Computational Genomics
Prof. Ron Shamir & Prof. Roded Sharan
School of Computer Science, Tel Aviv University

גנומיקה חישובית
פרופ' רון שמיר ופרופ' רודד שרן
ביה"ס למדעי המחשב,אוניברסיטת תל אביב

Lecture 4: Multiple alignment

13,15/11/12

Multiple Sequence Alignment

Reference:

Gusfield, *Algorithms on Strings, Trees & Sequences*

Some slides from:

- Jones, Pevzner, USC *Intro to Bioinformatics Algorithms* http://www.bioalgorithms.info/
- S. Batzoglu, Stanford http://ai.stanford.edu/~serafim/CS262_2006/
- Geiger, Wexler, Technion http://www.cs.technion.ac.il/~cs236522/
- Ruzzo, Tompa U. Washington CSE 590bi
- Poch, Strasbourg www.inra.fr/internet/Projets/agroBI/PHYLO/Poch.ppt
- A. Drummond, Auckland, NZ

Multiple Alignment vs. Pairwise Alignment

- Up until now we have only tried to align two sequences.
- What about more than two? And what for?
- A faint similarity between two sequences becomes significant if present in many
- Multiple alignments can reveal subtle similarities that pairwise alignments do not reveal

Multiple Alignment vs. Pairwise Alignment

- "Pairwise alignment whispers … multiple alignment shouts out loud"

  Hubbard, Lesk, Tramontano, Nature Structural Biology 1996.

Multiple Alignment Definition

Input: Sequences $S_1, S_2, \ldots, S_k$ over the same alphabet
Output: Gapped sequences $S'_1, S'_2, \ldots, S'_k$ of equal length

1. $|S'_1| = |S'_2| = \ldots = |S'_k|$
2. Removal of spaces from S'_i gives S_i for all i

Example

S_1=AGGTC

S_2=GTTCG

S_3=TGAAC

Possible alignment

A	G	G	T	-	C	-
-	G	-	T	T	C	G
T	G	-	A	A	C	-

Possible alignment

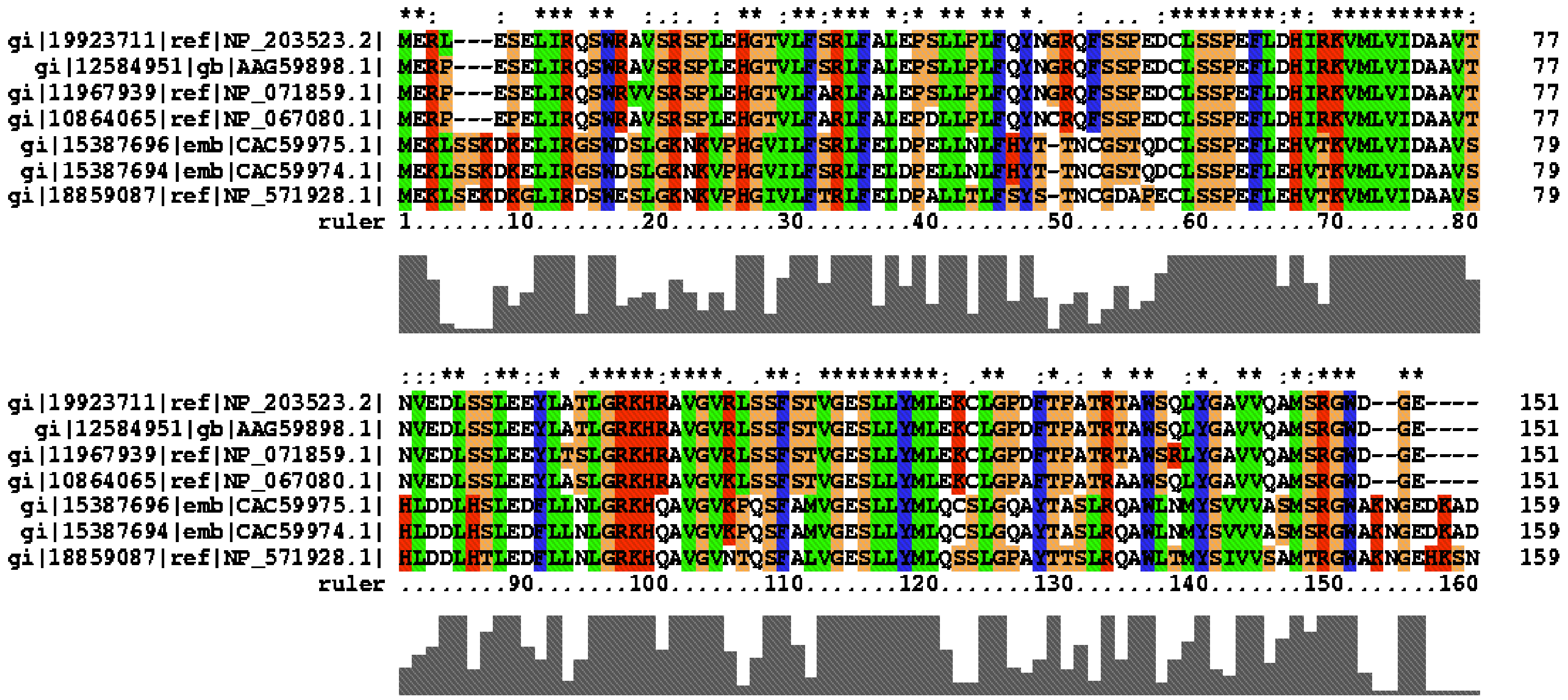
A	G	G	T	-	C	-
G	T	T	-	-	C	G
-	T	G	A	A	A	C

```
                α1                           α2                          
  vp7_btv10                                                              
               1         10         20         30         40         50         60
1 vp7_btv10    MDTIAARALTVMRACATLQEARIVLEANVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
  vp7_btv1s    MDTIAARALTVMRACATLQEARIVLEANVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
  vp7_btv17    MDTIAARALTVMRACATLQEARIVLEANVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
  vp7_btv13    MDTIAARALTVMRVCATLQEARIVLEPNVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
  vp7_btv2a    MDTIAARALTVMRACATLQEARIVLEANVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
  vp7_btv1a    MDTIAARALTVMRACATLQEARIVLEANVMEILGIAINRYNGLTLRGVTMRPTSLAQRNE
2 vp7_ehdv1    MDTIAARALTVIKACNTLKEVRIVVESNVLEILGIAVNRYNGLTLRSVTMRPTSQEQRNE
  vp7_ahsv4    MDAIAARALSVVRACVTVTDARVSLDPGVMETLGIAINRYNGLTMHSVSMRPQTQAERNE
3 vp7_brd      MDAYTARALSVLEGLAISGDPRSHRDPTTEATMSIFAMRFNSTTSRPVMGRPQ.RGKRGV
         sac

               α3                          α4     TT      α5
  vp7_btv10                                                              
                      70         80         90        100        110        120
1 vp7_btv10    MFFMSLDMMLSAAGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
  vp7_btv1s    MFFMSLDMMLSAAGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
  vp7_btv17    MFFMSLDMMLSATGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
  vp7_btv13    MFFMSLDMMVSAAGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
  vp7_btv2a    MFFMSLDMMLSAAGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
  vp7_btv1a    MFFMSLDMMLSAAGINVGPISPDYTQHMATIGVLATPEIPFTTEAANEIARVTGETSTWG
2 vp7_ehdv1    MFYMSLDMILAAANLNVGNISPDYIQNLATIGVLATPEIPYTMESANEIARMSGETGTWG
  vp7_ahsv4    MFFMSTDMVLAALNVQIGNISPDYDQALATVGALATTEIPYMVQAMNDIVRITGQMQTFG
3 vp7_brd      ITSTPQRCGIRDLEHHVAFVMPGYIQNPQTLAILARDEIPYTPSTPRRVQRYVSARMDIQ
         sac

               βA             βB       TT      TT   βC      TT  βD    TT  βE    η1
  vp7_btv10                                                              
                          130        140        150        160        170
1 vp7_btv10    PARQPYGFFLETEETFQPGRWF.MRAAQAVTAVVCGPDMIQVSLNAGARGDVQQIFQ.GR
  vp7_btv1s    PARQPYGFFLETEETFQPGRWF.MRAAQAATAVVCGPDMIQVSLNAGARGDVQQIFQ.GR
  vp7_btv17    PARQPYGFFLETEETFQPGRWF.MRAAQAVTAVVCGPDMIQVSLNAGARGDVQQIFQ.GR
  vp7_btv13    PARQPYGFFLETEEVYQPGRWF.MRAAQVVTPVVCGPMMVQVSLNAGAIGDVQQIFQ.GR
  vp7_btv2a    PARQPYGFFLETEETFQPGRWF.MRAAQAVAAVVCGPDMIQVSLNAGARGDVQQIFQ.GR
  vp7_btv1a    PARQPYGFFLETEETYQPGRWF.MRAAQAVTAVVCGPDMIQVSLNAGARGDVQQIFQ.GR
2 vp7_ehdv1    PDRQPFGYFLAAPEVTQHSRFR.QRVGQNVTTSIVSSTLAQVSMNAGARGDIQALFQ.NQ
  vp7_ahsv4    PSKVQTGPYAGAVEVQQSSRYY.VPQGRTRGGYIN.SNIAEVCMDAGAAGQVMALLAPRR
3 vp7_brd      MCERNDMLTPHYEAVTSFSVVLPAPAGGGPRCYAISATTLQIHIDPMQNEVLTPLIFPDQ
         sac
```

