BMI/CS 776
Lecture #16
Multiple Alignment - AMAP

Colin Dewey
2007.03.15

Overemphasis on sensitivity

- Performance of most multiple alignment programs has been evaluated in terms of sensitivity (recall) alone
  - sensitivity = number of correctly predicted homologous positions / number of true pairs of homologous positions
- Precision (referred to as specificity in the multiple alignment literature), is also very important, if not equally important
- Lack of ROC analysis, tunable parameters for tradeoff

AMAP

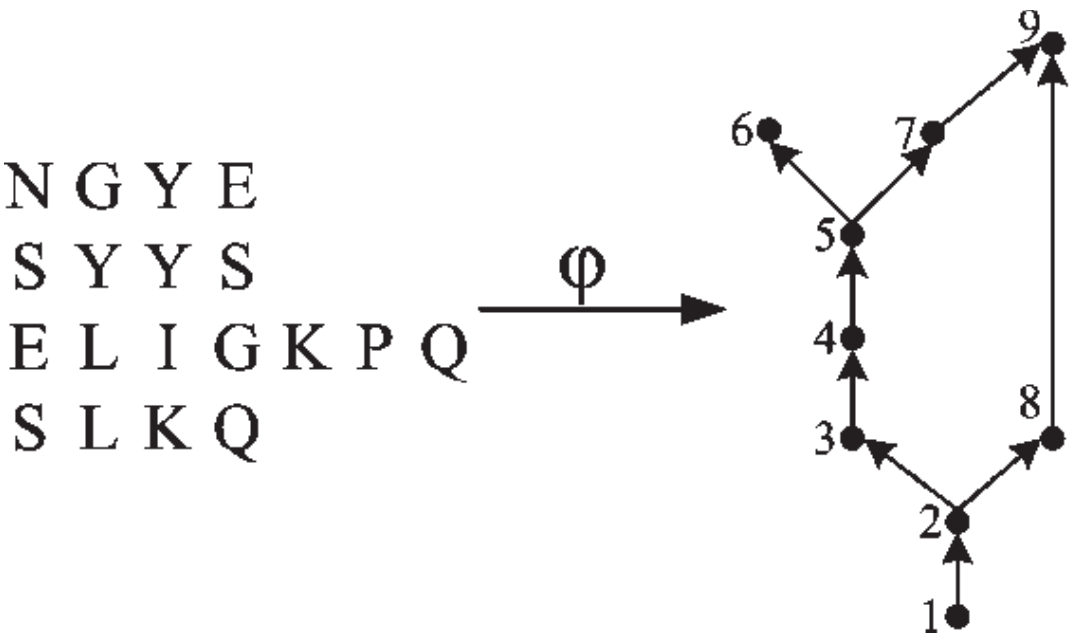
- Schwartz & Pachter, 2007
- Key ideas:
  - Objective function based on PHMM probabilities and alignment metric - *alignment metric accuracy*
  - *Sequence annealing* - Alignment constructed one match at a time. Not progressive!

Partial global multiple alignment

A *partial global multiple alignment* of sequences $\sigma^1, \ldots, \sigma^k$ is a partially ordered set $P = \{c_1, \ldots, c_m\}$ together with a surjective function $\varphi : S_{\sigma^1,\ldots,\sigma^k} \to P$ such that $\varphi((i, j_1, \epsilon_1)) \leq \varphi((i, j_2, \epsilon_2))$ if $j_1 \leq j_2$.

- $c_1, \ldots, c_m$: columns in multiple alignment
- P : the "alignment poset"
- *surjective*: φ maps at least one sequence position to every column c_i
- (i, j, ϵ): position j in sequence i on strand ϵ

Partial global alignment example



unaligned sequences

poset

– – N G Y E – – –
– – S Y Y S – – –
E L I G K – P – Q
S L – – – – – K Q
1 2 3 4 5 6 7 8 9

linear extension of poset (global multiple alignment)

Schwartz & Pachter , 2007

Sequence annealing algorithm

1: $M_L \leftarrow M_{Null}$
2: $i \leftarrow L$
3: **while** $\exists c_k^{M_i}, c_l^{M_i}$ such that $c_k^{M_i}$ and $c_l^{M_i}$ can be merged to produce M' and $f(M') \geq f(M_i)$ **do**
4: $M_{i-1} \leftarrow M'$
5: $i \leftarrow i - 1$
6: **end while**

null alignment

total length of sequences

column *l* of alignment *i*

score of alignment *i*

Schwartz & Pachter , 2007

Sequence annealing properties

$$M_L \supset M_{L-1} \supset M_{L-2} \ldots \supset M_r$$

M_i associated with poset P_i, where $|P_i| = i$

$$f(M_{i+1}) \leq f(M_i)$$

M_{i+1} transformed to M_i by merging two columns, $c_j^{M_{i+1}}$ and $c_k^{M_{i+1}}$, into one $c_l^{M_i}$

Example of Multiple Alignment by Sequence Annealing

Ariel Schwartz and Lior Pachter
University of California, Berkeley
http://bio.math.berkeley.edu/amap/

