CMSC423: Bioinformatic Algorithms, Databases and Tools

Multiple sequence alignment

Motif finding

Multiple sequence alignment

- Simultaneously identify relationship between multiple sequences

```
HBB_HUMAN      FFESFGDLSTPDAVMGNPKVKAHGKKVL-----GAFSDGLAHLDNLKGTF
HBB_HORSE      FFDSFGDLSNPGAVMGNPKVKAHGKKVL-----HSFGEGVHHLDNLKGTF
HBA_HUMAN      YFPHF-DLS-----HGSAQVKGHGKKVA-----DALTNAVAHVDDMPNAL
HBA_HORSE      YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVGHLDDLPGAL
MYG_PHYCA      KFDRFKHLKTEAEMKASEDLKKHGVTVL-----TALGAILKKKGHHEAEL
GLB5_PETMA     FFPKFKGLTTADQLKKSADVRWHAERII-----NAVNDAVASMDDTEKMS
LGB2_LUPLU     LFSFLKGTSEVP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATL
                *  :   .       . .:: *.  :       :.   :
```

- Note: multiple alignment implies (not necessarily optimal) pairwise alignment between the individual sequences

```
HBA_HUMAN      YFPHF-DLS-----HGSAQVKGHGKKVA-----DALTNAVAHVDDMPNAL
HBA_HORSE      YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVGHLDDLPGAL
```

Multiple alignment – formal definition

- M – multiple sequence alignment for $s_1,...,s_k$
- $D(s_i,s_j)$ – optimal score of alignment between s_i, s_j
- $d(s_i,s_j)$ – score of alignment btwn s_i, s_j induced by M
- score of M $d(M) = \text{sum}_{\text{all pairs si, sj}}\ d(s_i, s_j)$
- also called sum-of-pairs

- Optimal multiple alignment minimizes d(M)

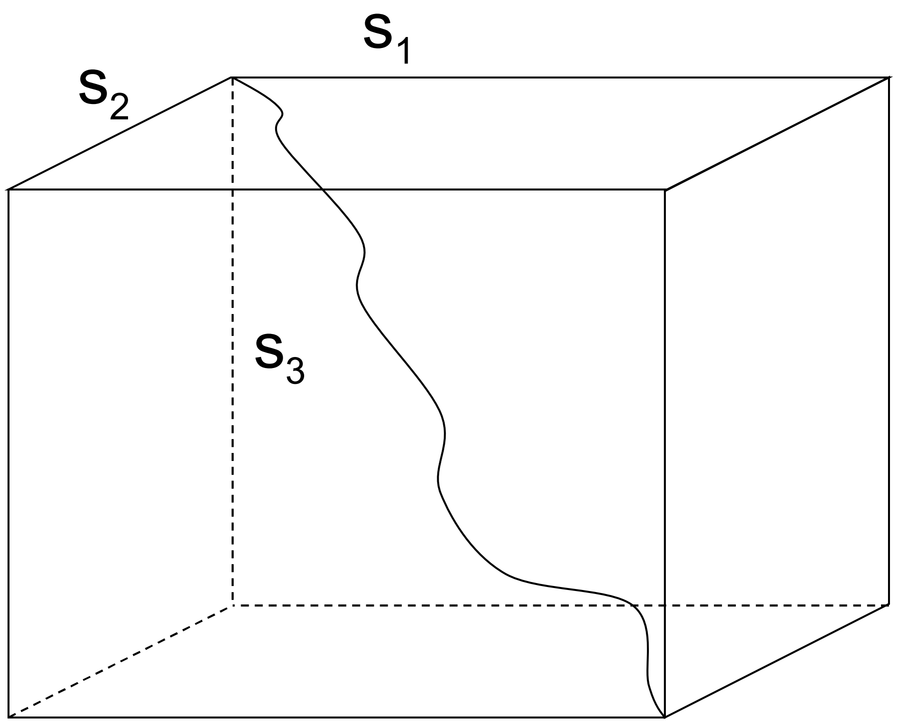
- Computing optimal d(M) is NP hard
- Note: in multiple alignment we think of "distance" rather than "similarity"

But....here's a solution

- Dynamic programming solution.   e.g. 3 sequences

- Score(i, j, k) – optimal alignment between s1[1..i], s2[1..j], s3[1..k] – do DP as usual

- s(i,j,k) = max {
        s(i-1, j-1, k-1) +
 match(s1[i], s2[j], s3[k]),
 ...