Example

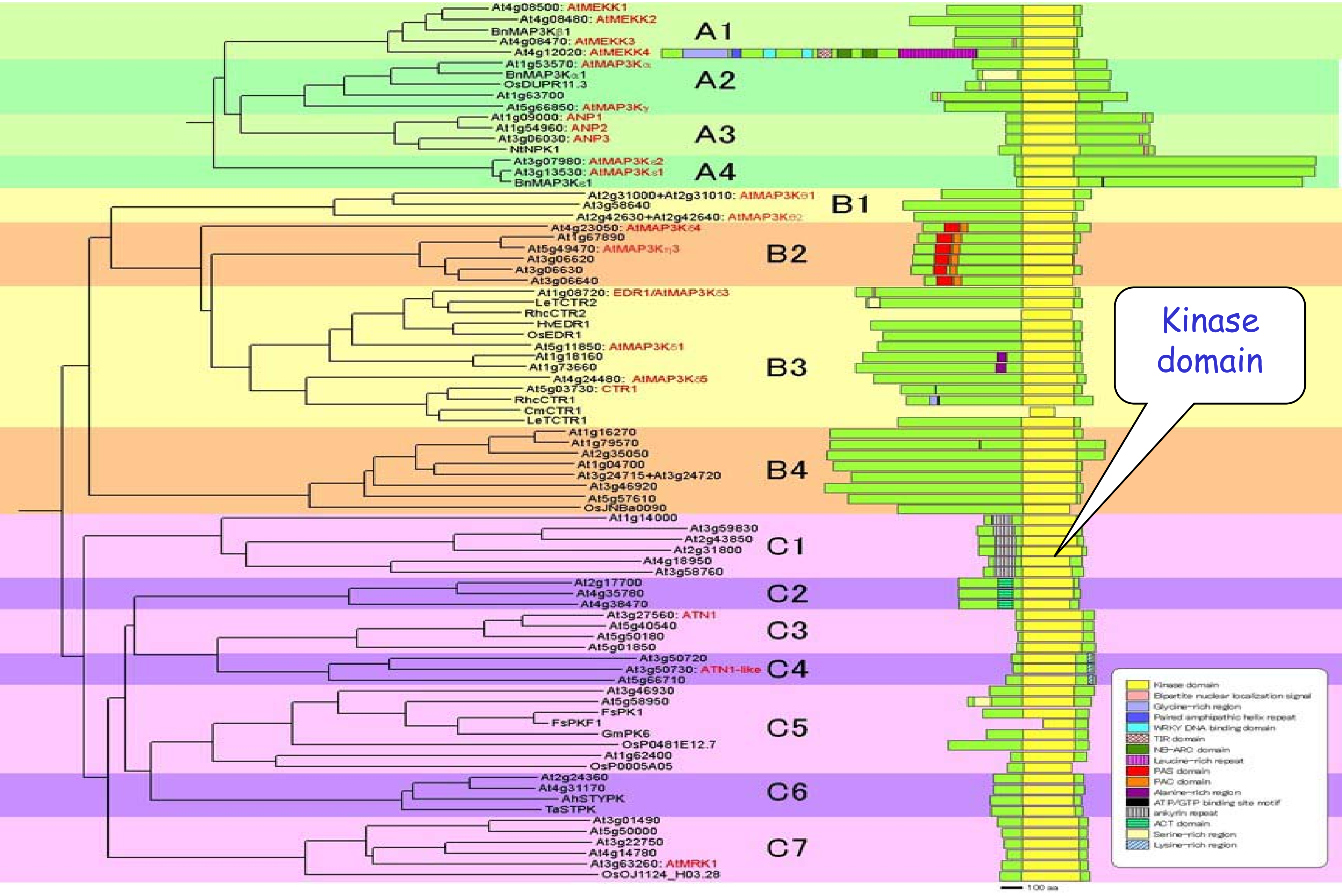


```
                             **:    :  *** ** :.:.  : ** :**:*** *:* ** ** *.  : ... :********:*: **********:
gi|19923711|ref|NP_203523.2| MERL---ESELIRQSWRAVSRSPLEHGTVLFSRLFALEPSLLPLFQYNGRQFSSPEDCLSSPEFLDHIRKVMLVIDAAVT     77
  gi|12584951|gb|AAG59898.1| MERP---ESELIRQSWRAVSRSPLEHGTVLFSRLFALEPSLLPLFQYNGRQFSSPEDCLSSPEFLDHIRKVMLVIDAAVT     77
gi|11967939|ref|NP_071859.1| MERP---ESELIRQSWRVVSRSPLEHGTVLFARLFALEPSLLPLFQYNGRQFSSPEDCLSSPEFLDHIRKVMLVIDAAVT     77
gi|10864065|ref|NP_067080.1| MERP---EPELIRQSWRAVSRSPLEHGTVLFARLFALEPDLLPLFQYNCRQFSSPEDCLSSPEFLDHIRKVMLVIDAAVT     77
 gi|15387696|emb|CAC59975.1| MEKLSSKDKELIRGSWDSLGKNKVPHGVILFSRLFELDPELLNLFHYT-TNCGSTQDCLSSPEFLEHVTKVMLVIDAAVS     79
 gi|15387694|emb|CAC59974.1| MEKLSSKDKELIRGSWDSLGKNKVPHGVILFSRLFELDPELLNLFHYT-TNCGSTQDCLSSPEFLEHVTKVMLVIDAAVS     79
gi|18859087|ref|NP_571928.1| MEKLSEKDKGLIRDSWESLGKNKVPHGIVLFTRLFELDPALLTLFSYS-TNCGDAPECLSSPEFLEHVTKVMLVIDAAVS     79
                       ruler 1.......10........20........30........40........50........60........70........80

                             :::** :**::* .*****:****. .**: *********: .**  :*.: * **  :*. ** :*:***   **
gi|19923711|ref|NP_203523.2| NVEDLSSLEEYLATLGRKHRAVGVRLSSFSTVGESLLYMLEKCLGPDFTPATRTAWSQLYGAVVQAMSRGWD--GE----    151
  gi|12584951|gb|AAG59898.1| NVEDLSSLEEYLATLGRKHRAVGVRLSSFSTVGESLLYMLEKCLGPDFTPATRTAWSQLYGAVVQAMSRGWD--GE----    151
gi|11967939|ref|NP_071859.1| NVEDLSSLEEYLTSLGRKHRAVGVRLSSFSTVGESLLYMLEKCLGPDFTPATRTAWSRLYGAVVQAMSRGWD--GE----    151
gi|10864065|ref|NP_067080.1| NVEDLSSLEEYLASLGRKHRAVGVKLSSFSTVGESLLYMLEKCLGPAFTPATRAAWSQLYGAVVQAMSRGWD--GE----    151
 gi|15387696|emb|CAC59975.1| HLDDLHSLEDFLLNLGRKHQAVGVKPQSFAMVGESLLYMLQCSLGQAYTASLRQAWLNMYSVVVASMSRGWAKNGEDKAD    159
 gi|15387694|emb|CAC59974.1| HLDDLHSLEDFLLNLGRKHQAVGVKPQSFAMVGESLLYMLQCSLGQAYTASLRQAWLNMYSVVVASMSRGWAKNGEDKAD    159
gi|18859087|ref|NP_571928.1| HLDDLHTLEDFLLNLGRKHQAVGVNTQSFALVGESLLYMLQSSLGPAYTTSLRQAWLTMYSIVVSAMTRGWAKNGEHKSN    159
                       ruler ........90.......100.......110.......120.......130.......140.......150.......160
```

