Multiple Sequence Alignment

From:


and


Multiple Sequence Alignment

Shows multiple similarities:

- Common structure of protein product
- Common function
- Common evolutionary process

- Family of proteins => more information than single sequences
- Characterization: signatures of protein families
**2nd Fact of Biological Sequence Analysis [4]:**

Evolutionary and functionally related molecular strings can **differ significantly** throughout much of the string and yet preserve:

1. the same three dimensional structures
2. the same two-dimensional substructures (motifs, domains)
3. the same active sites
4. the same or related dispersed residues (DNA or amino acid)

**Evolutionary Conservation**

**Evolutionary preserved features:**
- 3-dimensional structures (well preserved)
- 2-dimensional substructures (well preserved)
- active sites: functions (less common)
- Amino acid sequence (least common)

**Three biological uses:**
- Representation of protein families
- Identification of conserved sequence features correlating with structure and function
- Deduction of evolutionary history
An Amazing Example: Hemoglobin

Hemoglobin:
- An almost universal protein found in birds, mammals, etc.
- 4 chains of ~140 amino acids
- Functions the same in all birds, mammals, etc.: binds and transports oxygen
- Insects and mammals diverged ~600 million years ago
  => On average 100 amino acids mutations per chain
- Pair wise alignment:
  - human - chimpanzee equal
  - Mammal - mammal suggests functional similarity
  - Insect-mammal very little similarity!
- Secondary and 3-dimensional structure well preserved
Multiple Alignment Examples

**Related sequences** can have so few conserved or so dispersed matching amino acids: *statistically indistinguishable from the best alignment of 2 random strings.*

For example this is true for:
- Hemoglobin, immunoglobulin (antibody proteins)
- E-Cadherin (adhesion molecule)

Compare to:
1 DNA nucleotide change:
- Sickle cell anemia
**Global) Multiple Alignment**

**Definition**

A *multiple alignment* of a set of strings \( \{S_1, S_2, \ldots, S_k\} \) is a series of strings \( S'_1, S'_2, \ldots, S'_k \) such that

1. \( |S'_1| = |S'_2| = \ldots = |S'_k| \) (all \( S'_i \) have the same length)
2. For every \( i \): \( S'_i \) is an extension of \( S_i \) obtained by insertion of spaces.

A multiple alignment of strings ACBCBD, CADDB, and ACABCD:

Note: similarly, *local multiple alignment* (for substrings) can be defined.

**Multiple Alignment**

**Definition Induced Pair-Wise Alignment:**

Given a multiple alignment \( M \), the *induced pair-wise alignment* of two string \( S_i \) and \( S_j \) is obtained from \( M \) by removing all rows except rows \( i \) and \( j \).

Note: two opposing spaces can be removed

**Definition Score of Induced Pair-Wise Alignment:**

The score of an induced pair-wise alignment is determined using any chosen *scoring scheme* for 2-string alignment in the standard manner.
Multiple Alignment: Sum-of-Pairs

Definition: **Sum of Pairs Score**

The sum-of-pairs (SP) score of a multiple alignment $M$ is the sum of the scores of pair-wise global alignments induced by $M$.

Problem: **The SP Alignment Problem**

Compute a global multiple alignment $M$ with minimum sum-of-pairs score.

**Note:** Intuitively this is a reasonable score, but: **no theoretical justification**!

---

Common Alignment Methods

- Aligning a String to a Profile
- Iterative Pair-wise Alignment
- Progressive Multiple Alignment
  - Feng-Doolittle (1987)[2]
  - CLUSTALW, CLUSTALX
- PAGAN Phylogeny Aware MSA
- Etc.
Helicases
- A protein to unwind DNA for further read for duplication, transcription, recombination or repair.
- Werner’s syndrome an aging disease is believed to be due to a gene **WRN** that codes for a helicase protein.

