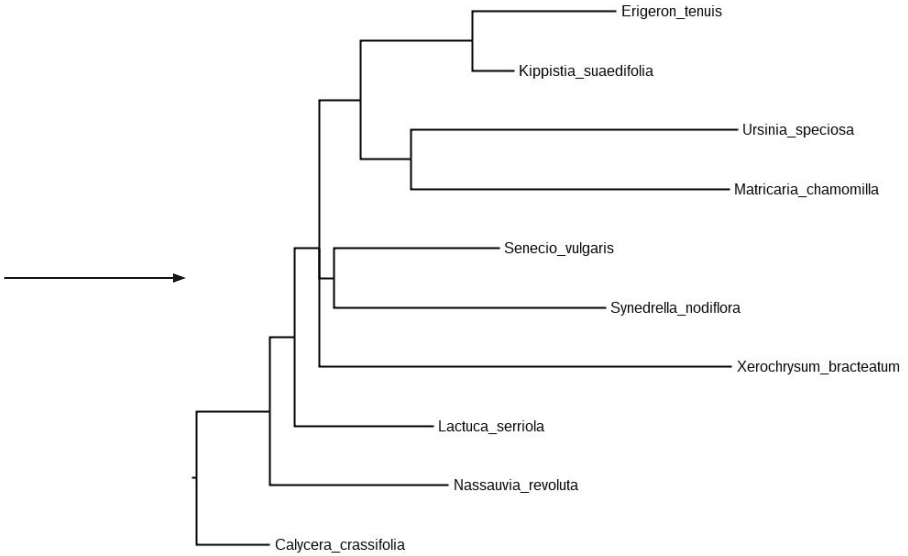




https://upload.wikimedia.org/wikipedia/commons/thumb/7/79/RPLP0_90_ClustalW_aln.gif/575px-RPLP0_90_ClustalW_aln.gif
https://4.bp.blogspot.com/-npAYS0fwDYE/WaAZESVZ6eI/AAAAAAAACr4/Xgo8e3zQDR0UanGQWf2ENFafIfj9NIVzACLcBGAs/s1600/nexml_sucks.png

Abstract





https://upload.wikimedia.org/wikipedia/commons/thumb/7/79/RPLP0_90_ClustalW_aln.gif/575px-RPLP0_90_ClustalW_aln.gif
https://4.bp.blogspot.com/-npAYS0fwDYE/WaAZESVZ6eI/AAAAAAAACr4/Xgo8e3zQDR0UanGQWf2ENFafIfj9NIVzACLcBGAs/s1600/nexml_sucks.png

Tree estimation is sensitive to the input alignment

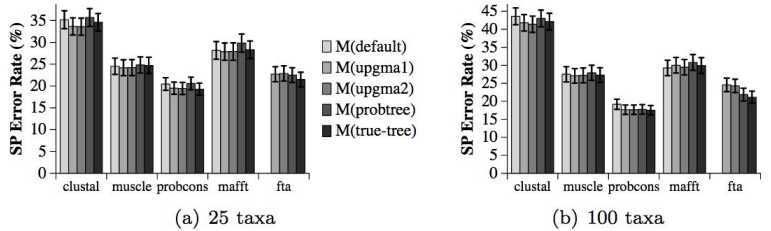


Figure 3. SP-error rates of alignments. M(guide tree) indicates multiple sequence alignment generated using the indicated guide tree.

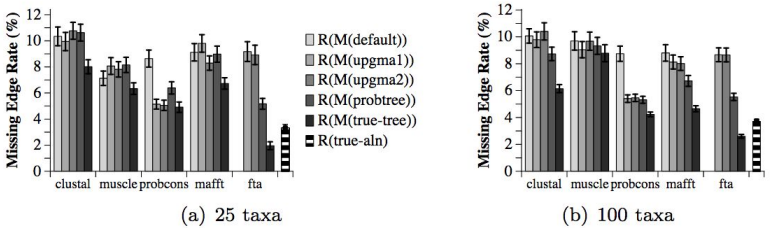


Figure 4. Missing edge rate of estimated trees. R(M(guide tree) indicates RAxML run on the alignment generated by the multiple sequence alignment method using the guide tree indicated. R(true-aln) indicates the tree generated by RAxML when given the true alignment.