Multiple sequence alignment of 7 neuroglobins using clustalx

Protein Phylogenies - Example

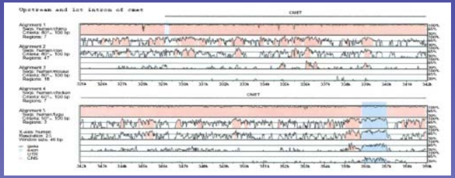


Motivation again

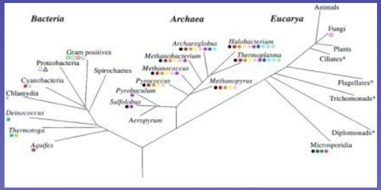
- Common structure, function or origin may be only weakly reflected in sequence – multiple comparisons may highlight weak signals
- Major uses:
  - Identify and represent protein families
  - Identify and represent conserved sequence or structure elements (e.g. domains)
  - Deduce evolutionary history

MSA : central role in biology

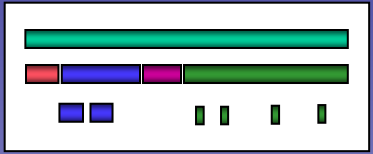
Comparative genomics



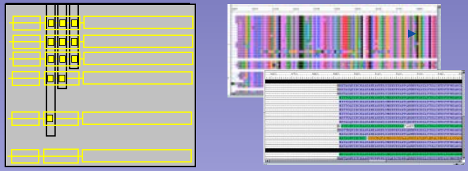
Phylogenetic studies



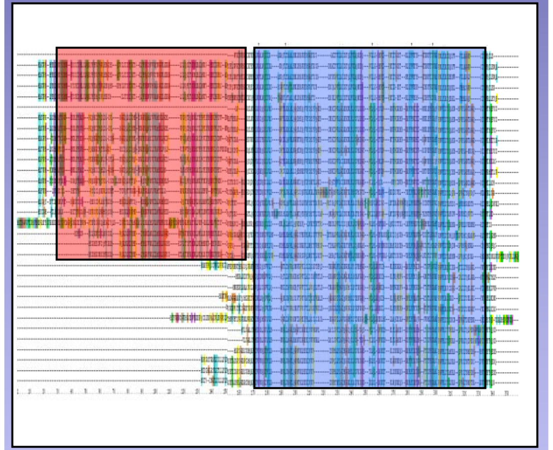
Hierarchical function annotation: homologs, domains, motifs



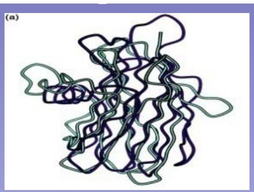
Gene identification, validation



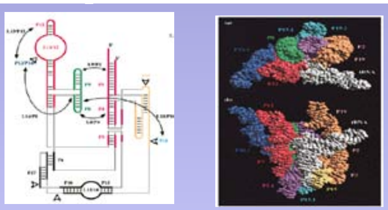
MSA



Structure comparison, modelling



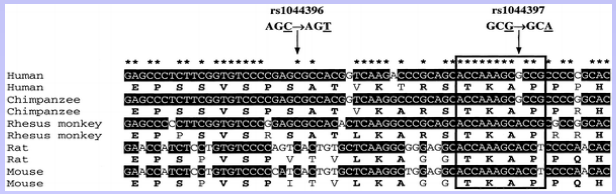
RNA sequence, structure, function



Interaction networks

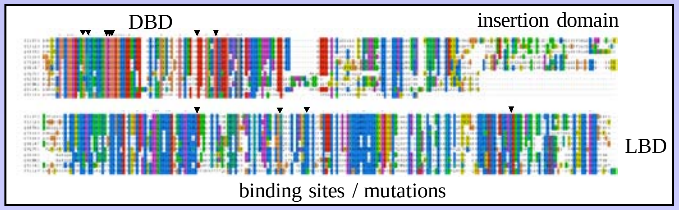
Tra1
yTAFII90
Ada2
Gcn5
Ada3
Ada1
yTAFII25
yTAFII68
SPT20
SPT7
SPT8
TBP
yTAFII60/yTAFII17
(dTAFII42/dTAFII62)
ySPT3

Human genetics, SNPs



```
                          rs1044396                      rs1044397
                          AGC→AGT                        GCG→GCA
                 ** **  ** * *******  * *   ***** *  *  ************ * **  ***
Human          GAGCCCTCTTCGGTGTCCCCGAGCGCCACGGTCAAGACCCGCAGCACCAAAGCGCCGCCCCCGCAC
Human           E  P  S  S  V  S  P  S  A  T  V  K  T  R  S  T  K  A  P  P  P  H
Chimpanzee     GAGCCCTCTTCGGTGTCCCCGAGCGCCACGGTCAAGGCCCGCAGCACCAAAGCGCCGCCCCGGCAC
Chimpanzee      E  P  S  S  V  S  P  S  A  T  V  K  A  R  S  T  K  A  P  P  R  H
Rhesus monkey  GAGCCCCCTTCGGTGTCCCGGAGCGCCACACTCAAGGCCCGCAGCACCAAAGCACCGCGCCGGCAC
Rhesus monkey   E  P  P  S  V  S  R  S  A  T  L  K  A  R  S  T  K  A  P  R  R  H
Rat            GAACCATCTCCTGTGTCCCCAGTCACTGTGCTCAAGGCGGCAGGCACCAAAGCACCTCCCCAACAC
Rat             E  P  S  P  V  S  P  V  T  V  L  K  A  G  G  T  K  A  P  P  Q  H
Mouse          GAACCATCTCCTGTGTCCCCATCACTGTGCTCAAGGCTGGAGGCACCAAAGCACCTCCCCAACAC
Mouse           E  P  S  P  V  S  P  I  T  V  L  K  A  G  G  T  K  A  P  P  Q  H
```