A Signature Profile for Helicases
- Conserved sequence signatures or motifs
- Some of these motifs are unique identifiers for helicases
- Maybe functional units

Multiple Alignment Profile
- Character frequencies given per column
- $p_i(a)$ is the fraction of $a$’s in column $i$
- $p(a)$ is the fraction of $a$’s overall
- Log likelihood ratio $\log( p_i(a)/p(a) )$ can be used.
Aligning a String to a Profile

**Definition 4.16** For an alignment \( S \) of length \( l \), a **profile** is an \( l \times |\Sigma| + 1 \) matrix, whose columns are probability vectors denoting the frequencies of each symbol in the corresponding alignment column.

<table>
<thead>
<tr>
<th>Code</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>-5</td>
<td>1</td>
<td>-10</td>
<td>-8</td>
<td>-5</td>
<td>-6</td>
<td>-9</td>
<td>0</td>
<td>-14</td>
<td>-7</td>
<td>-3</td>
<td>-1</td>
<td>0</td>
<td>-5</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>-9</td>
<td>1</td>
<td>-10</td>
<td>-2</td>
<td>-4</td>
<td>-9</td>
<td>0</td>
<td>-10</td>
<td>-6</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>0</td>
<td>-15</td>
<td>3</td>
<td>1</td>
<td>-16</td>
<td>-10</td>
<td>-5</td>
<td>-14</td>
<td>0</td>
<td>-14</td>
<td>-9</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>G</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>-5</td>
<td>1</td>
<td>-6</td>
<td>-1</td>
<td>-3</td>
<td>-9</td>
<td>0</td>
<td>-4</td>
<td>-9</td>
<td>-4</td>
<td>-1</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>S</td>
<td>-5</td>
<td>7</td>
<td>-26</td>
<td>13</td>
<td>24</td>
<td>-24</td>
<td>-15</td>
<td>-4</td>
<td>-25</td>
<td>0</td>
<td>7</td>
<td>-24</td>
<td>-15</td>
<td>0</td>
<td>0</td>
<td>-4</td>
<td>9</td>
<td>-5</td>
</tr>
<tr>
<td>B</td>
<td>-6</td>
<td>3</td>
<td>-24</td>
<td>7</td>
<td>22</td>
<td>-20</td>
<td>-15</td>
<td>-1</td>
<td>-24</td>
<td>0</td>
<td>3</td>
<td>-22</td>
<td>-15</td>
<td>0</td>
<td>0</td>
<td>-10</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>Y</td>
<td>-5</td>
<td>-6</td>
<td>17</td>
<td>-2</td>
<td>0</td>
<td>-8</td>
<td>-15</td>
<td>-3</td>
<td>-6</td>
<td>0</td>
<td>-5</td>
<td>-10</td>
<td>-6</td>
<td>-9</td>
<td>0</td>
<td>-12</td>
<td>-11</td>
<td>-11</td>
</tr>
<tr>
<td>V</td>
<td>-10</td>
<td>-10</td>
<td>-17</td>
<td>-17</td>
<td>10</td>
<td>-20</td>
<td>0</td>
<td>-20</td>
<td>-12</td>
<td>0</td>
<td>-20</td>
<td>-6</td>
<td>-12</td>
<td>0</td>
<td>-20</td>
<td>-9</td>
<td>-12</td>
<td>12</td>
</tr>
<tr>
<td>U</td>
<td>-8</td>
<td>-12</td>
<td>-20</td>
<td>-21</td>
<td>-27</td>
<td>-27</td>
<td>0</td>
<td>-20</td>
<td>9</td>
<td>6</td>
<td>-29</td>
<td>0</td>
<td>-15</td>
<td>-20</td>
<td>0</td>
<td>0</td>
<td>-13</td>
<td>10</td>
</tr>
<tr>
<td>K</td>
<td>16</td>
<td>16</td>
<td>52</td>
<td>20</td>
<td>98</td>
<td>-43</td>
<td>-28</td>
<td>0</td>
<td>89</td>
<td>9</td>
<td>19</td>
<td>-40</td>
<td>-30</td>
<td>0</td>
<td>0</td>
<td>-13</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>-2</td>
<td>0</td>
<td>-14</td>
<td>-9</td>
<td>1</td>
<td>-20</td>
<td>-12</td>
<td>-5</td>
<td>-21</td>
<td>0</td>
<td>19</td>
<td>-15</td>
<td>-8</td>
<td>-2</td>
<td>0</td>
<td>-15</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Y</td>
<td>-8</td>
<td>-16</td>
<td>-17</td>
<td>-12</td>
<td>-3</td>
<td>-21</td>
<td>-3</td>
<td>-1</td>
<td>-31</td>
<td>0</td>
<td>-10</td>
<td>21</td>
<td>-10</td>
<td>-11</td>
<td>0</td>
<td>-28</td>
<td>-10</td>
<td>-7</td>
</tr>
<tr>
<td>D</td>
<td>-8</td>
<td>-8</td>
<td>-6</td>
<td>-15</td>
<td>-10</td>
<td>-3</td>
<td>-9</td>
<td>-30</td>
<td>17</td>
<td>0</td>
<td>-16</td>
<td>21</td>
<td>-11</td>
<td>-21</td>
<td>0</td>
<td>-21</td>
<td>-15</td>
<td>-15</td>
</tr>
<tr>
<td>G</td>
<td>-5</td>
<td>9</td>
<td>13</td>
<td>-13</td>
<td>8</td>
<td>-22</td>
<td>-8</td>
<td>-9</td>
<td>-28</td>
<td>0</td>
<td>-4</td>
<td>-25</td>
<td>-17</td>
<td>0</td>
<td>0</td>
<td>-5</td>
<td>0</td>
<td>-11</td>
</tr>
<tr>
<td>S</td>
<td>-8</td>
<td>-8</td>
<td>-18</td>
<td>-13</td>
<td>0</td>
<td>-17</td>
<td>-16</td>
<td>-4</td>
<td>-18</td>
<td>0</td>
<td>13</td>
<td>-12</td>
<td>-5</td>
<td>-3</td>
<td>0</td>
<td>-13</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