Merging columns

- Need to perform two tasks
  - Check if two columns can be merged
  - Need to update poset after merge
- Solved by using an *online topological ordering* algorithm
  - Edges given one at a time

Choosing columns to merge

- Each pair of columns is assigned a weight
- Positively weighted pairs placed in heap
  - Heap gives highest weight pair in constant time
- When columns are merged, weights change
  - Require that weights decrease on merge
    - Don’t need to update weights on merge, only calculate new weight on pop from heap

Alignment metric accuracy

Alignment metric:

$$d(h^i, h^j) = n + m - 2|h^i_H \cap h^j_H| - |h^i_I \cap h^j_I| - |h^i_D \cap h^j_D|$$

Alignment metric accuracy:

$$g(h^i, h^j) = 1 - \frac{d(h^i, h^j)}{n+m} = \frac{2|h^i_H \cap h^j_H| + |h^i_I \cap h^j_I| + |h^i_D \cap h^j_D|}{n+m}$$

= fraction of positions aligned the same in both alignments

For multiple alignments:

$$d(h^i, h^j) = \sum_{s^1=1}^{k-1} \sum_{s^2>s^1}^{k} d(h^i_{s^1,s^2}, h^j_{s^1,s^2}) \qquad g(h^p, h^r) = 1 - \frac{d(h^p, h^r)}{(k-1)\sum_{i=1}^{k} n_i}.$$

Expected AMA

- Using Pair HMM to give posterior probability of true alignment h^r we can calculate the expected alignment metric accuracy:

$$\mathbb{E}_{h^r}[g(h,h^r)] = \frac{1}{n+m}\left(\sum_{(i,j)\in h_H}\mathbb{P}[\sigma_i^1 \diamond \sigma_j^2|\sigma^1,\sigma^2] + \sum_{i\in h_D}\mathbb{P}[\sigma_i^1 \diamond -|\sigma^1,\sigma^2] + \sum_{j\in h_I}\mathbb{P}[- \diamond \sigma_j^2|\sigma^1,\sigma^2]\right)$$

- This is the pairwise expected AMA, the multiple alignment expected AMA is simply the sum-of-pairs version

AMAP objective function

$$f^{G_f}(M) = \sum_{\sigma^a, \sigma^b \mid a \neq b} \left(\sum_{\{(j,k) \mid \varphi^M(\sigma_j^a) = \varphi^M(\sigma_k^b)\}} P(\sigma_j^a \diamond \sigma_k^b \mid \sigma^a, \sigma^b, \theta) \right.$$

$$+ G_f \sum_{\{j \mid \forall \sigma_k^b \varphi^M(\sigma_j^a) \neq \varphi^M(\sigma_k^b)\}} P(\sigma_j^a \diamond - \mid \sigma^a, \sigma^b, \theta)$$

$$\left. + G_f \sum_{\{k \mid \forall \sigma_j^a \varphi^M(\sigma_j^a) \neq \varphi^M(\sigma_k^b)\}} P(- \diamond \sigma_k^b \mid \sigma^a, \sigma^b, \theta) \right).$$

- Family of functions, parameterized by G_f ("gap-factor")
  - $G_f = 0$: maximize f_D score (sensitivity)
  - $G_f = 0.5$: maximize expected AMA score
  - $G_f > 0.5$: higher specificity, lower sensitivity

Weight functions

- With the following definitions:

$$P_{match} = \sum_{\sigma_i^a \in \varphi^{-1}(c_k)} \sum_{\sigma_j^b \in \varphi^{-1}(c_l)} \mathbb{P}[\sigma_i^a \diamond \sigma_j^b | \sigma^a, \sigma^b]$$