Therapeutics, drug design



Scoring alignments

- Given input seqs. S_1 , S_2 ,…, S_k find a multiple alignment of optimal score
- Scores preview:
  - Sum of pairs
  - Consensus
  - Tree

Sum of Pairs score

Def: Induced pairwise alignment

A pairwise alignment induced by the multiple alignment

Example:

```
x:  AC-GCGG-C
y:  AC-GC-GAG
z:  GCCGC-GAG
```

Induces:

```
x: ACGCGG-C;  x: AC-GCGG-C;  y: AC-GCGAG
y: ACGC-GAC;  z: GCCGC-GAG;  z: GCCGCGAG
```

$$S(M) = \Sigma_{k<l} \ \sigma(S'_k, S'_l)$$

SOP Score Example

Consider the following alignment:

```
AC-CDB-
-C-ADBD
A-BCDAD
```

Scoring scheme:	match -	0
	mismatch/indel -	-1

SP score: -3 -5 -4 =-12

Alignments = Paths

- Align 3 sequences: ATGC, AATC,ATGC

	A	--	T	G	C
	A	A	T	--	C
	--	A	T	G	C

Alignment Paths

0	1	1	2	3	4
	A	--	T	G	C

x coordinate

	A	A	T	--	C

	--	A	T	G	C

Alignment Paths

- Align 3 sequences: ATGC, AATC,ATGC

0	1	1	2	3	4	*x* coordinate
	A	--	T	G	C	
0	1	2	3	3	4	*y* coordinate
	A	A	T	--	C	

	--	A	T	G	C

-

Alignment Paths

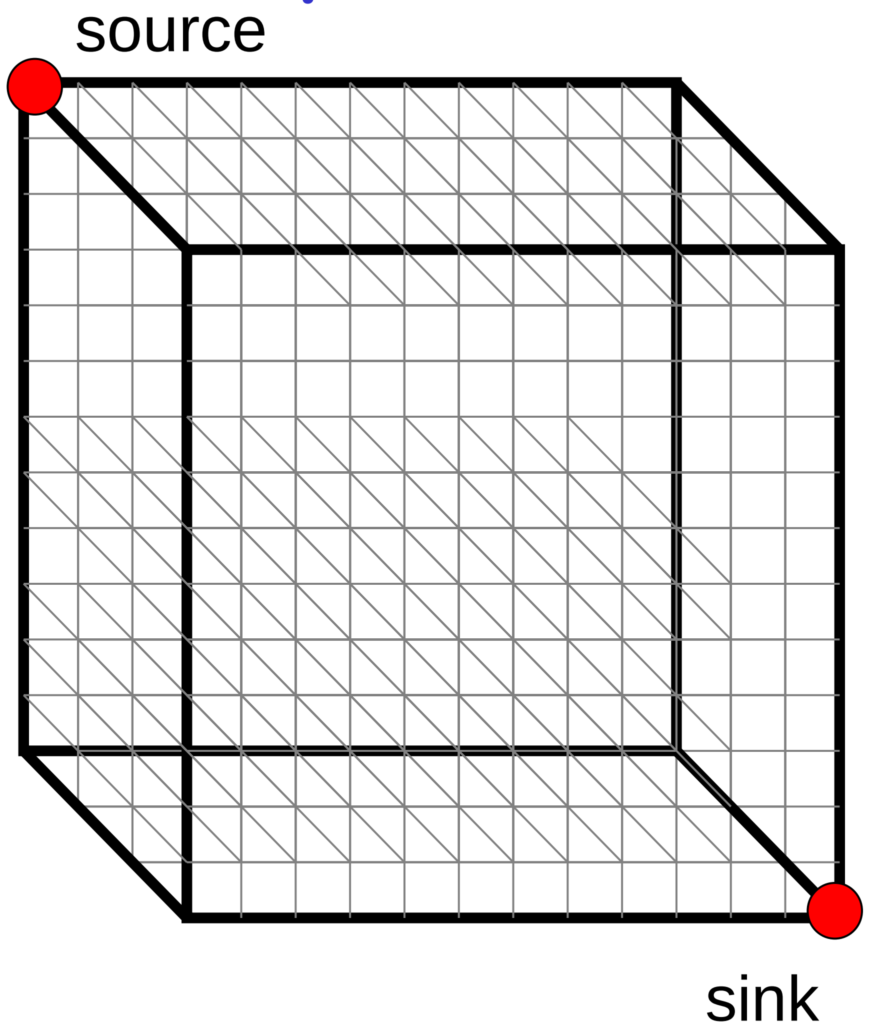
0	1	1	2	3	4	*x* coordinate
	A	--	T	G	C	
0	1	2	3	3	4	*y* coordinate
	A	A	T	--	C	
0	0	1	2	3	4	*z* coordinate
	--	A	T	G	C	

- Resulting path in *(x,y,z)* space:

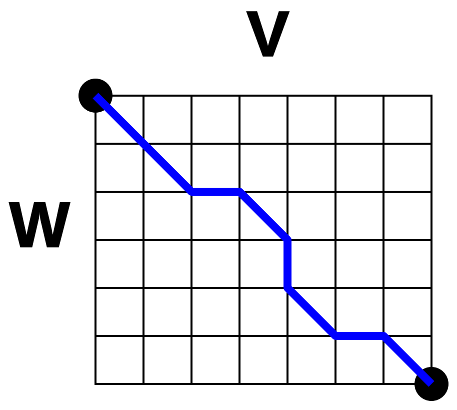
$(0,0,0)\rightarrow(1,1,0)\rightarrow(1,2,1)\rightarrow(2,3,2)\rightarrow(3,3,3)\rightarrow(4,4,4)$

