Basics of Multiple Sequence Alignment

Tandy Warnow

February 10, 2018

Basics of Multiple Sequence Alignment

Tandy Warnow

Basic issues

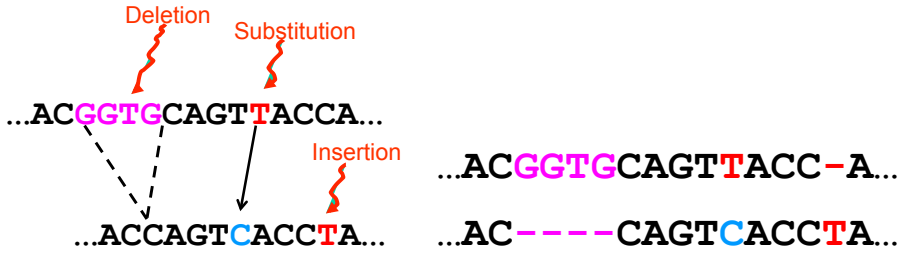
- What is a multiple sequence alignment?
- Evolutionary processes operating on sequences
- Using profile HMMs to model multiple sequence alignment
- Optimization problems
- Basic techniques of standard methods
- Fundamental limitations of nearly all multiple sequence alignment methods
- How to evaluate alignments
- Performance studies of multiple sequence alignment methods operate on data

A multiple sequence alignment

s_1	-	-	-	T	A	C
s_2	-	-	A	T	A	C
s_3	C	-	A	-	-	G
s_4	C	-	A	A	T	G
s_5	C	-	-	T	-	G
s_6	C	T	-	-	A	C
s_7	C	-	A	T	A	C
s_8	G	-	A	-	A	T

Homology

Two letters in two sequences are *homologous* if they descend from a letter in a common ancestor.



The true multiple alignment

- **Reflects historical substitution, insertion, and deletion events**
- **Defined using transitive closure of pairwise alignments computed on edges of the true tree**

Evolutionary multiple sequence alignment seeks to create a matrix in which the input sequences are the rows and each column has letters that are all homologous to each other.

Evolutionary processes operating on sequences

- Substitutions
- Insertions and deletions of strings (indels)
- Rearrangements (inversions and transpositions)
- Duplications of regions

Note: most alignment methods stretch out sequences so that the line up well, and so only address substitutions and indels.

Building a profile HMM for an MSA

For the MSA below, the standard technique for building profile HMMs would use an insertion state for position 2 (because more than 50% of the sequences are gapped in that position).

s_1	-	-	-	T	A	C
s_2	-	-	A	T	A	C
s_3	C	-	A	-	-	G
s_4	C	-	A	A	T	G
s_5	C	-	-	T	-	G
s_6	C	T	-	-	A	C
s_7	C	-	A	T	A	C
s_8	G	-	A	-	A	T

Optimization criteria

- Sum-of-pairs (sum of edit distances on induced pairwise alignments)
- Tree alignment (sum of costs of edges)
- Maximum likelihood under a statistical model of sequence evolution

All three are NP-hard, even if the tree is given.

Basic techniques

Multiple sequence alignment methods generally use one or more of the following techniques to align a set S of sequences:

- Align all sequences in S to a single sequence s^* or to a profile HMM (or some other model)
- **Progressive alignment**: compute a guide tree, and then align sequences from the bottom up
- **Consistency**: infer support for homology between two letters using third sequences
- **Divide-and-conquer** (especially based on a tree)
- **Iteration** between tree estimation and alignment estimation

Adding a sequence s to an alignment A

We are given an alignment A and its profile HMM, H, and we are also given $s = s_1 s_2 \ldots s_n$, which is homologous to the sequences in A. To add s to A, we:

1. Find the maximum likelihood path through H for s
2. Use that path to add s into A.

Details for Step 2:

- Align s to the profile HMM, and note which states emit the letters of s.
- If letter s_i is emitted by match state j, put s_i in the column for this match state (might not be j).
- If letter s_i is emitted by an insertion state, put s_i in its own column after s_{i-1}. (Don't put two letters in the same column if either is emitted by an insertion state.)

Aligning a set S of sequences

Suppose S is a set of unaligned sequences and we are told they are all homologous (i.e., share a common evolutionary history) with the sequences in a family $\mathcal{F}$.

How shall we compute a multiple sequence alignment for S?

Aligning a set S of homologous sequences

- Compute an MSA A for the sequences in $\mathcal{F}$.
- Build the profile HMM H for the alignment A.
- Add all the sequences in S to A, independently.
- The alignment produced will contain all the sequences of $\mathcal{F} \cup S$; you can then restrict to just the sequences in S.