**Average Profile Score (total) = -6**

**Average Profile Score (no gaps) = -6**

### Iterative Pair-wise Alignment

**Algorithm**

1. Align some pair
2. While (not done)
   (a) Pick an unaligned string which is ”near” some aligned one(s).
   (b) Align with the **profile** of the previously aligned group. Resulting new spaces are inserted into all strings in the group.

This approach uses pair-wise alignment scores to iteratively add one additional string to a growing multiple alignment.

1. We start by aligning the two strings whose edit distance is the minimum over all pairs of strings.
2. Then we iteratively consider the string with the smallest distance to any of the strings already in the multiple alignment.
Progressive Multiple Alignment

- Again a heuristic => not guaranteed to be optimal
- Progressive alignment of the sequences

Problems
- What are the initial sequences?
- What is the order in which the sequences are aligned?

Sketch
- Align all pairs of sequences
- Determine distance matrix
- Construct a guide tree from the distance matrix
- Progressive multiple alignment following the guide tree.
Progressive Alignment

- Feng-Doolittle (1987)
- CLUSTALW

Different Guide Tree Construction Methods:
- SB - Sequential Branching
- UPGMA - Unweighted Pair Grouping Method
- ML - Maximum Likelihood
- NJ - Neighbor-Joining

From: O.Poch, Ecole Phylogénomique, Carry le Rouet 2006
Multiple Sequence Alignment: HMM known

Multiple Sequence Alignment Problem:
Given sequence $S_1, \ldots, S_n$, how can they be optimally aligned?

Assume a profile HMM $P$ is known, then:
- Align each sequence $S_i$ to the profile separately
- Accumulate the obtained alignments to a multiple alignment
- Hereby insert runs are not aligned. (Just left-justify insert regions.)

Multiple Sequence Alignment: HMM unknown

Multiple Sequence Alignment Problem:
Given sequence $S_1, \ldots, S_n$, how can they be optimally aligned?