Figure from Nelesen et al., Pacific Symposium on Biocomputing, 2008

Motivation

- Phylogenetic analysis assume the alignment sequences are **homologous in every position**
- Therefore, sequence shouldn't be
  - too **similar** so they are not **informative**
  - too **divergent** so there are **multiple substitutions** (i.e. saturated) that have erased the phylogenetic information
- In a single alignment/sequence, different regions of a gene can **evolve in a different rate**
  - therefore, not all regions are suitable for phylogenetic analysis
- Common current (back in 2000) approach is usually **done arbitrarily**, and therefore hard to **reproduce**
- **Computational approaches** have been developed but they are far from ideal

Motivation

- Phylogenetic analysis assume the alignment sequences are **homologous in every position**
- Therefore, sequence shouldn't be
  - too **similar** so they are not **informative**
  - too **divergent** so there are **multiple substitutions** (i.e. saturated) that have erased the phylogenetic information
- In a single alignment/sequence, different regions of a gene can **evolve in a different rate**
  - therefore, not all regions are suitable for phylogenetic analysis
- Common current (back in 2000) approach is usually **done arbitrarily**, and therefore hard to **reproduce**
- **Computational approaches** have been developed but they are far from ideal

Motivation

- Phylogenetic analysis assume the alignment sequences are **homologous in every position**
- Therefore, sequence shouldn't be
  - too **similar** so they are not **informative**
  - too **divergent** so there are **multiple substitutions** (i.e. saturated) that have erased the phylogenetic information
- In a single alignment/sequence, different regions of a gene can **evolve in a different rate**
  - therefore, not all regions are suitable for phylogenetic analysis
- Common current (back in 2000) approach is usually **done arbitrarily**, and therefore hard to **reproduce**
- **Computational approaches** have been developed but they are far from ideal

Motivation

- Phylogenetic analysis assume the alignment sequences are **homologous in every position**
- Therefore, sequence shouldn't be
  - too **similar** so they are not **informative**
  - too **divergent** so there are **multiple substitutions** (i.e. saturated) that have erased the phylogenetic information
- In a single alignment/sequence, different regions of a gene can **evolve in a different rate**
  - therefore, not all regions are suitable for phylogenetic analysis
- Common current (back in 2000) approach is usually **done arbitrarily**, and therefore hard to **reproduce**
- **Computational approaches** have been developed but they are far from ideal

Motivation

- Phylogenetic analysis assume the alignment sequences are **homologous in every position**
- Therefore, sequence shouldn't be
  - too **similar** so they are not **informative**
  - too **divergent** so there are **multiple substitutions** (i.e. saturated) that have erased the phylogenetic information
- In a single alignment/sequence, different regions of a gene can **evolve in a different rate**
  - therefore, not all regions are suitable for phylogenetic analysis
- Common current (back in 2000) approach is usually **done arbitrarily**, and therefore hard to **reproduce**
- **Computational approaches** have been developed but they are far from ideal

Mitochondrial Genome Sequence

First Dataset

11 protein subunits of 16 + 1 species

out group

Second Dataset

5 protein subunits of 23 + 1 species

Alignment Method and Tree Estimation

ClustalW

J Mol Evol. 1996 Apr;42(4):459-68.

Model of amino acid substitution in proteins encoded by mitochondrial DNA.

Adachi J[1], Hasegawa M.