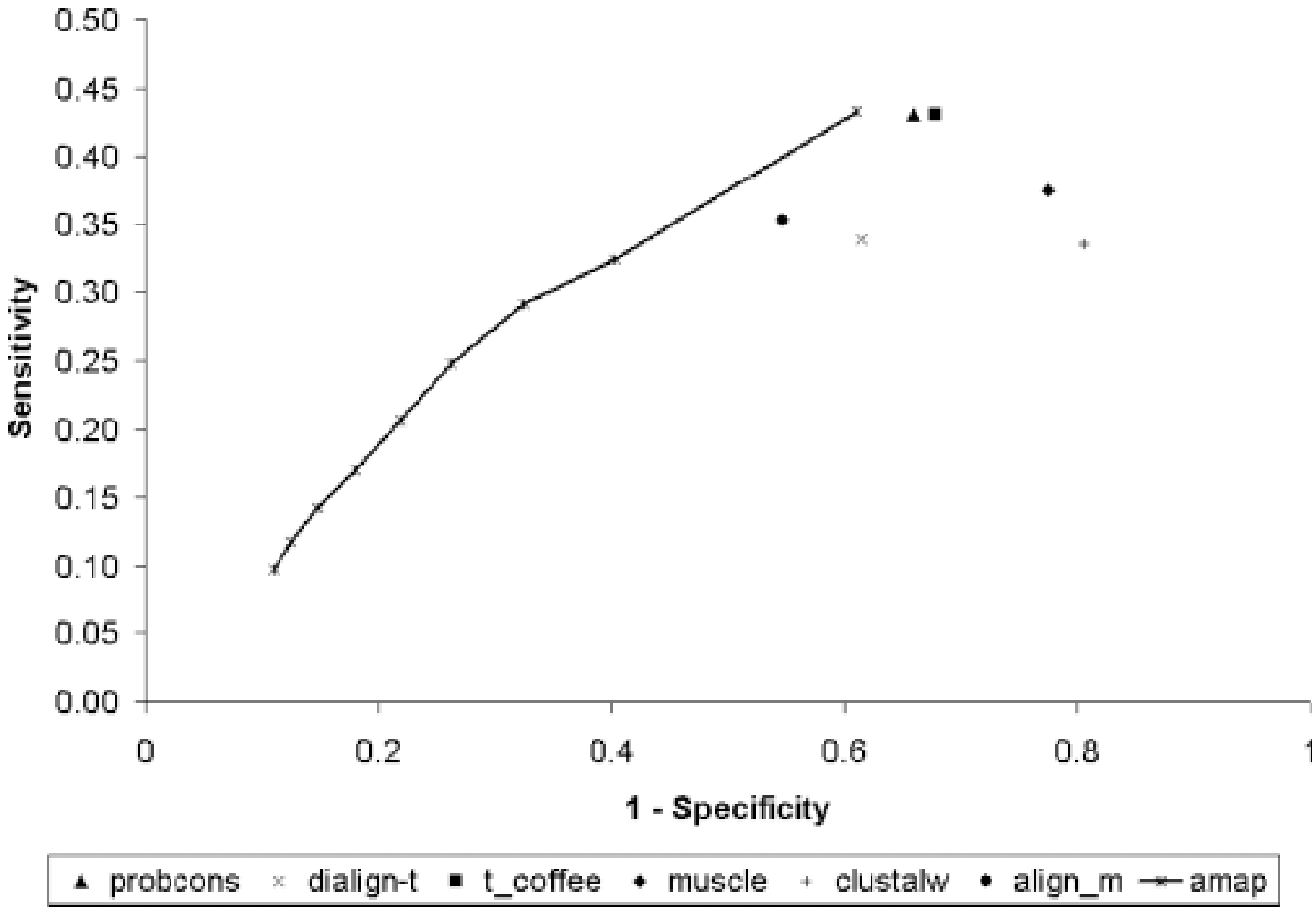
$$P_{gap} = \sum_{\sigma_i^a \in \varphi^{-1}(c_k)} \sum_{\sigma_j^b \in \varphi^{-1}(c_l)} \mathbb{P}[\sigma_i^a \diamond - | \sigma^a, \sigma^b] + \mathbb{P}[- \diamond \sigma_j^b | \sigma^a, \sigma^b]$$

- Define two possible weight functions:

$$w_{maxstep}^{G_f}(c_k, c_l) = \frac{P_{match} - G_f P_{gap}}{|\varphi^{-1}(c_k)||\varphi^{-1}(c_l)|}$$

$$w_{tgf}^{G_f}(c_k, c_l) = \frac{P_{match}}{P_{gap}} - G_f$$

Comparison of AMAP



Schwartz & Pachter , 2007

AMAP Performance

Program	Twilight (209)			Superfamilies (425)			Overall by alignments			Overall by positions			Average time
	f_D	f_M	*AMA*	f_D	f_M	*AMA*	f_D	f_M	*AMA*	f_D	f_M	*AMA*	Seconds
Align-*m*	21.6	23.6	51.7	49.2	45.6	56.9	40.1	38.3	55.2	35.2	45.4	56.6	12.7
CLUSTALW	25.6	14.7	24.9	54.0	38.1	43.8	44.7	30.4	37.6	33.6	19.5	28.2	**0.4**
DIALIGN-T	21.3	19.8	45.5	49.9	44.9	54.8	40.4	36.6	51.7	33.9	38.6	52.5	1.4
MUSCLE	27.3	16.4	27.6	56.3	40.3	46.4	46.8	32.4	40.2	37.5	22.5	31.7	2.1
ProbCons	**32.1**	21.1	37.4	**59.8**	44.4	51.8	**50.7**	36.7	47.0	43.0	34.3	47.0	4.5
T-Coffee	26.7	18.1	35.2	56.5	42.8	50.3	46.7	34.7	45.3	39.4	31.5	44.5	11.3
$AMAP_{sn}$	30.9	22.4	40.9	58.8	45.3	53.3	49.6	37.8	49.2	**43.3**	39.1	51.9	2.4
AMAP	24.0	28.3	51.2	52.8	54.6	59.5	43.3	45.9	56.8	32.5	59.7	**59.6**	1.7
$AMAP_{sp}$	14.5	**41.5**	**56.5**	38.7	**69.4**	**60.2**	30.7	**60.2**	**59.0**	20.7	**78.1**	58.9	1.4

Entries show the average developer (f_D), modeler (f_M) and *AMA* scores. Best results are shown in boldface. All numbers have been multiplied by 100.

Schwartz & Pachter , 2007