Assume a profile HMM $P$ is not known, then obtain an HMM profile $P$ from $S_1, \ldots, S_n$ as follows:
- Choose a length $L$ for the profile HMM and initialize the transition and emission probabilities.
- Train the HMM using Baum-Welch on all sequences $S_1, \ldots, S_n$.

Now obtain the multiple alignment using this HMM $P$ as in the previous case:
- Align each sequence $S_i$ to the profile separately
- Accumulate the obtained alignments to a multiple alignment
- Hereby insert runs are not aligned. (Just left-justify insert regions.)
Characteristics of some of the best MSA programs

- ClustalW
  - Less memory
  - Less accurate, less scalable

- DIALIGN
  - Distinction of alignable vs non-alignable
  - Less accurate than ClustalW on some benchmarks

- MAFFT, MUSCLE
  - Faster, more accurate than CLUSTALW, trade-off accuracy/speed (in case of >1000 sequences)

- T-COFFEE
  - High accuracy, uses heterogenous information
  - Computational and space intensive (limiting factor for >100 sequences)

MSA Benchmarks

A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives

J.D. Thompson, B. Linard, O. Lecompte, O. Poch
Typical problems when aligning large sets of protein sequences addressed by benchmarks:

- Locally conserved regions: functional specificities; modulation of proteins function in context of cell
- Motifs in natively disordered regions (often misaligned)
- Badly predicted proteins or fragmentary protein sequences (more and more a problem in today’s databases)

Selected slides from Mark Ragan’s presentation:

Phylogenetics without multiple sequence alignment

Mark Ragan
Institute for Molecular Bioscience
and
School of Information Technology & Electrical Engineering
The University of Queensland, Brisbane, Australia

IPAM Workshop on Multiple Sequence Alignment
UCLA, 13 January 2015

See also: [http://www.ipam.ucla.edu/programs/workshops/multiple-sequence-alignment/?tab=schedule](http://www.ipam.ucla.edu/programs/workshops/multiple-sequence-alignment/?tab=schedule)
An MSA* gives us (visual) access to...

- Patterns within columns
- Local adjacency relationships within rows (across columns): context
- Global architecture

* MSA = multiple sequence alignment

Here, we focus on MSA as input into a tree-inference program

For this application to phylogeny, we interpret the MSA as a
position-by-position (i.e. column-by-column) hypothesis of homology
among these sequences
Tree inference from MSA: a few comments

- The sequences must be *homologous* for tree inference to make sense
- Trees and matrices are related complementary *data structures*
- Trees show inferred relationships; MSAs show conserved regions

**Homology signal**

We use the **homology signal** inherent in the sequences to make an inference about treelike relationships

*Homology signal inheres in the sequences*, not in their MSA

MSA can make it easier to see*, but doesn’t create it

* and easier for existing computer programs to work with
Homology signal (continued)

We shouldn’t assume that MSA captures it all, or uses it optimally

MSA gives us access to

- Patterns within columns
- Local adjacency relationships
- Global architecture

Let’s consider these to be components of the homology signal

Here we’ll focus on the first two of these components

Pattern and adjacency

The column component needs to capture “sameness” of a character across sequences

For application in phylogenetics, “sameness” has to mean homology (or orthology). It’s difficult to build a statistical case that a particular single character in one sequence is homologous with a particular one in a second sequence.

MSA uses adjacency (and sometimes global) information to build this support. Alternatively we might compare sets of adjacent characters (strings), which are less likely to occur by chance.

The adjacency component doesn’t just provide statistical support for the column component

Because conserved function arises in part from chemical properties of adjacent residues (e.g. in making that part of the molecule an active site or α-helix), we expect homology signal to have an adjacency component in its own right.
MSA: potential (and real) problems

*Genomes are dynamic, data can be dirty, and MSA is hard*

Within some but not all members of a gene set...
- Homologous regions may be inserted / deleted
- Homologous regions may be rearranged / duplicated
- Regions may have different evolutionary histories (LGT)
- Transcriptional variation \(\rightarrow\) similar issues for protein sets

Sequences may be **mis-assembled** (or not assembled in the first place) and/or truncated

MSA is computationally difficult and/or heuristic

---

Can we extract enough/most/all of the homology signal without MSA?