Author information

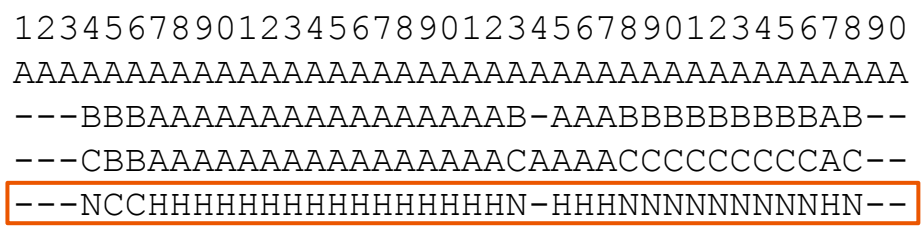
Abstract

Mitochondrial DNA (mtDNA) sequences are widely used for inferring the phylogenetic relationships among species. Clearly, the assumed model of nucleotide or amino acid substitution used should be as realistic as possible. Dependence among neighboring nucleotides in a codon complicates modeling of nucleotide substitutions in protein-encoding genes. It seems preferable to model amino acid substitution rather than nucleotide substitution. Therefore, we present a transition probability matrix of the general reversible Markov model of amino acid substitution for mtDNA-encoded proteins. The matrix is estimated by the maximum likelihood (ML) method from the complete sequence data of mtDNA from 20 vertebrate species. This matrix represents the substitution pattern of the mtDNA-encoded proteins and shows some differences from the matrix estimated from the nuclear-encoded proteins. The use of this matrix would be recommended in inferring trees from mtDNA-encoded protein sequences by the ML method.

First Dataset: MOLPHY + mtREV

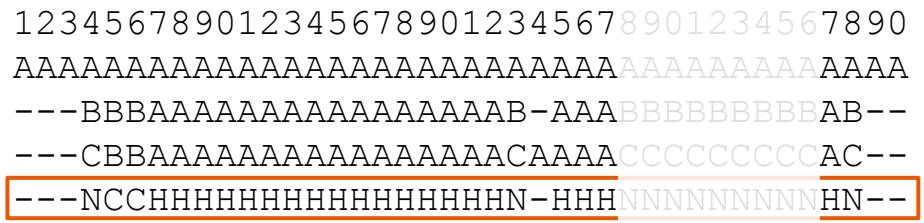
Second Dataset: NJ + mtREV + Heuristic

Gblock



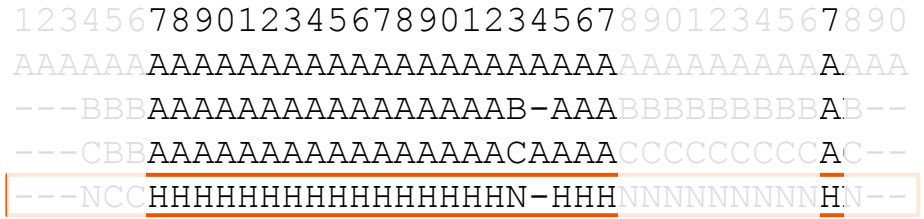
1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

Gblock



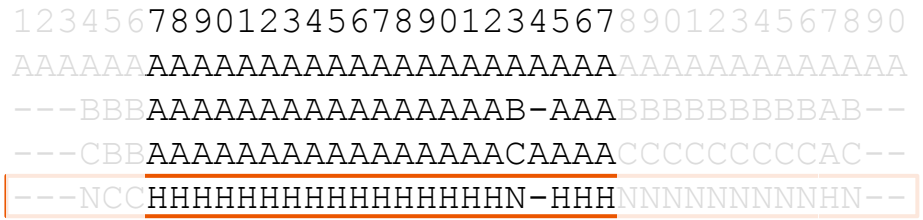
1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

Gblock



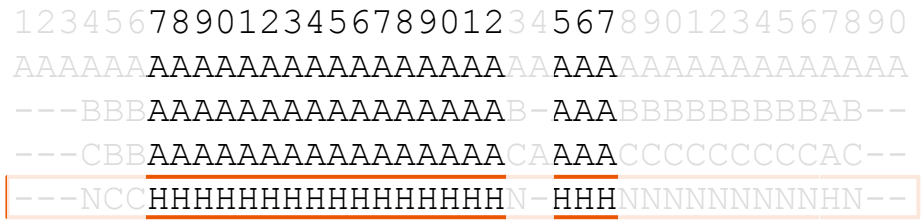
1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