Aligning Three Sequences

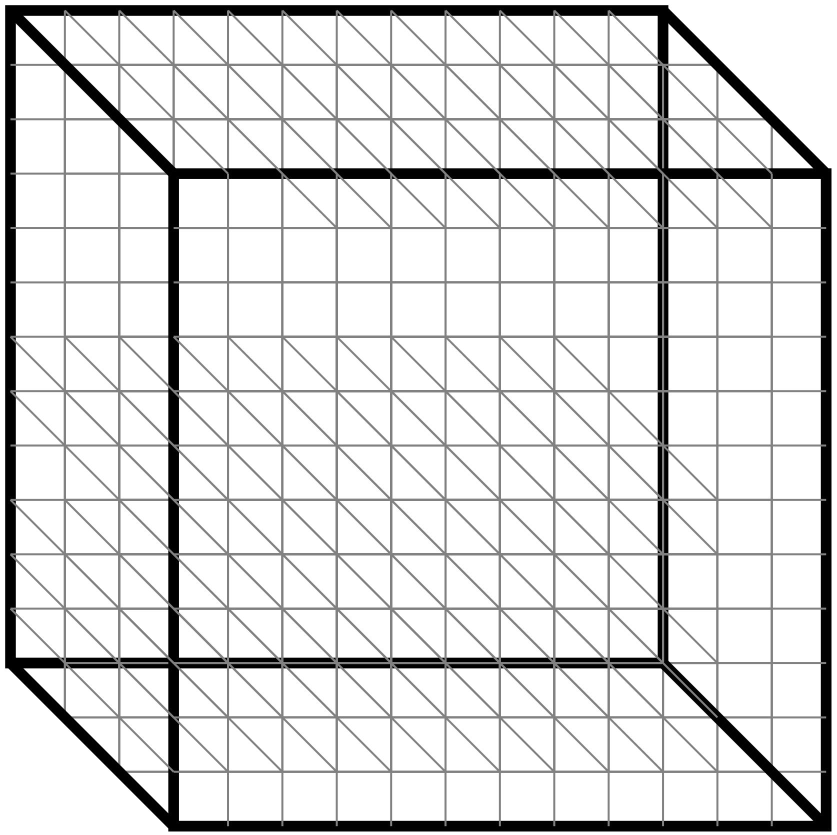
- Same strategy as aligning two sequences
- Use a 3-D "Manhattan Cube", with each axis representing a sequence to align
- For global alignments, go from source to sink



