CSE 584A Class 25

Jeremy Buhler

April 25, 2016

1 Intro to Multiple Alignment

Remember the motivation for inexact matching?

- Why did we formulate alignment of 2 seqs like we did?
- Assumed that there was an *ancestral* sequence that gave rise to them both
- Only via permitted set of mutation events (subs, indel)
- *This idea makes sense for more than two sequences*

Consider a correspondence among 3 or more sequences, all from a common ancestor, like this:

- As before, we have modern residues with a common ancestor
- We can align any two seqs as we did for pairwise alignment
- In fact, can align *all three* seqs at once, showing correspondence across them all!

- This is called a (global) *multiple alignment*
- Can include any number of sequences related by common ancestry

Multiple alignment is essential to modern genomics, since we are getting more and more related species' genomes!

2 Scoring Multiple Alignments

There are lots of criteria by which to judge a multiple alignment. Here is a classic one that we can analyze.

- Let M be a multiple alignment of strings $s_1 \dots s_n$.
- **Defn**: the pairwise alignment of s_i, s_j *induced* by M, denoted $A_{ij}(M)$, is obtained by taking just the rows for s_i and s_j from M.

- Pick your favorite scoring system for pairwise alignments.
- (Note that induced pairwise alignments can align two gaps; treat these as having score 0, or equivalently just ignore them.)
- **Notn**: Let $D_{ij}(M)$ be the score of the pairwise alignment of s_i and s_j induced by M.
- **Defn**: the *sum-of-pairs (SP) score* of alignment M, denoted $\mathrm{SP}(M)$, is the sum of scores of all its induced pairwise alignments; that is,

$$\mathrm{SP}(M) = \sum_{i<j} D_{ij}(M).$$

- (Can think of the SP score as rating the joint log-likelihood of each pairwise hypothesis generated by M separately.)

As before, once we have a scoring criterion, we have an optimization problem.

- Let $s_1 \dots s_n$ be a collection of sequences.
- Find a multiple alignment M of $s_1 \dots s_n$, such that the SP score of M is maximal.
- Is there a Needleman-Wunsch-like algorithm for this problem?
- As for pairwise alignments, need to be able to decompose scores of multiple alignments into scores per position.
- As before, assume scoring system for alignments is based on matrix $\sigma(x, y)$ for pairs of residues x, y.

- Extend σ with a *linear* gap penalty: $\sigma(x, -) = \sigma(-, x) = g$.
- As suggested above, define $\sigma(-, -) = 0$.
- **Notn**: for a pairwise alignment A, let $A[k]$ denote the kth symbol pair in A, and let $\sigma(A[k])$ denote its score.
- For a single alignment A with m positions, score of A is simply

$$\sum_{k=1}^{m} \sigma(A[k])$$

- For a multiple alignment M with m positions,

$$\begin{aligned} \mathrm{SP}(M) &= \sum_{i<j} D_{ij}(M) \\ &= \sum_{i<j} \sum_{k=1}^{m} \sigma(A_{ij}[k]) \\ &= \sum_{k=1}^{m} \left[\sum_{i<j} \sigma(A_{ij}[k]) \right] \end{aligned}$$

- Last expression is a sum of per-position SP scores $\mathrm{SP}(M[k])$.
- Hence, as before, score of alignment M is score of last position plus score of "rest," and these two parts are independent given M.

What does algorithm look like?