Gblock



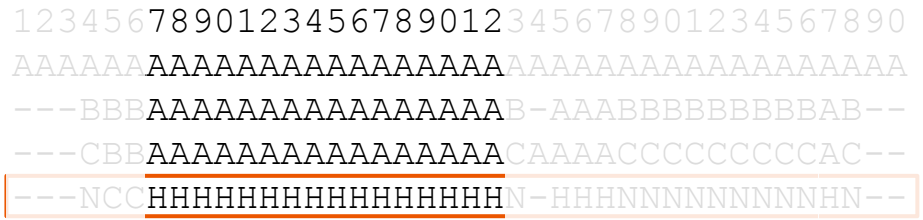
1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

Gblock



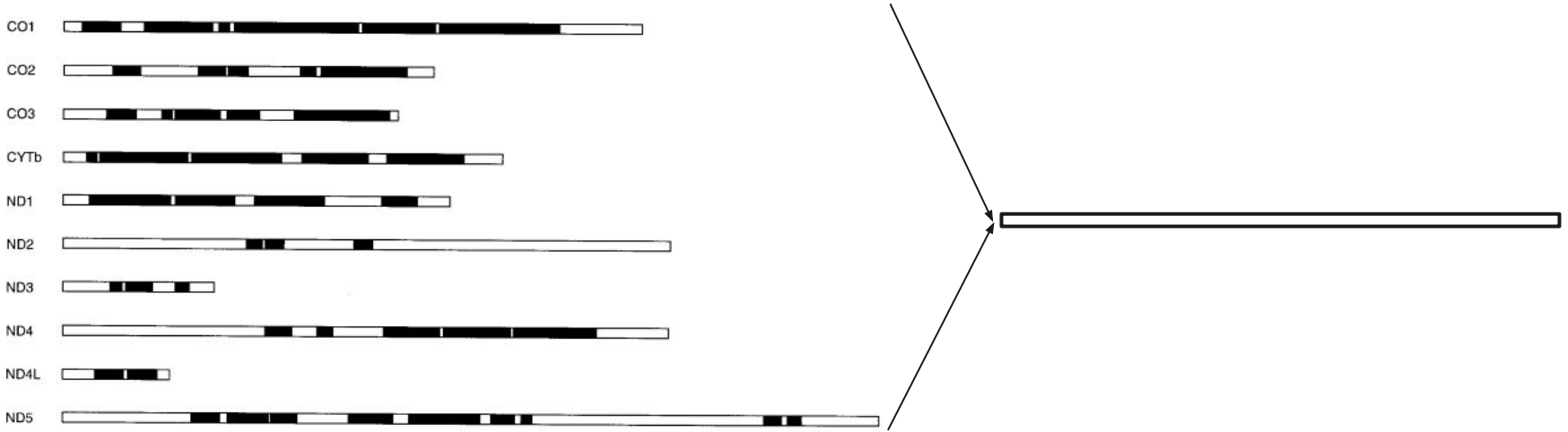
1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

Gblock



1. Calculate degree of conservation
   a. non-conserved < **IS** < conserved < **FS** < highly conserved
   b. IS = 50%, FS = 85%
2. Reject contiguous *non-conserved* position > **CP**
   a. CP = 8
3. Exam flank and reject columns until surround by *highly conserved* column
4. For the remaining positions, which form blocks, take the one with length >= **BL1**
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length >= **BL2**
   a. BL2 = 10