But... it's expensive

- 3 sequences – need to fill in the cube $O(n^3)$
- k sequences – k-dimensional cube $O(n^k)$ time/space

- There are tricks that can help – similar to AI techniques for reducing the search space

- Basic idea – if we can estimate optimal score, we can prune the search space.

- Note – these are just heuristics – not guaranteed to work faster

Alternative – approximation algorithm

- Can we efficiently compute a multiple alignment with a score that's not too bad?
- The Star method:
  - build all k^2 pairwise alignments ($O(k^2n^2)$)
  - pick sequence sc that is closest to all other sequences: sum $_{si}$ D(sc, s_i) is minimal over all choices of sc
  - iteratively align each sequence to sc

- Theorem: sum-of-pairs score of star alignment is at most twice as big as optimal multiple alignment score

Iterative alignment

```
SC YFPHFDLSHGSAQVKAHGKKVGDALTLAVGHLDDLPGAL
```

- Take sequences si in order:
  - align s1 with sc - results in gaps being inserted in both sequences

```
SC YFPHFDLSHGSAQVKAHGKKVGDALTLAVGHLDDLPGAL
S1 YFPHFDLSHG-AQVKG--KKVADALTNAVAHVDDMPNAL
```

  - align s2 with sc - if gaps must be inserted – insert in previously aligned sequences

```
SC YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVAHLDDLPGAL
S1 YFPHF-DLS-----HG-AQVKG—GKKVA-----DALTNAVAHVDDMPNAL
S2 FFPKFKGLTTADQLKKSADVRWHAERII-----NAVNDAVASMDDTEKMS
```

  - and so on (note: if gaps coincide with previously introduced gaps no need to change previously aligned sequences)

```
SC YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVAHLDDLPGAL
S1 YFPHF-DLS-----HG-AQVKG—GKKVA-----DALTNAVAHVDDMPNAL
S2 FFPKFKGLTTADQLKKSADVRWHAERII-----NAVNDAVASMDDTEKMS
S3 LFSFLKGTSEVP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATL
```

Theorem proof

- Theorem: star alignment is 2-optimal
- Assumption: distances obey triangle inequality

OPT = $\sum_{si,sj} d(s_i,s_j) \geq \sum_{si,sj} D(s_i,s_j) \geq k \sum_{si} D(s_i, sc)$

STAR = $\sum_{si,sj} d^*(s_i,s_j) \leq \sum_{si,sj}(D(s_i, sc) + D(s_j, sc))$ # triangle ineq.

$= \sum_{sj,sj} D(s_j, sc) + \sum_{sj,sj} D(s_i, sc)$

$= 2k \sum_{si} D(s_i, sc)$

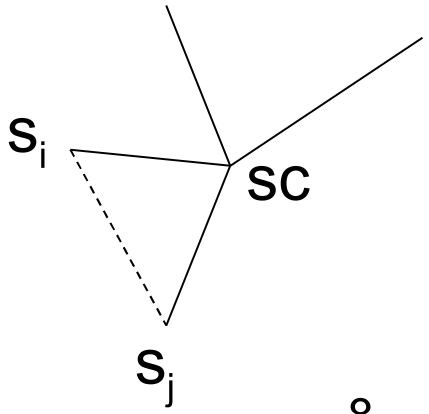
=> STAR/OPT ≤ 2 Q.E.D

note: $\sum_{si} D(s_i, sc)$ – is score optimized by choice of sc

d(si,sj) – score of alignment btwn si, sj within optimal alignment

d*(si,sj) – score of alignment btwn si, sj within star alignment

D(si,sj) – score of optimal alignment btwn si, sj



Consensus sequence

- For every column j in the alignment, pick the amino-acid AA that minimizes $\sum_i d(AA, S_i[j])$ (usually becomes majority rule)
- Intuitively – this is the sequence of the ancestor of all the sequences in the multiple alignment
- We can define the multiple alignment problem as:
  - find the multiple alignment that minimizes $\sum_i D(CO, S_i)$
- Still NP – hard, but consensus sequence useful on it's own.

```
CO YFPHFKDLS-----HGSAQVKAHGKKVG-----DALTLAVAHVDDTPGAL
S1 YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVAHLDDLPGAL
S2 YFPHF-DLS-----HG-AQVKG—GKKVA-----DALTNAVAHVDDMPNAL
S3 FFPKFKGLTTADQLKKSADVRWHAERII-----NAVNDAVASMDDTEKMS
S4 LFSFLKGTSEVP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATL
```

Iterative alignment revisited

- Pick a sequence (e.g. SC) as a starting point
- Align S1 to it & build consensus for the alignment
- Take S2 and align it to the consensus (instead of SC)
- repeat...
- Problem: consensus (or any single sequence) ignores the other sequences being aligned.
- Solution: keep track of % of each amino-acid aligned in each column
- score of alignment to profile – combination of scores to each AA.