2-D vs 3-D Alignment Grid

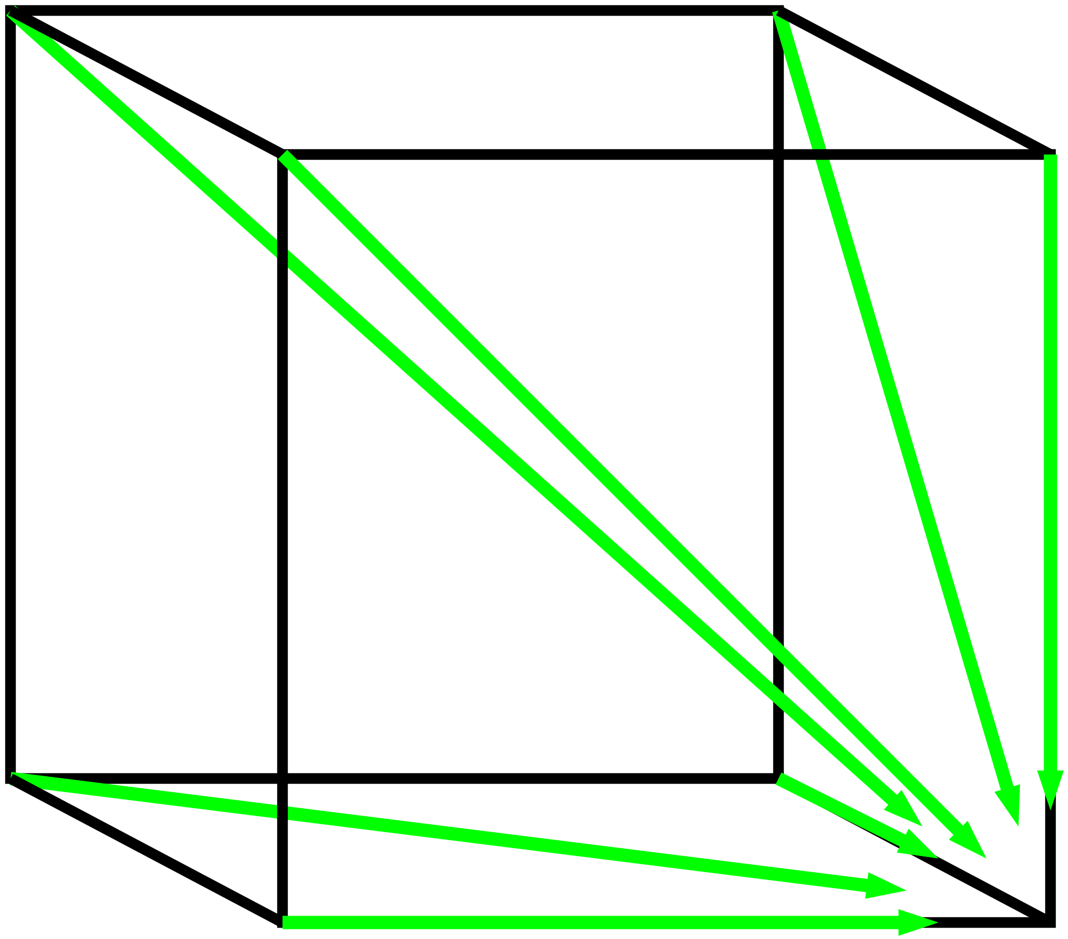


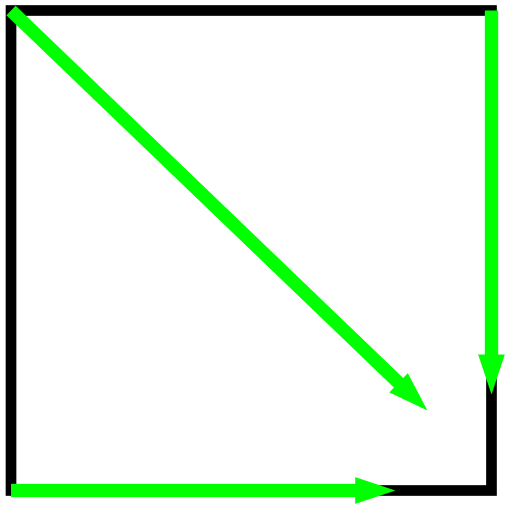
2-D edit graph



3-D edit graph

2-D cell versus 2-D Alignment Cell

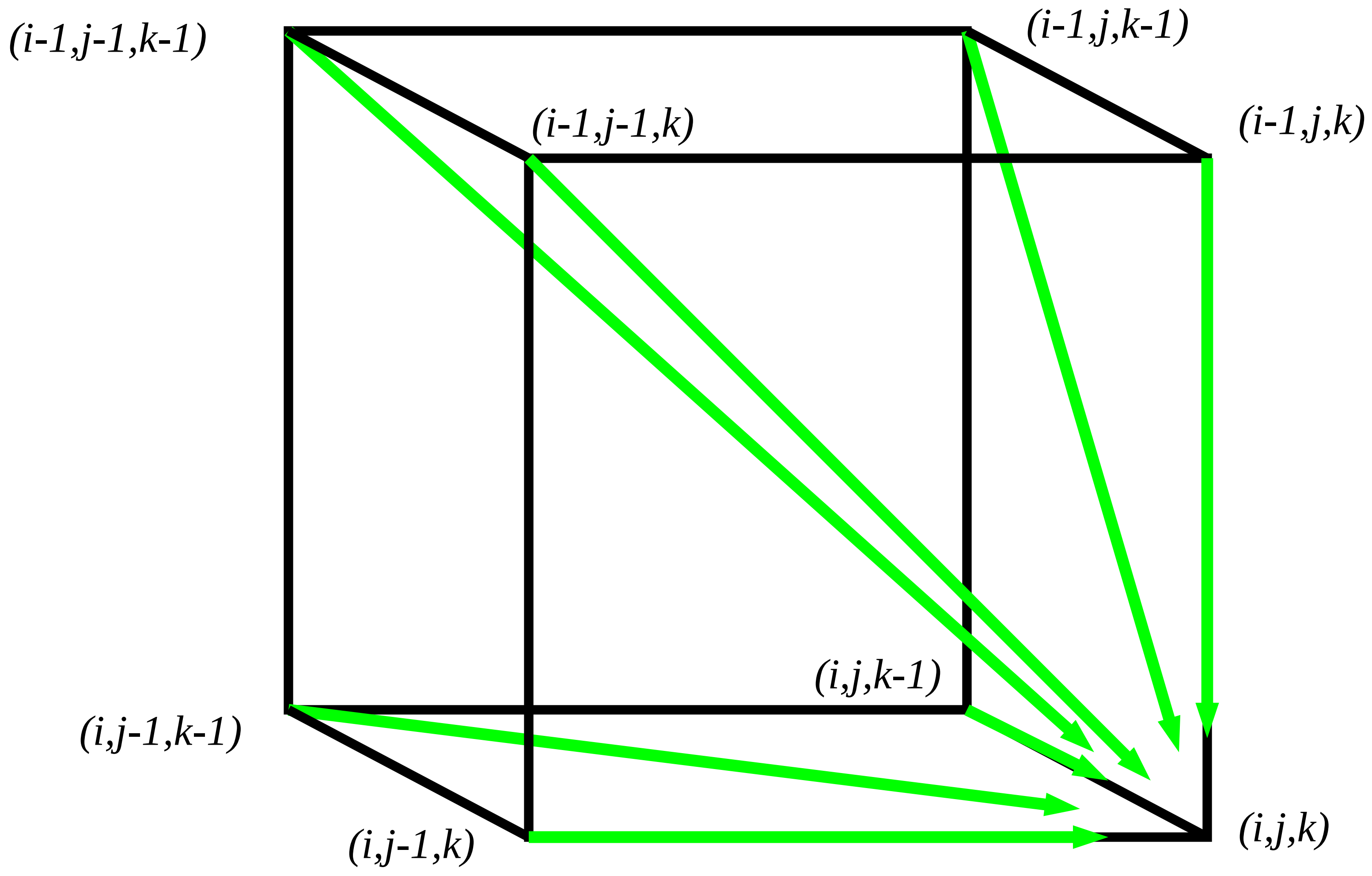




In **2-D**, 3 edges in each unit square

In **3-D**, 7 edges in each unit cube

Architecture of 3-D Alignment Cell



Multiple Alignment: Dynamic Programming

- $$s_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) & \text{cube diagonal: no indels} \\ s_{i-1,j-1,k} + \delta(v_i, w_j, \_) & \\ s_{i-1,j,k-1} + \delta(v_i, \_, u_k) & \text{face diagonal: one indel} \\ s_{i,j-1,k-1} + \delta(\_, w_j, u_k) & \\ s_{i-1,j,k} + \delta(v_i, \_, \_) & \\ s_{i,j-1,k} + \delta(\_, w_j, \_) & \text{edge diagonal: two indels} \\ s_{i,j,k-1} + \delta(\_, \_, u_k) & \end{cases}$$

- $\delta(x, y, z)$ is an entry in the 3-D scoring matrix

Running Time

- For 3 sequences of length n, the run time is $O(n^3)$
- For k sequences, build a k-dimensional cube, with run time $O(2^k n^k)$ [another 2^k factor for affine gaps]
- Impractical for most realistic cases
- NP-hard (Elias'03 for general matrices)

Minimum cost – SOP

We use min cost instead of max score

→ Find alignment of minimal cost

Assumption (for the approx. algs to follow):

the cost function **δ** is a distance function

- $\delta(x,x) = 0$ (also for gaps)
- $\delta(x,y) = \delta(y,x) \geq 0$
- $\delta(x,y) + \delta(y,z) \geq \delta(x,z)$ (triangle inequality)

Forward Dynamic Programming

- An alternative approach to DP for pairwise (and multiple) alignment:
- D(v) – opt value of path source→v
- p(w) – best-yet solution of path source→w
- When D(v) is computed, send its value forward on the arcs from v→w: p(w)=min{p(w),D(v)+cost(v,w)}
- Once p(w) has been updated by all incoming edges – that value is optimal; set as D(w)

Forward Dynamic Programming (2)

- Maintain a queue of nodes whose D is not set yet
- For the node w at the head of the queue: Set D(w)←p(w) and remove
- ∀ out-neighbor x of w – update p; if x is not in the queue – add it at the end
  - Breaking ties lexicographically
  - Only x-s with some forward transmission are added to the queue
- Same complexity as the regular (backwards) DP

Faster DP Algorithm for MultiAlign

Carillo-Lipman 88

- Use forward DP.
- We'll demonstrate on three sequences
- $f_{12}(i,j)$ = opt pairwise alignment score of suffixes $S_1(i+1,..n_1)$, $S_2(j+1,..n_2)$, etc.
- Key idea: if $\exists$ a known soln of cost z, if

$D(i,j,k)+ f_{12}(i,j) + f_{13}(i,k) +f_{23}(j,k) > z$

$\rightarrow$Do not send $D(i,j,k)$ forward

- Guarantees opt soln – no improved time bound, but often saves a lot in practice.

An approximation algorithm
Gusfield 93

- Compute a *center* string, minimizing the sum of pairwise distances to the other strings
- Use it as a "pivot" for the alignment

D(S,T) - cost of minimum global alignment between S and T

The Center Star algorithm

Input: Γ - set of k strings $S_1,\dots,S_k$.

1. Find the string $S^*\in$ Γ (center) that minimizes $\sum_{S\in\Gamma\setminus\{S^*\}} D(S^*,S)$
2. Denote $S_1=S^*$ and the rest of the strings as $S_2,\dots,S_k$
3. Iteratively add $S_2,\dots,S_k$ to the alignment as follows:
   a. Suppose $S_1,\dots,S_{i-1}$ are already aligned as $S'_1,\dots,S'_{i-1}$
   b. Optimally align S_i to S'_1 to produce S'_i and S''_1 aligned
   c. Adjust $S'_2,\dots,S'_{i-1}$ by adding spaces where spaces were added to S''_1
   d. Replace S'_1 by S''_1

Inheriting gaps

```
x:   AGAC
y:   ATGA          ← center
z:   ATGGA
```

1:

```
y: ATGA-
x: A-GAC
```

2:

```
y: ATG-A
Z: ATGGA
```

```
y: ATG-A-
x: A-G-AC
Z: ATGGA-
```

Running time

- Choosing S_1 – execute DP for all sequence-pairs - $O(k^2n^2)$
- Adding S_i to the alignment - execute DP for S_i , S'_1 - $O(i \cdot n^2)$.