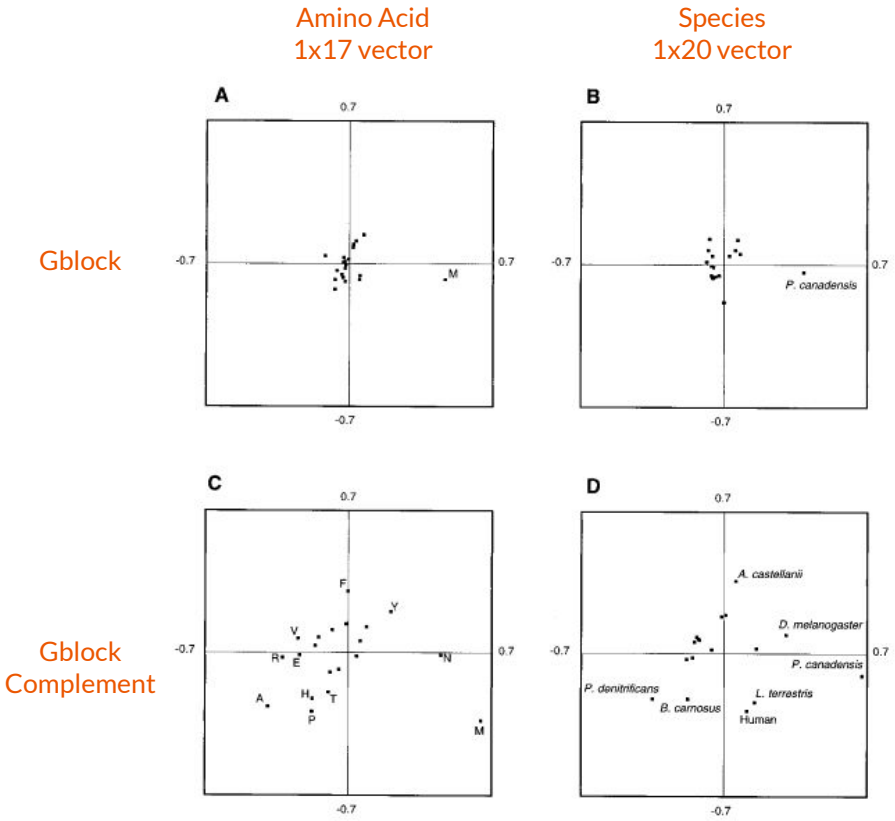
Concatenation



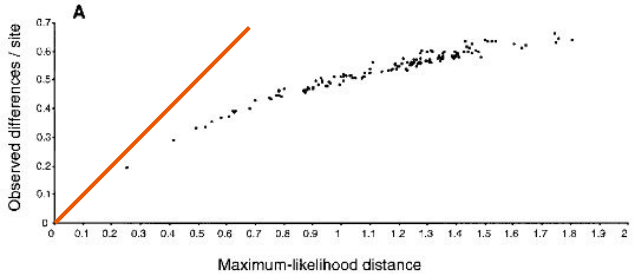
Amino acid composition is more uniform

for the first dataset with 17 species

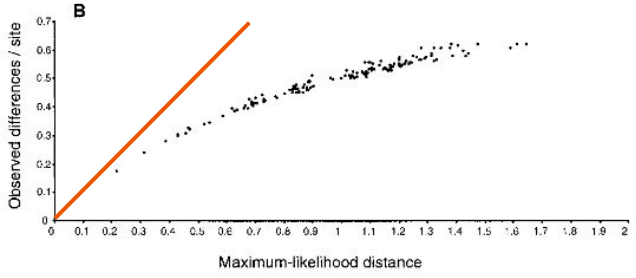
PCA dimension reduction



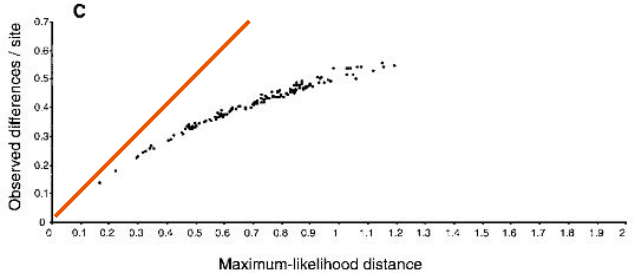
Tree branch length is more similar to pairwise distance