Profile alignment

- Solution: keep track of % of each amino-acid aligned in each column
- score of alignment to profile – combination of scores to each AA.

100% F

50% S
25% N
25% -

75% A
25% Q

```
S1 YFPHF-DLS-----HGSAQVKAHGKKVG-----DALTLAVAHLDDLPGAL
S2 YFPHF-DLS-----HG-AQVKG—GKKVA-----DALTNAVAHVDDMPNAL
S3 FFPKFKGLTTADQLKKSADVRWHAERII-----NAVNDAVASMDDTEKMS
S4 LFSFLKGTSEVP--QNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATL
```

- Score(prof1, prof2) = weighted average of all pairs of amino-acids

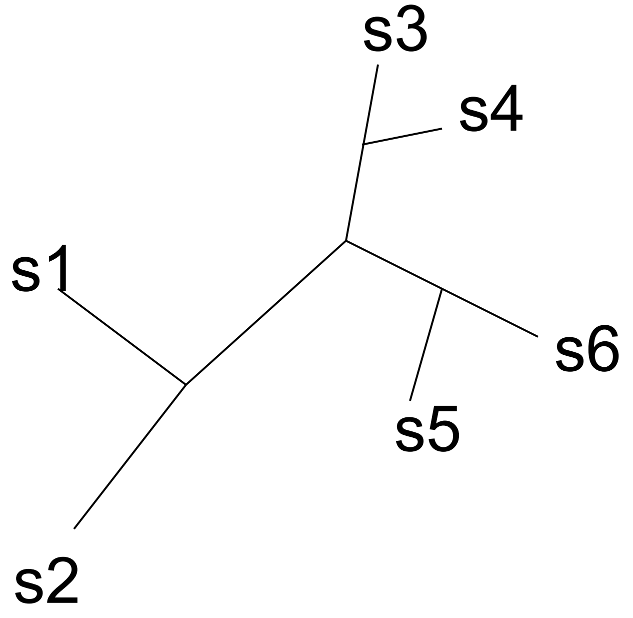
Profile alignment..cont

- Score(50%A,20%L,30%K ; 20%A,30%C,50%G) =
  0.5 * 0.2 * Score(A,A) +
  0.5 * 0.3 * Score(A,C) +
  0.5 * 0.5 * Score(A,G) +
  0.2 * 0.2 * Score(L,A) +
  0.2 * 0.3 * Score(L,C) +
  0.2 * 0.5 * Score(L,G) +
  0.3 * 0.2 * Score(K,A) +
  0.3 * 0.3 * Score(K,C) +
  0.3 * 0.5 * Score(K,G)

- hard to write down, easy to compute

CLUSTALW

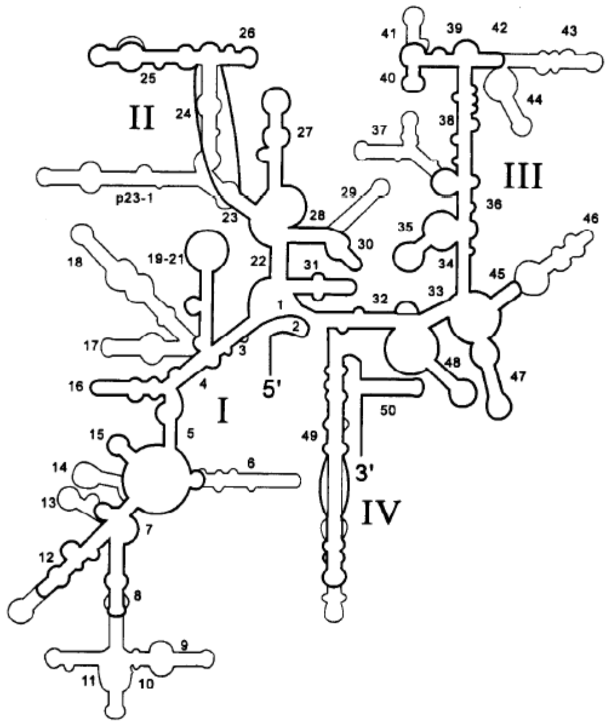
- Compute pairwise distances between strings
- Build phylogenetic tree
- Build iterative alignment by following tree edges