- In general, a partial alignment may use i_1 chars from s_1, i_2 chars from s_2, and so on up to i_n chars from s_n.
- Want to compute $G(i_1, i_2, \ldots i_n)$, score of opt alignment between $s_1[1..i_1], s_2[1..i_2], \ldots s_n[1..i_n]$.
- How many ways can this alignment end?
- With $n = 2$, we had 3 "configurations": one, or the other, or both seqs end with a gap.
- With n sequences, M can end with *any* configuration that does not put gaps in all seqs.
- There are $2^n - 1$ such configurations $c_1 \ldots c_{2^n-1}$.
- Example for $n = 3$:

- As before, opt score $G(i_1, i_2, \ldots i_n)$ is best of scores for opt alignments ending in each configuration.
- Score for a fixed configuration is score of last posn plus opt score of "rest" of alignment.
- Example for $n = 3$:

- As before, this idea derives a recurrence. Need to compute $G(|s_1|, |s_2|, \ldots |s_n|)$.
- Cost per cell is $\Theta(2^n)$; number of cells is $\Theta(\prod_{i=1}^{n} |s_i|)$.
- Hence, for n seqs of length m, cost is $\Theta((2m)^n)$.

3 Can We Do Better?

Exponential time to compute an MA kinda sucks. Is it really this hard?

- Probably. Optimal MA problem with SP scoring is NP-hard.
- Should we give up hope? Never!
- Consider the following (not necessarily optimal!) algorithm for aligning the set of strings $S = \{s_1 \ldots s_n\}$.
- Pick a *center string* $s_c \in S$.
- For each $s_i \in S - \{s_c\}$,
  - Compute an optimal pairwise alignment A_{ic} of s_i to s_c.
  - If this alignment introduces a new gap in s_c after position p, add this gap to all seqs s_j that have been aligned to s_c so far.
  - Let s'_c be the string derived from s_c in alignment A_{ic}, including all its gaps.
  - Replace s_c by s'_c.
- To compute final M, stack all s_i's with their gaps together.
- Repeat above for all possible $s_c \in S$, and keep the alignment M with best SP score.

Example:

This method is called *center-star* alignment.

4 Setup for Proving Quality of Center-STar Alignment

To prove that center-star alignment is a good idea, we start with an assumption.

- Recall that our alignment scoring function is $\sigma(x, y)$.
- We will assume that σ is a *metric distance*.
- Three implications:
  1. $\sigma(x, x) = 0$
  2. $\sigma(x, y) = \sigma(y, x)$ (symmetry)
  3. *triangle inequality*: for all $x, y, z \in \Sigma \cup \{-\}$,

$$\sigma(x, y) \leq \sigma(x, z) + \sigma(z, y)$$

- This rule makes good sense for, e.g., edit distances
- But note that match bonus / mismatch penalty does *not* satisfy this property!
- But often, we can convert similarity measures into distances.
- For example, if $\sigma(x, x) = \alpha$ for all x, and $\sigma(x, y) < \alpha$ for $y \neq x$, then define distance $\delta(x, y) = \alpha - \sigma(x, y)$.
- (This still may not satisfy the triangle inequality, but then again, it might – BLOSUM62 on protein is at least close)
- Note that with this change, optimal alignments have *minimal*, not *maximal*, scores.

Now, let's establish some notation...

- Consider a center star alignment M_c of string set $S = \{s_1 \ldots s_n\}$ with center string $s_c \in S$.
- Let $d_c(s_i, s_j) = D_{ij}(M_c)$ be score of pairwise alignment A_{ij} of s_i, s_j induced by M_c.
- By defn, $\mathrm{SP}(M_c) = \sum_{i<j} d_c(s_i, s_j)$.
- Define $D(s_i, s_j)$ be score of opt pairwise alignment of s_i, s_j (independent of any mult align).
- (Note that $d_c(s_i, s_i) = D(s_i, s_i) = 0$.)

And now, some important observations...