Progressive Alignment

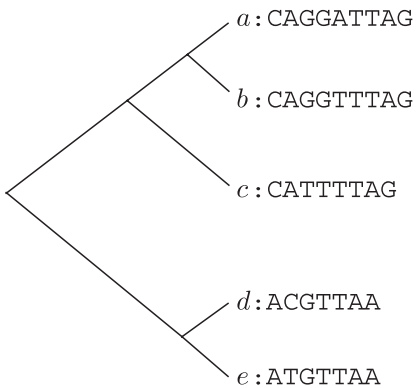
- Build a **guide tree** from the sequences
- Align the sequences from the bottom-up (aligning alignments as you go up)

a: CAGGATTAG
b: CAGGTTTAG
c: CATTTTAG
d: ACGTTAA
e: ATGTTAA

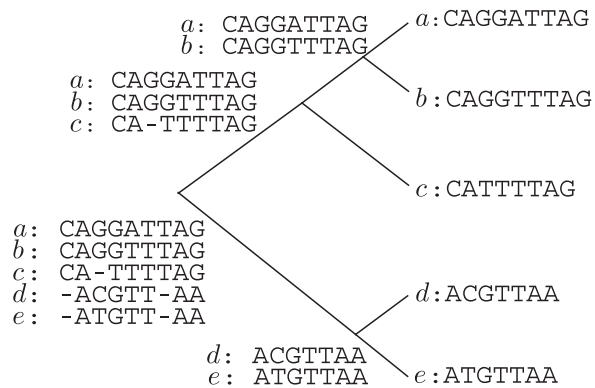
(a) input

	a	b	c	d	e
a	0	1	3	4	4
b	1	0	2	4	4
c	3	2	0	5	5
d	4	4	5	0	1
e	4	4	5	1	0

(b) pairwise distances



(c) Guide tree



(d) Progressive alignment

Aligning alignments

In a progressive alignment, alignments on disjoint sets are aligned together, to make an alignment on the combined set of sequences.

To do this, the two alignments are first represented by profiles, and then these profiles are aligned to each other.

This is performed using dynamic programming, similar to Needleman-Wunsch.

Examples of methods that can align two alignments include Opal and Muscle.

However, another approach is to represent each of the alignments as profile HMMs, and then align the two profile HMMs.

Using libraries of pairwise alignments, part 1

Suppose we have a set S of sequences, and a library L of pairwise alignments for the sequences in S.

For each pair x, y of letters (one from each of two sequences), you have the frequency with which the two letters are aligned in L (i.e., the *support* for the homology pair x, y).

Given a *library* of pairwise alignments, we can define the support for all homology pairs, and then seek the best MSA for these support values.

Using libraries of pairwise alignments, part 2

The *consistency technique* is another way of using a library L of pairwise alignments.

Another way of defining the support for the homology pair is the number of letters z (in a third sequence) such that x and z and y and z are aligned in some pairwise alignments in P.

This is how "consistency" is defined – support via a third sequence.

Many of the best MSA methods (e.g., T-Coffee and ProbCons) use the consistency technique in some way, and differ mainly in how they construct the library.

Statistical alignment estimation

Some alignment methods are based on explicit parametric models of sequence evolution that include insertions and deletions (indels) as well as substitutions. Examples:

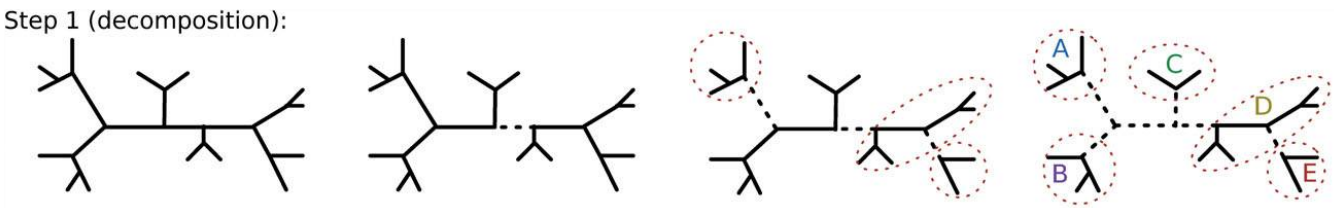
- BAli-Phy
- StatAlign
- Prank
- PAGAN

These likelihood-based methods tend to be more computationally intensive than standard MSA methods, but have appealing statistical properties.