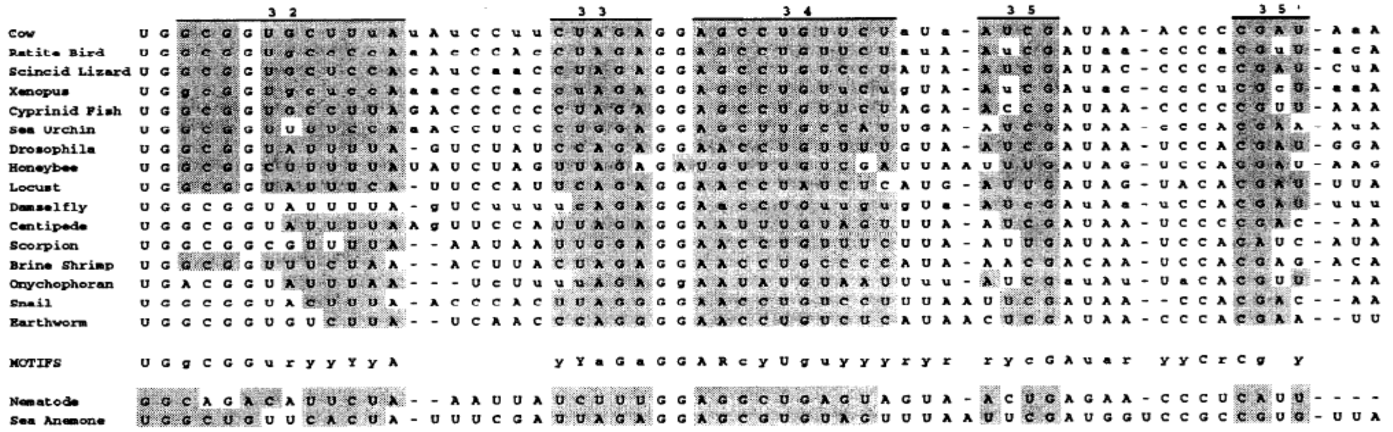


MUSCLE

- Just like ClustalW but different
- Build pairwise distances – uses fast heuristic (just count # of k-mers in common)
- Build phylogenetic tree
- Build multiple alignment based on tree
- Re-estimate distances based on tree
- Re-build tree
- Re-build multiple alignment
- etc. etc. etc.

Biological relevance of multiple alignments





```
                             3 2                           3 3                 3 4                   3 5                3 5 '
Cow            U G G C G G U G C U U U A u A u C u u C U A G A G G A G C C U G U U C U a U a - A U C G A U A A - A C C C C G A U - A a A
Ratite Bird    U G G C G G U g C c C c A a A c C C A c C U A G A G G A G C C U G U U C U a u A - A u C G A U a a - c C C a C G u U - a c A
Scincid Lizard U G G C G G U G C U C C A c A u C a a C C U A G A G G A G C C U G U C C U A U A - A U C G A U A C - C C C c C G A U - C u A
Xenopus        U G g c G G U g c u c c A a a c C C a c c u A G A G G A G C C U G U u C u g U A - A u C G A u a c - c c C u C G c U - a a A
Cyprinid Fish  U G G C G G U G C C U U A G A C C C C C C U A G A G G A G C C U G U U C U A G A - A C C G A U A A - C C C C C G U U - A A A
Sea Urchin     U G G C G G U U U U C C A a A C C U C C C U G G A G G A G C U U G C C A U U G A - A U C G A U A A - c C C A C G A A - A u A
Drosophila     U G G C G G U A U U U U A - G U C U A U C C A G A G G A A C C U G U U U U G U A - A U C G A U A A - U C C A C G A U - G G A
Honeybee       U G G C G G C U U U U U A U A U C U A G U U A G A G A U G U U U G U C G A U U A A U U U G A U A G - U C C A G G A U - A A G
Locust         U G G C G G U A U U U C A - U U C C A U U C A G A G G A A C C U A U C U C A U G - A U U G A U A G - U A C A C G A U - U U A
Damselfly      U G G C G G U A U U U U A - g U C u u u u c A G A G G A a c C U G u u g u g U a - A U c G A u A a - u C C A C G A U - u u u
Centipede      U G G C G G U A U U U U A A g U U C C A U U A G A G G A A U U U G U A G U U U A - A U C G A U A A - U C C C C G A C - - A A
Scorpion       U G G C G G C G U U U U A - - A A U A A U U G G A G G A A C C U G U U U C U U A - A U U G A U A A - U C C A C A U C - A U A
Brine Shrimp   U G G C G G U U U C U A A - - A C U U A C U A G A G G A A C C U G C C C C A U A - A A C G A C A A - U C C A C G A G - A C A
Onychophoran   U G A C G G U A U U U A A - - - U c U u u u A G A G g A A U A U G U A A U U u u - A U C G a u A u - U a C A C G U U - - A A
Snail          U G G C C G U A C U U U A - A C C C A C U U A G G G G A A C C U G U C C U U U A A U U C G A U A A - - C C A C G A C - - A A
Earthworm      U G G C G G U G U C U U A - - U C A A C C C A G G G G A A C C U G U C U C A U A A C U C G A U A A - C C C A C G A A - - U U

MOTIFS         U G g C G G u r y y Y y A             y Y a G a G G A R c y U g u y y y r y r   r y c G A u a r   y y C r C g   y

Nematode       G G C A G A C A U U C U A - - A A U U A U C U U U G G A G G C U G A G U A G U A - A C U G A G A A - C C C U C A U U - - - -
Sea Anemone    U G G C U G U U C A C U A - U U U C G A U U A G A G G A G C G U G U A G U U U A A U U C G A U G G U C C G C C G U G - U U A
```