---

**Molecular phylogenetics before sequences**

Oligonucleotide catalogs as k-mer spectra

Mark A. Raught, Guillaume Bernard, and Cheong Xin-Chan

*Institute for Molecular Bioscience, and ARC Centre of Excellence in Bioinformatics, The University of Queensland, Brisbane, QLD, Australia*

---

Ribosomal RNA (1964)

*Electrophoresis of Ribosomal RNA segments out of a lot of different species.*
Oligonucleotide catalogs

Similarity coefficient:

\[ S_{AB} = \frac{2 N_{AB}}{(N_A + N_B)} \]

where

- \( N_A \) = number of oligomers of at least length \( L \) for RNA of organism A, and
- \( N_{AB} \) = total number of coincident oligomers between catalogs A and B

See (Fox et al. USB 1977) for a detailed definition.


Fox et al., *PNAS* 1977 (top)

The three kingdoms (domains) of life

Phylogenetic structure of the prokaryotic domain: The primary kingdoms

From Wikimedia Commons after Carl Woese and colleagues (~1972 ff.)

Image courtesy of Institute for Genomic Biology University of Illinois
Alignment-free methods

$k$-mers / $k$-tuples / $k$-words / $n$-mers / $n$-grams

For a sequence of length $S$, there are $S - k + 1$ $k$-mers, not all of which are necessarily unique

There's also a parallel world of patterns

**D<sub>2</sub> statistics: a brief overview**

*The D<sub>2</sub> statistic is the count of exact word matches of length k between two sequences*

For alphabet A, there are |A|^k possible words w of length k. Given sequences X and Y,

\[ D_2 = \sum_{w \in A^k} X_w Y_w \]

Because D<sub>2</sub> is sensitive to sequence length, the statistic is often normalised by the probability of occurrence of specific words (D<sub>2</sub>^S), or by assuming a Poisson distribution of word occurrence (D<sub>2</sub>^P) for long words.

Although defined for exact word matches, D<sub>2</sub> can be easily extended to n mismatches (neighbourhood of order n): D<sub>2</sub>^n

Chor et al., Genome Biol 10:R108 (2009); Reinert et al., J Comput Biol 16:1615-1634 (2009);

---

**D<sub>2</sub>-based distance**

\[ D_2 = \sum_{w \in A^k} X_w Y_w \]

(or D<sub>2</sub>^S, D<sub>2</sub>^P etc.)

Compute pairwise distances
Generate distance matrix
Tree via N-J or similar
### Other Alignment Free methods based on word counts

<table>
<thead>
<tr>
<th>Method</th>
<th>Details</th>
</tr>
</thead>
</table>
| **Feature frequency profile**  | Sims & Kim, PNAS 2011
Comparates $k$-mer frequency profiles (Jensen-Shannon divergence) & computes a pairwise distance |
| **Composition vector**         | Wang & Hao, JME 2004
using word frequencies normalised by probability of chance of occurrence |
| **Word context**               | Co-phylog: Yi & Jin, NAR 2013
Pairwise distances based on proportions of $k$-mers that differ in a certain position; more-realistic branch lengths |
| **Spaced word frequencies**    | Leimeister, Bioinformatics 2014
Considers word mismatches as well as matches; less statistical dependency between neighbouring matches |

### Alignment Free (AF) methods based on match length

*In general, similar sequences share longer exact words*

<table>
<thead>
<tr>
<th>Method</th>
<th>Details</th>
</tr>
</thead>
</table>
| **Grammar-based distance**     | Russell, BMC Bioinf 2010
The concatenate of two sequences is more compressible (e.g. by Lempel-Ziv) if the sequences are similar |
| **Average common substring**   | Ulitsky, J Comp Biol 2006
Mean of longest matches between sequences, starting from each position; unlike L-Z, word overlap is allowed |
| **Shortest unique substring**  | Haubold, J Comp Biol 2009
Longest common substring + 1, corrected for random matches: “AF version of Jukes-Cantor distance” |
| **Underlying subwords**        | Comin, Algorith Mol Biol 2012
Like ACS, but discards common subwords that are covered by longer (more-significant) ones |
| **$k$-Mismatch ACS (kmacs)**    | Leimeister, Bioinformatics 2014
ACS with $k$ (in our notation, $n$) mismatches |
Shortest unique substring (shustring) algorithm

