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
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
- 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
- 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
- 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

Second Dataset

5 protein subunits of 23 + 1 species
Alignment Method and Tree Estimation

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

Adachi J, Hasegawa M.

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
Methods

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

```plaintext
1234567890123456789012345678901234567890
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
---BBBAAAAAAAAAAAAAAAAAAB-AAABBABBBBABBAB--
---CBBAAAAAFFFFFFFFFFFFACAAAACCCCCCccccac--
---NCCCHHHHHHHHHHHHHHHHHNN-HHNNNNNNNNNNNNN
```
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 $\geq$ BL1
   a. BL1 = 15
5. Remove column with gaps + adjacent non-conserved columns
6. Similar to 4, now takes the blocks with length $\geq$ BL2
   a. BL2 = 10
Methods

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
Methods

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
Methods

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 \(\geq BL1\)
   a. BL1 = 15

5. Remove column with gaps + adjacent non-conserved columns

6. Similar to 4, now takes the blocks with length \(\geq BL2\)
   a. BL2 = 10
Methods

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 for Amino Acid (1x17 vector) and Species (1x20 vector).
Tree branch length is more similar to pairwise distance
Same topology but shorter length

on the first dataset
Input alignment does matter

on the second dataset, topology is different for the additional species
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>
<thead>
<tr>
<th>PROTEIN</th>
<th>NO. OF POSITIONS</th>
<th>AVERAGE OUTGROUP DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original Alignment</td>
<td>Gblocks Alignment</td>
</tr>
<tr>
<td>CO1</td>
<td>586</td>
<td>447</td>
</tr>
<tr>
<td>CO2</td>
<td>375</td>
<td>184</td>
</tr>
<tr>
<td>CO3</td>
<td>339</td>
<td>221</td>
</tr>
<tr>
<td>CYTb</td>
<td>445</td>
<td>340</td>
</tr>
<tr>
<td>ND1</td>
<td>392</td>
<td>253</td>
</tr>
<tr>
<td>ND2</td>
<td>615</td>
<td>57</td>
</tr>
<tr>
<td>ND3</td>
<td>153</td>
<td>56</td>
</tr>
<tr>
<td>ND4</td>
<td>613</td>
<td>257</td>
</tr>
<tr>
<td>ND4L</td>
<td>108</td>
<td>61</td>
</tr>
<tr>
<td>ND5</td>
<td>827</td>
<td>291</td>
</tr>
<tr>
<td>All concatenated</td>
<td>4,453</td>
<td>2,167</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>CLUSTAL W PARAMETERS</th>
<th>No. of Positions</th>
<th>Average Out-group Distance in Gblocks Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original Alignment</td>
<td>Gblocks Alignment</td>
</tr>
<tr>
<td>GOP = 10, GEP = 0.05 (default)</td>
<td>4,453</td>
<td>2,167</td>
</tr>
<tr>
<td>GOP = 10, GEP = 0.5</td>
<td>4,384</td>
<td>2,141</td>
</tr>
<tr>
<td>GOP = 5, GEP = 0.05</td>
<td>4,501</td>
<td>2,107</td>
</tr>
<tr>
<td>GOP = 5, GEP = 0.5</td>
<td>4,401</td>
<td>2,122</td>
</tr>
<tr>
<td>GOP = 20, GEP = 0.05</td>
<td>4,459</td>
<td>2,174</td>
</tr>
<tr>
<td>GOP = 20, GEP = 0.5</td>
<td>4,365</td>
<td>2,118</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Type of Alignment</th>
<th>No. of Positions</th>
<th>$\ln L$ of Best Tree$^a$</th>
<th>Number of Similar Trees$^a$</th>
<th>Bootstrap Proportion$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>4,453</td>
<td>$-101,171.4$</td>
<td>9</td>
<td>41.02</td>
</tr>
<tr>
<td>Ungapped</td>
<td>2,895</td>
<td>$-73,772.9$</td>
<td>8</td>
<td>57.34</td>
</tr>
<tr>
<td>Gblocks (default)</td>
<td>2,167</td>
<td>$-46,470.6$</td>
<td>24</td>
<td>26.75</td>
</tr>
</tbody>
</table>

$^a$ Values calculated from the 945 possible tree topologies relating seven clades, as explained in the text.
Discussion

**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!