original



non-gap

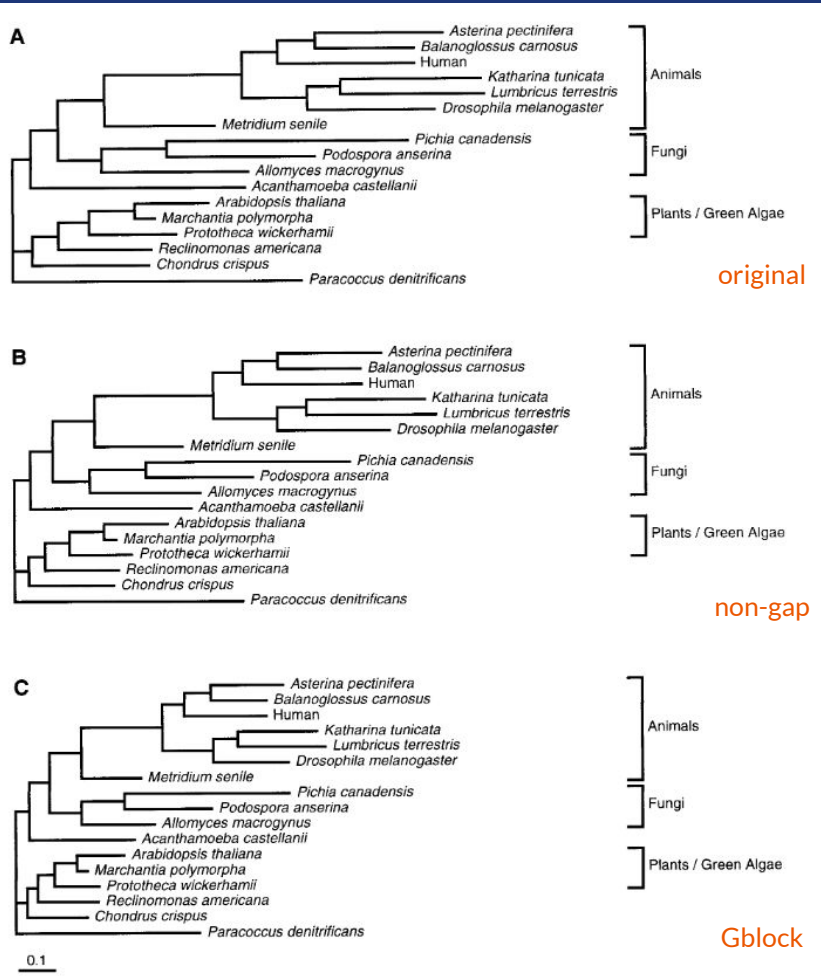


Gblock

Result

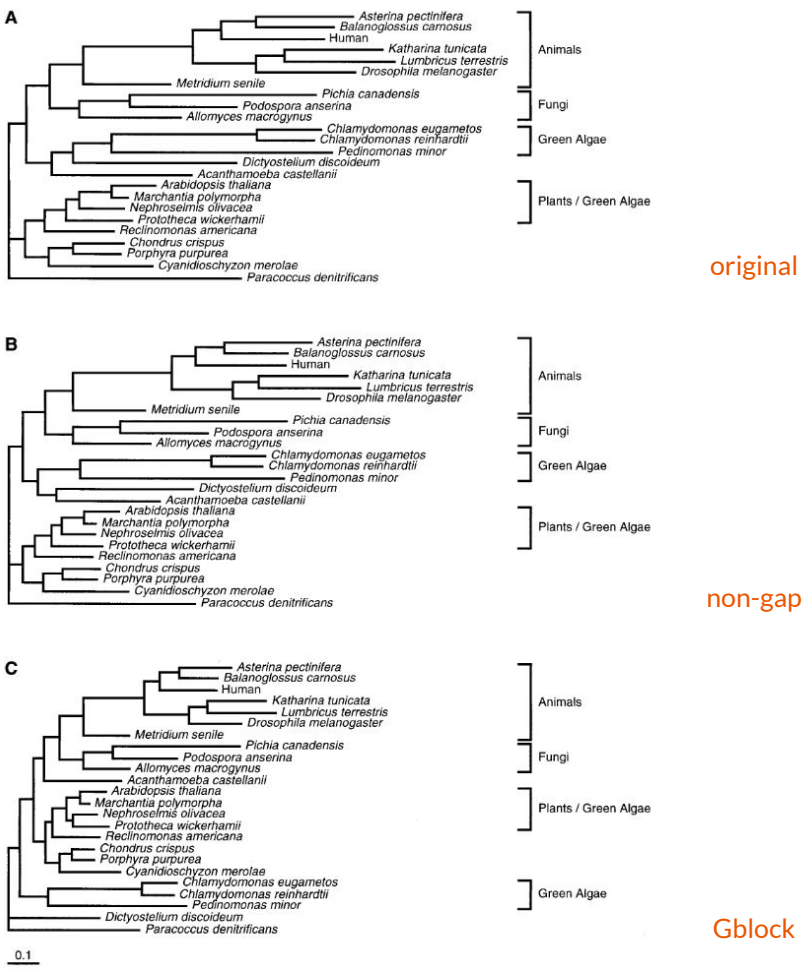
Same topology but shorter length