Unique substrings remain unique upon extension, so use only the shortest ones

The length of the shortest unique substring is inversely related to information content of the sequence

- *C. elegans* autosomes $L=11$ (one example of 10)
- human autosomes $L=11$, but Y chromosome $L=12$
- mouse autosomes $L=11$, but Y chromosome $L=12$

Given a random sequence model, the probability of finding even one shustring of $L=11$ in human is $<10^{-100}$

In human and mouse (and presumably other) genomes, shustrings are preferentially located within 1 kb of protein-coding genes

Haubold et al., J Comp Biol 2009

---

Shustring Example

Consider $S = \text{ACCG}$

Has a total of 10 substrings of which 8 are unique:
{A,AC, ACC, ACCG, CC, CCG, CG,G }

2 shortest globally unique substrings:
{A, G}
Can occur everywhere in $S$.

Shortest locally unique substrings:
For every position $i$ in $S$ the length $x$ of the substring $S[i...i+x-1]$ such that it is unique while $S[i...i+x-2]$ is not.
$X=1$ for $i=1$, $x=2$ for $i=2$ $x=2$ for $i=3$, $x=1$ for $i=4$
Under simplifying assumptions, there’s a relationship between 
\( d \) (≈ mutational distance between two sequences) and average shustring length 
Haubold et al., J Comp Biol 2009

The probability that a shustring of length \( X \) is longer than a threshold \( t \) is given by

\[
Pr \{ X > t \} = (1 - m/l)^l \approx e^{-tm/lt}
\]

where \( m= \) number of mutations and \( l= \) length of sequence

If all nucleotides are equally frequent, the correction for random matches is

\[
Pr \{ X \leq t \} = (1 - e^{-tm/lt})(1 - 4^{-l})^l
\]

Correction for multiple substitutions yields an AF version of classical (Juges-Cantor) distance

\[
d_{kr} = \frac{3}{4} \ln \left( 1 - \frac{4}{3l} m \right)
\]

---

**Can we compute accurate trees using AF-based distances?**

*How do we best ask this question?*

**Simulated data**

- Generate replicate data on a known tree, varying data size, substitution model, tree shape, branch lengths etc.
- Extract \( k \)-mers & compute a tree; sweep over relevant parameters
- Compare topologies (Robinson-Foulds metric)
- Measure performance (precision, recall, sensitivity...)

**Advantages/disadvantages**

- We can study effects of different factors & scenarios individually
- Sequence models may be too simplistic

**Empirical data**

- Identify empirical datasets for which someone has ventured a phylogenetic tree
- Extract \( k \)-mers & compute a tree; sweep over \( k \)
- Compare topologies (Robinson-Foulds metric)
- Count congruent/incongruent edges & try to interpret

**Advantages/disadvantages**

- Sequences are (by definition) real
- We can’t study effects of different factors & scenarios individually
- The true tree remains unknown
First we simulated sequence data on a tree

Simulation software ranges from simplistic to maddeningly complex.

Using *evolver* (PAML) we simulated DNA and protein sequence sets on trees of different size (8 / 32 / 128 taxa), symmetry, and absolute and relative branch lengths.

We also simulated DNA sequences on trees generated under a coalescent model (not shown).

We extracted \(k\)-mers at different \(k\), computed distances under different variants of the \(D_2\) statistic, and generated a N-J tree.

No method for confidence estimation is currently available, but one can imagine using a variant of the nonparametric bootstrap, or by jackknifing.
Then we compared the $D_2 + NJ$ tree with the known true topology, and with the topologies inferred using MSA + MrBayes.