Motif finding

Motif finding

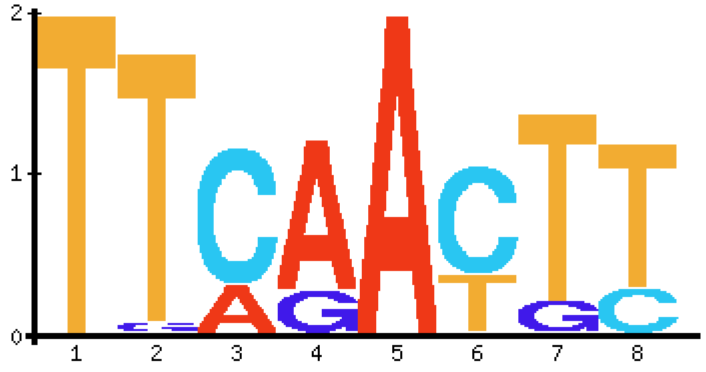
- Special case of multiple alignment – find short "motif" that occurs almost identically in multiple DNA sequences
- Local multiple alignment (the definition of multiple alignment sofar was global)
- Motif finding – special requirements
  - inexact alignment sought
  - but no gaps allowed
- Biological significance
  - gene promoters
  - transcription factor binding sites
  - other elements involved in gene regulation

Motif finding...example

TTAGAGGTTGACTA**TTCAACTT**TTGAGGAGGCCTAG*TAGAGC*
AGCCGACT**TGCAACTT**AGGCGTGGTCAGTGCCCTAA*TAGAGC*
GGCCTATTTGGGCCACTTAGACC**TTCAACTT**TTGCA*TAGAGC*
CCACAG**TTAGATGT**CCAAAAGACAAATATAGAGGGC*TAGAGC*
ACACGGACTGCG**TTCAATGC**TTACAGCAGATTGAGT*TAGAGC*
TTCAAAGACTTGACTATTG**TTCAACTT**TGAAGACTA*TAGAGC*

Promoter region $\longrightarrow$ Gene

Motif "sequence logo"



From genetics.mgh.harvard.edu/sheenlab/

Finding motifs – Gibbs sampling

- Observations:
  - since no gaps – all motifs have equal length (assume known value - m)
  - exhaustive search of promoter region is impractical: all combinations of substrings of length m among k sequences of length L = $(L - m + 1)^k$
-

Gibbs sampling... the algorithm

1. Pick random substring of length m from each of the strings
2. Construct multiple alignment (easy since no gaps) and compute profile
3. Pick random sequence s and remove from multiple alignment. Recompute profile.
4. Within removed sequence, search for best fit to profile and insert into alignment
5. Repeat until profile does not improve

Gibbs sampling...cont

- How do you find best match to profile?
- What is overall running time of algorithm?