on the **first** dataset



Input alignment does matter

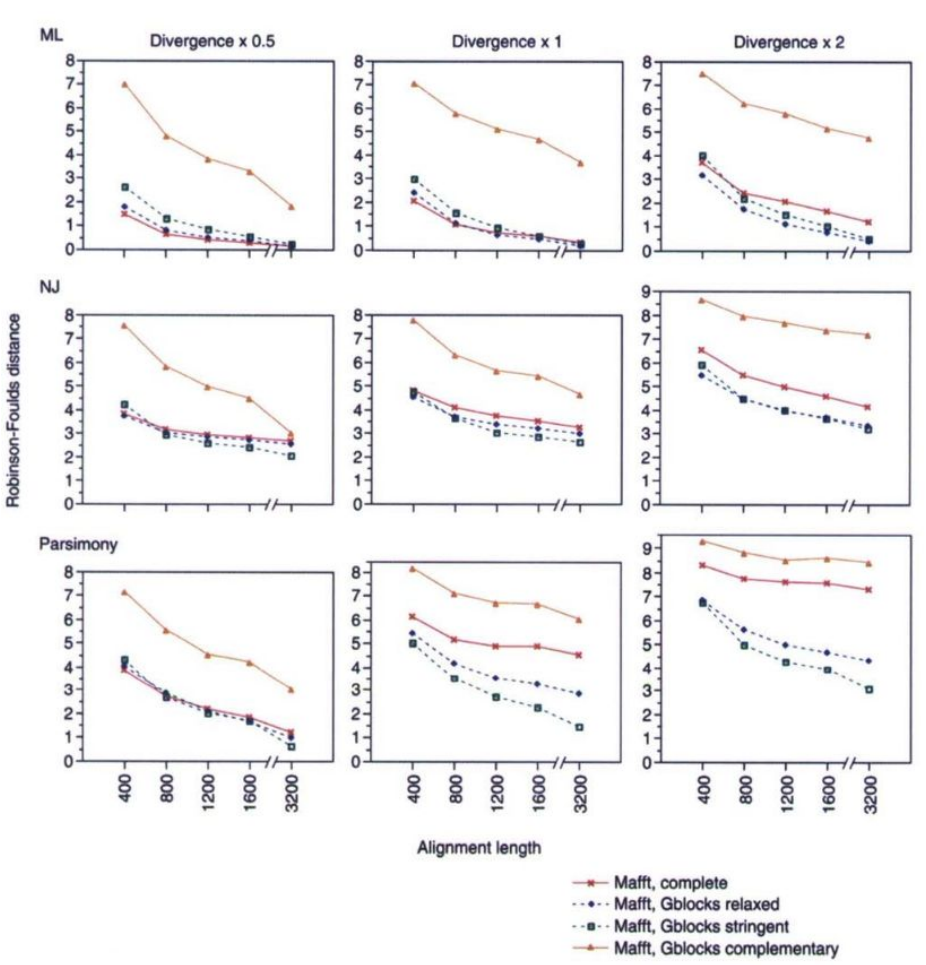
on the **second** dataset, topology is different for the additional species