- Observe that $d_c(s_i, s_j) \geq D(s_i, s_j)$.
- Moreover, $d_c(s_i, s_c) = D(s_i, s_c)$ for every s_i!
- Why? Center-star algorithm explicitly aligns each s_i to s_c (perhaps with extra gaps)
- (Best alignment w/o extra gaps implies algn of same score with them, and we cannot do *better* by adding more gaps.)
- Finally, can extend triangle inequality to aligned seqs:
- **Lemma**: for all sets of three strings $s_i, s_j, s_q \in S$,

$$d_c(s_i, s_j) \leq d_c(s_i, s_q) + d_c(s_q, s_j)$$

- **Pf**: observe that we can split $d_c(s_i, s_j)$ up into scores of its component columns. Hence

$$\begin{aligned} d_c(s_i, s_j) &= \sum_{k=1}^{m} \sigma(s_i[k], s_j[k]) \\ &\leq \sum_{k=1}^{m} \sigma(s_i[k], s_q[k]) + \sigma(s_q[k], s_j[k]) \\ &= d_c(s_i, s_q) + d_c(s_q, s_k). \end{aligned}$$

5 And Now, the Proof

Claim: Let M^* be an alignment of seqs $S = \{s_1 \ldots s_n\}$ with optimal (i.e., minimal) SP score $\mathrm{SP}(M^*)$, and let M_{c^*} be best center-star alignment of S, which is around center s_{c^*}. Then

$$\frac{\mathrm{SP}(M_{c^*})}{\mathrm{SP}(M^*)} \leq \frac{2(n-1)}{n}.$$

- **Pf**: proceed in three steps:
  1. get lower bound on $\mathrm{SP}(M^*)$
  2. get upper bound on $\mathrm{SP}(M_{c^*})$
  3. show that ratio of these two bounds is as claimed

- First step:

$$\begin{aligned} \mathrm{SP}(M^*) &= \sum_{i<j} D_{ij}(M^*) \\ &\geq \sum_{i<j} D(s_i, s_j) \\ &= 1/2 \sum_{i \neq j} D(s_i, s_j) \end{aligned}$$

(because induced alignments, even under M^*, are never better than optimal pairwise)

- Now s_{c^*} is *best* center string, so we have that, for any j,

$$\sum_i D(s_i, s_{c^*}) \leq \sum_i D(s_i, s_j)$$

- It follows that

$$\begin{aligned} \mathrm{SP}(M^*) &\geq 1/2 \sum_{i \neq j} D(s_i, s_j) \\ &= 1/2 \sum_j \sum_i D(s_i, s_j) \\ &\geq 1/2 \sum_j \sum_i D(s_i, s_{c^*}) \\ &= 1/2n \sum_i D(s_i, s_{c^*}) \end{aligned}$$

- Second step:

$$\begin{aligned} \mathrm{SP}(M_{c^*}) &= \sum_{i<j} D_{ij}(M_{c^*}) \\ &= \sum_{i<j} d_{c^*}(s_i, s_j) \\ &= 1/2 \sum_{i \neq j} d_{c^*}(s_i, s_j) \\ &\leq 1/2 \sum_{i \neq j} d_{c^*}(s_i, s_{c^*}) + d_{c^*}(s_{c^*}, s_j) \\ &= 1/2 \sum_{i \neq j} D(s_i, s_{c^*}) + D(s_{c^*}, s_j) \\ &= 1/2 \cdot 2(n-1) \sum_i D(s_i, s_{c^*}) \end{aligned}$$

(fourth step by triangle ineq, final step by sneakiness)

- Combining these inequalities, conclude that

$$\begin{aligned}\frac{\mathrm{SP}(M_{c^*})}{\mathrm{SP}(M^*)} &\leq \frac{1/2 \cdot 2(n-1)\sum_i D(s_i, s_{c^*})}{1/2 \cdot n \sum_i D(s_i, s_{c^*})} \\ &= \frac{2(n-1)}{n}.\end{aligned}$$

  QED!

So what?

- For $n = 3$, center-star alignment score is within 4/3 of optimal.
- For *any* number of seqs, score is within factor of 2 of optimal.
- For any given alignment M_{c^*}, we can estimate how bad it is vs opt by comparing its SP-score to $\sum_{i<j} D(s_i, s_j)$,
- (Pf showed this to be lower bound on opt)
- Typically, might use as starting pt for "tweaking" algos to get a better alignment