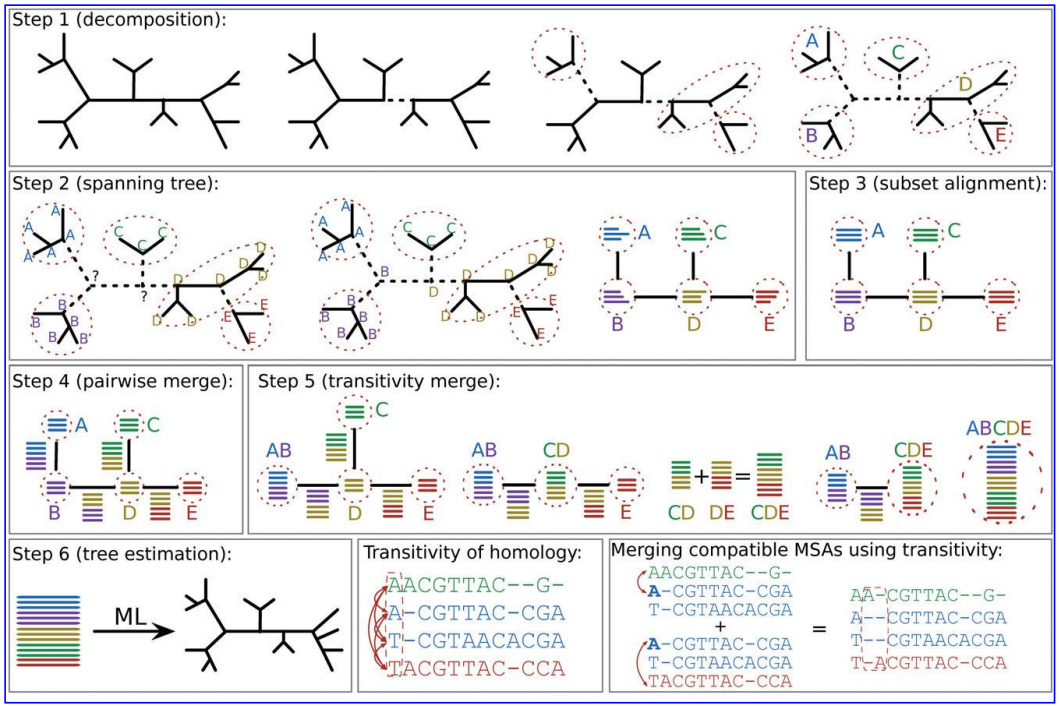
Divide-and-conquer using trees, cont.

The main objective of divide-and-conquer is to scale good MSA methods to larger datasets, so that they are more accurate or can analyze larger datasets.

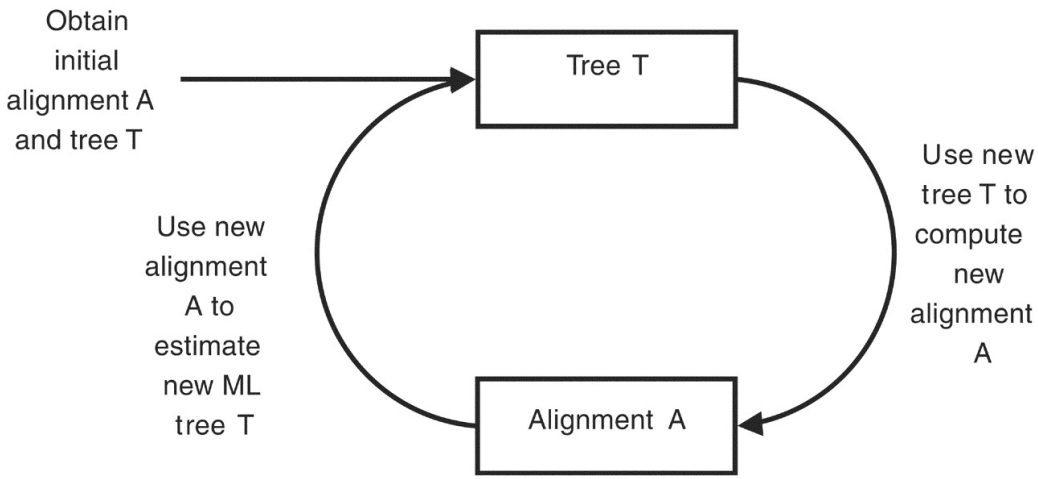
- Build a guide tree (perhaps by computing pairwise edit distances and then a tree based on the distances)
- Divide sequence dataset into disjoint subsets using the guide tree
- Align subsets
- Align alignments together (e.g., profile-profile alignment)



One PASTA iteration



Iteration between MSA and tree estimation



How to evaluate methods

Given an estimated and true (or reference alignment), we can compute various statistics, many of which are based on "homology pairs":

- SPFN: sum of the false negative homology pairs
- SPFP: sum of the false positive homology pairs
- TC: total column score
- Compression: ratio of the estimated alignment length to true alignment length
- Distance between gap length distributions

Issues to consider

- Most methods can only handle indels and substitutions (i.e., no rearrangements or duplications).
- Most methods assume full-length sequences.
- Statistical methods are all based on models of sequence evolution, and the models are limited.
- Mosts methods cannot analyze very large datasets.
- Evaluation of methods is tricky.