Result

The performance in a simulated study

Improvement of Phylogenies after Removing Divergent and Ambiguously Aligned Blocks from Protein Sequence Alignments
by Gerard Talavera and **Jose Castresana**



https://doi.org/10.1080/10635150701472164

Evolution rate is different

- different remove %
- smaller outgroup distance

Table 1
Positions Removed by Gblocks with Default Parameters, and Reduction in the Average Pairwise Distance from the Outgroup to Other Sequences in Different Mitochondrial Protein Alignments

Protein	No. of Positions			Average Outgroup Distance	
	Original Alignment	Gblocks Alignment	% Removed	Original Alignment	Gblocks Alignment
CO1	586	447	23.7	0.817	0.663
CO2	375	184	50.9	1.639	1.225
CO3	339	221	34.8	1.169	0.950
CYTb	445	340	23.6	1.083	0.930
ND1	392	253	35.5	1.430	1.061
ND2	615	57	90.7	ND	ND
ND3	153	56	63.4	1.401	0.586
ND4	613	257	58.1	1.751	1.143
ND4L	108	61	43.5	1.771	1.453
ND5	827	291	64.8	1.867	0.969
All concatenated	4,453	2,167	51.3	1.438	0.919

Note.—ND = not determined (some pairwise distances were too large).

Robustness towards different *parameters*

- similar % and outgroup distance
- each parameter response to a similar amount of removal

Table 2
Effect of Different Parameters of the Gblocks Program on the Final Alignment

Type of Alignment	No. of Positions	% Removed	Average Outgroup Distance
Original	4,453		1.438
Ungapped	2,895	35.0	1.232
Gblocks (default)	2,167	51.3	0.919
Gblocks (CP = 12)	2,178	51.1	0.923
Gblocks (CP = 4)	1,926	56.7	0.832
Gblocks (IS = 11)	1,969	55.8	0.851
Gblocks (IS = 13)[a]	1,849	58.5	0.797
Gblocks (FS = 12)	2,271	49.0	0.946
Gblocks (FS = 16)	1,972	55.7	0.876
Gblocks (BL1 = 20)	2,135	52.1	0.923
Gblocks (BL2 = 0)	2,210	50.4	0.914

Robustness towards different *alignments*

Table 3
Effects of Different CLUSTAL W Alignment Parameters on the Final Blocks Selected by Gblocks

CLUSTAL W Parameters	No. of Positions			Average Out-group Dis-tance in Gblocks Align-ment
	Origin-al Align-ment	Gblocks Align-ment	% Re-moved	
GOP = 10, GEP = 0.05 (default) . . .	4,453	2,167	51.3	0.919
GOP = 10, GEP = 0.5	4,384	2,141	51.2	0.932
GOP = 5, GEP = 0.05	4,501	2,107	53.2	0.878
GOP = 5, GEP = 0.5	4,401	2,122	52.4	0.918
GOP = 20, GEP = 0.05	4,459	2,174	50.2	0.931
GOP = 20, GEP = 0.5	4,365	2,118	51.9	0.923

Note.—GOP = gap opening penalty; GEP = gap extension penalty.

More Ambiguity

- removing saturated and poorly aligned regions should get us better resolution. **NO!**
- this is not due to lower # of positions
- one possibility: guided tree used by the alignment can created biased divergent area help rejecting more trees

Table 4
Properties of Maximum-Likelihood Trees Derived from Different Alignments

Type of Alignment	No. of Positions	ln *L* of Best Tree[a]	Number of Similar Trees[a]	Bootstrap Proportion[a]
Original	4,453	−101,171.4	9	41.02
Ungapped	2,895	−73,772.9	8	57.34
Gblocks (default) . . .	2,167	−46,470.6	24	26.75

[a] Values calculated from the 945 possible tree topologies relating seven clades, as explained in the text.

Advantage

- Eliminate non homologous positions
- Distribution is more homogeneous and more suitable for modeling
- Remove human bias and better reproducibility

Disadvantage

- Less support for final tree
  - partially unresolved tree > biased tree
- Does not handle misalignment of sequence
  - there are methods that could do this

Thanks!