DNA alphabet, $L = 1500$ nt, 100 replicates

Chan et al., *Scientific Reports* 2014

$D_2 + NJ$ performs well with rearranged sequences

Non-overlapping rearrangement of R% of a DNA sequence set, $N=8$, $L=5000$. MSA = MUSCLE + MrBayes. MAFFT performs slightly worse than MUSCLE.

Chan et al., *Scientific Reports* 2014
**Summary: trees computed from k-mer distances**

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Accuracy of $D_2$ methods increases with L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence length</td>
<td>$D_2$</td>
</tr>
<tr>
<td>Recent sequence divergence</td>
<td>MSA</td>
</tr>
<tr>
<td>Ancient sequence divergence</td>
<td>$D_2$</td>
</tr>
<tr>
<td>Among-site rate heterogeneity</td>
<td>$D_2$ or MSA</td>
</tr>
<tr>
<td>Compositional bias</td>
<td>$D_2$ or MSA</td>
</tr>
<tr>
<td>Genetic rearrangement</td>
<td>$D_2$</td>
</tr>
<tr>
<td>Incomplete sequence data</td>
<td>MSA</td>
</tr>
<tr>
<td>Insertions/deletions</td>
<td>$D_2$</td>
</tr>
<tr>
<td>Computational scalability</td>
<td>$D_2$</td>
</tr>
<tr>
<td>Memory consumption</td>
<td>MSA</td>
</tr>
</tbody>
</table>

$D_2$ methods are more robust to ancient sequence divergence, to rearrangement and to indel frequency.

$D_2$ methods are more sensitive to recent sequence divergence and to the presence of incomplete (truncated) data.

Optimal $k$ is negatively correlated with alphabet size, and is not greatly affected by $N$ or $L$ in a biologically relevant range.

$D_2$ methods are more scalable to large data than are MSA-based approaches, but usually require more memory.

**With data simulated under a coalescent model, $D_2^{nj} + NJ$ results are similar to MSA except at high/low sequence divergence**

**$D_2^S + NJ$ is more-robust to indels than leading MSA methods**

Numbers in box are $N_e = \text{effective population size}$

Smaller $N_e$ implies shorter branch lengths on the tree

Chan et al., *Scientific Reports* 2014
Eight Yersinia genomes: AF versus inversion phylogeny

Consensus phylogenetic network based on inversions. Mauve (78 locally collinear blocks) then BADGER (Larget, *MBE* 2005). Requires extensive parameter estimation, with each run 500K MCMC generations. Took ~ 2 weeks.


Kr (Haubold, *BMC Bioinformatics* 2005) yields a congruent phylogeny; no parameter optimisation, runtime 1 minute on laptop.

Bernard, Chan & Ragan, unpublished

27 Escherichia coli + Shigella genomes

ProgressiveMauve alignment (17 hours on HPC), extract 5282 single-copy gene sets \(n \geq 4\), GBlocks, MrBayes (5M MCMC generations, 10 models) followed by MRP


Co-phylog (Yi & Jin, *NAR* 2013) with \(k=8\), 113sec on laptop

Bernard, Chan & Ragan, unpublished
Conclusions & outlook

AF methods hold considerable potential in phylogenetics & phylogenomics

But MSA-based approaches have a six-decade head start

With synthetic data, AF methods perform better than MSA-based approaches under some evolutionarily relevant scenarios, but worse under others

With empirical data, the jury is still out

(Some) AF methods could likely be subsumed under a rigorous model, although probably at the cost of speed & scalability

i.e. what makes them attractive in the first place

Efficient data structures & precomputation have much to offer

Other application areas include LGT analysis, and trees directly from NGS data

Song et al., J Comp Biol 2013; Yi & Jin, NAR 2013

Bibliography

Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm.

A non-independent energy-based multiple sequence alignment improves prediction of transcription factor binding sites.
Rafik A. Salama1 and Dov J. Stekel, Bioinformatics, Vol. 29 no. 21 2013, pages 2699–2704.

Protein Multiple Sequence Alignment,