  (In the i^{th} stage the length of S'_1 can be up-to $i \cdot n$)

$$\sum_{i=1}^{k-1} O(i \cdot n^2) = \underbrace{O(k^2 n^2)}_{\text{total complexity}}$$

Approximation ratio

- ***M**** - An optimal alignment
- ***M*** - The alignment produced by this algorithm
- **d(i,j)** - The distance M induces on the pair S_i, S_j
-

$$v(M) = \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} d(i,j) = 2 \sum_{i<j} d(i,j)$$

- recall $D(S,T)$ – min cost of alignment between S and T

For all i: $\mathbf{d(1,i)=D(S_1,S_i)}$

(we perform optimal alignment between S'_1 and S_i and $\delta(\text{-},\text{-}) = 0$)

Approximation ratio (2)

Triangle inequality + symmetry

$$v(M) = \sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d(i,j) \le \sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} \left(d(1,i)+d(1,j)\right)$$

$$= 2(k-1)\sum_{l=2}^{k} d(1,l) = 2(k-1)\sum_{l=2}^{k} D(S_1,S_l)$$

$$\frac{v(M)}{v(M^*)} \le \frac{2(k-1)}{k} \le 2$$

$$v(M^*) = \sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} d^*(i,j) \ge \sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k} D(S_i,S_j) \ge$$

$$\ge \sum_{i=1}^{k}\sum_{j=2}^{k} D(S_1,S_j) = k\sum_{j=2}^{k} D(S_1,S_j)$$

Definition of S_1:

$$\forall i : \sum_{j=2}^{k} D(S_1,S_j) \le \sum_{\substack{j=1\\ j\neq i}}^{k} D(S_i,S_j)$$

Theorem (Gusfield 93)

- We have proved:
- The center star algorithm is a polynomial algorithm that guarantees a solution at most twice the optimum.

- "a 2-approximation"
- "an approximation ratio of 2"

Steiner/consensus string

- Input: Γ - set of k strings $S_1, \ldots, S_k$.
- D(X,Y) – score of aligning X, Y.
- S – arbitrary sequence (unrelated to Γ)

- The consensus error of S relative to Γ:
  $E(S) = \Sigma_{i \leq k} D(S, S_i)$

- S* is an optimal Steiner string for Γ if it minimizes E(S)
- Different objective function – linear no of terms
- No direct relation to multialign! (for now)

Optimal Steiner String: Approximation

Thm: Assume D satisfies triangle ineq. Then $\exists S \in \Gamma$ that gives a 2-approximation.

Pf: Pick $S \in \Gamma$

$$E(S) = \sum_{S \neq S_i} D(S, S_i) \leq \sum_{S \neq S_i} \left(D(S, S^*) + D(S^*, S_i) \right)$$

$$= (k-2)D(S,S^*) + D(S,S^*) + \sum_{S_i \neq S} D(S^*, S_i)$$

$$= (k-2)D(S,S^*) + E(S^*)$$

Pick $S \in \Gamma$ closest to S* (not constructively) $\Rightarrow \dfrac{E(S)}{E(S^*)} \leq \dfrac{(k-2)}{k} + 1 < 2$

$$E(S^*) = \sum_{S_i \in \Gamma} D(S^*, S_i) \geq k \cdot D(S, S^*)$$

Consensus string from MA

- The consensus string of a MA is obtained by taking the most frequent character in each position

```
S*:  AC-GC-GAG
x:   AC-GCGG-C
y:   AC-GC-GAG
z:   GCCGA-GAG
u:   AC-T-GGCA
v:   -CAGT-GAG
w:   AC-GC-GAG
```

Alignment error: $S(M) = \Sigma_k \ \sigma(S'_k, S^*)$

The opt consensus MA of the set of input sequences Γ: MA with least alignment error

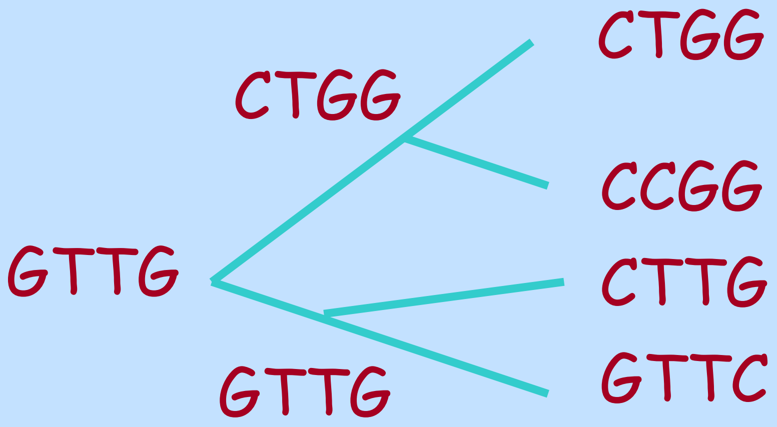
Consensus multiple alignment

Thm: opt soln of consensus MA = Steiner string (up to spaces)

-

Tree Multiple Alignment

- Input: Tree T, a string for each leaf
- Phylogenetic (also Tree) alignment for T: Assignment of a string to each internal node



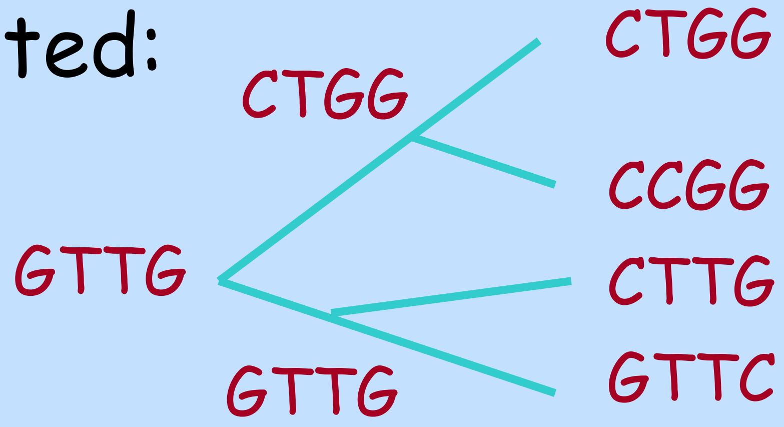
- Score – (weighted) sum of scores along edges
- Goal: find tree alignment of optimal score
- Consensus = tree alignment where T is a star

Tree MA – complexity

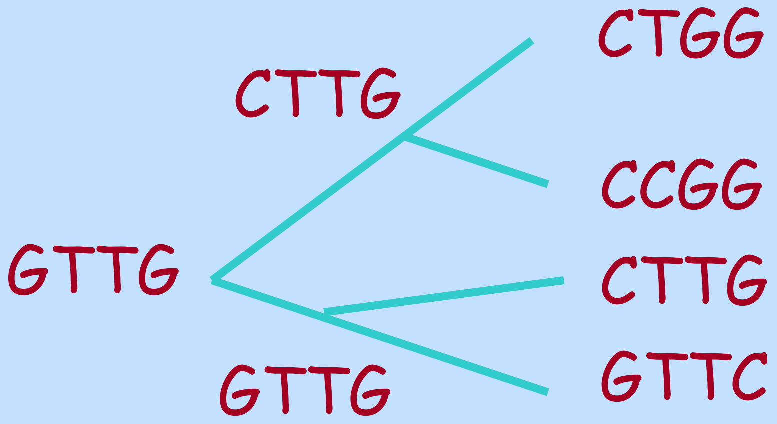
- NP-hard
- Poly time approximations:
  - 2-approximation
  - Better approximation with more time (PTAS)

Lifted alignment

- The seq. label at every internal node is lifted from one of its children
- Lifted:

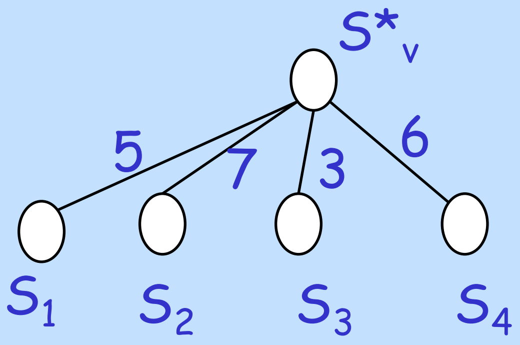


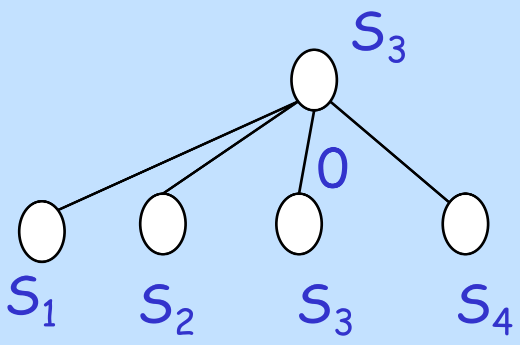
- Not lifted



A 2-approximation to Tree MSA [Jiang, Wang, Lawler 1996]

- Assumes triangle inequality
- (non constructively) Transform an optimal tree T^* to a lifted alignment T^L:
  - At each internal node v, assign seq. of a child that is closest to the optimal label of v





- Claim: T^L has at most twice the distance of T^*

Pf sketch: $cost(T^L) \le 2\ cost(T^*)$

- In T^L, take $e=(v,w)$, $v=Pa(w)$ with labels S_j for v, S_i for w, $S_i \neq S_j$
  - $D(S_j,S_i) \le D(S_j,S^*_v) + D(S^*_v,S_i) \le 2D(S_i,S^*_v)$ (why?)
  - Path P_e from leaf labeled S_i up to v has cost:
    - $D(S_j,S_i)$ in T^L
    - At least $D(S^*_v,S_i)$ in T^*
- Paths $\{P_e\}$ are edge disjoint and cover all nonzero edges in T^L

Dynamic Programming alg for optimal lifted alignment

- d(v,S) – distance of the best lifted alignment of T_v s.t. string S is assigned to node v

$$\mathbf{d(v,S) = \Sigma_w \ \min_T [D(S,T)+d(w,T)]}$$

here w – child of v, T – string at a leaf of T_w

- Complexity: k = no of leaves, tot length N
  - Compute all pairwise leaf distances in $O(N^2)$
  - Computation per internal node: $O(k^2)$
  - → $O(N^2+k^3)$ (can do $O(N^2+k^2)$)

Wrapping up lifted alignment

- ∃ a lifted alignment that is ≤ 2 OPT
- We can find the min cost lifted alignment in poly time
- That alignment is also ≤ 2 OPT

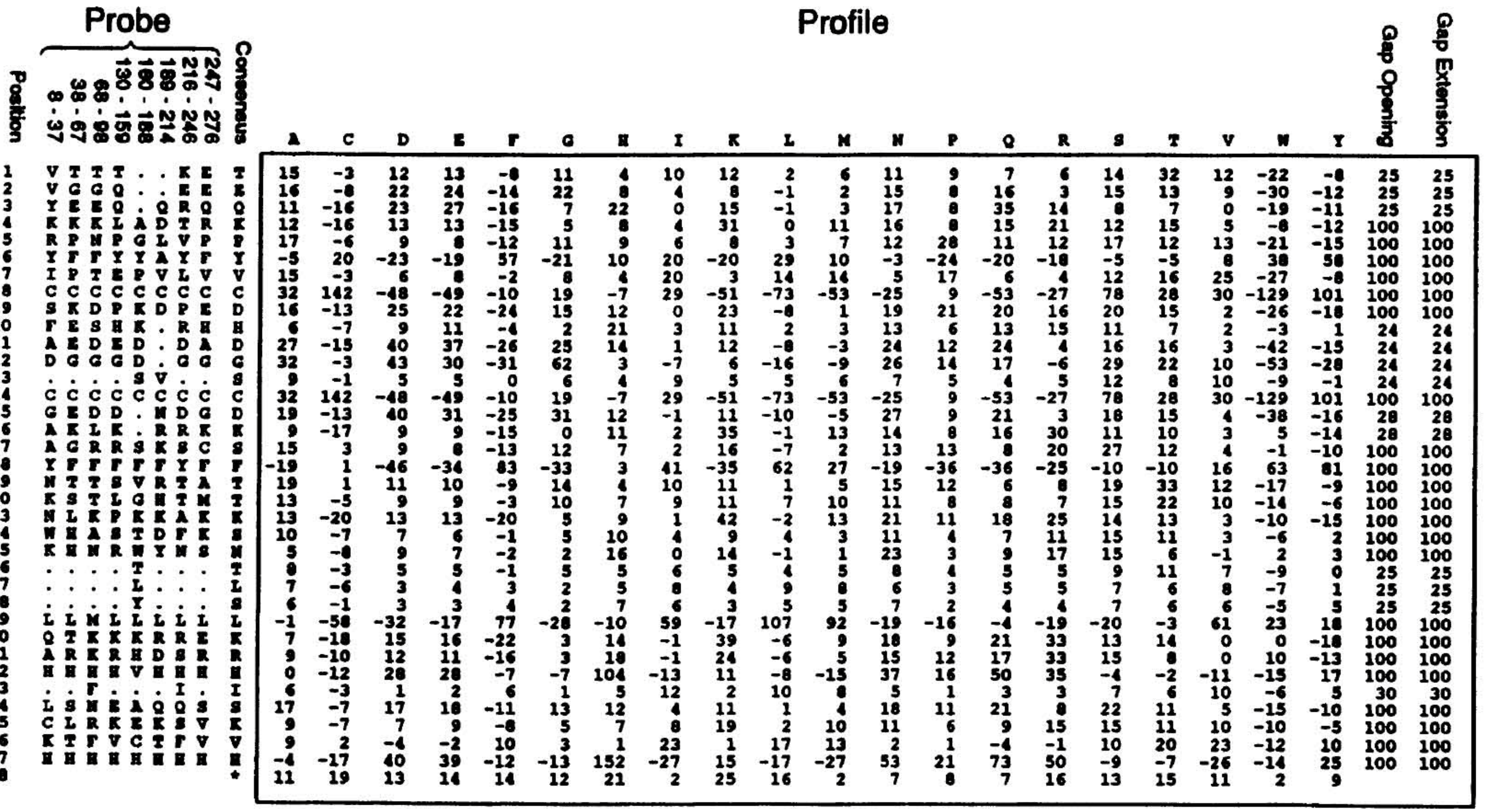
- ➔Thm: lifted alignment alg gives a poly-time 2-approximation to Tree Alignment

Profile Representation of MA

-	A	G	G	C	T	A	T	C	A	C	C	T	G
T	A	G	–	C	T	A	C	C	A	-	-	-	G
C	A	G	–	C	T	A	C	C	A	-	-	-	G
C	A	G	–	C	T	A	T	C	A	C	–	G	G
C	A	G	–	C	T	A	T	C	G	C	–	G	G

A		1					1			.8				
C	.6				1			.4	1		.6	.2		
G			1	.2						.2			.4	1
T	.2					1		.6					.2	
-	.2			.8							.4	.8	.4	

- Alternatively, use log odds:
- $p_i(a)$ = fraction of a's in col i
- $p(a)$ = fraction of a's overall
- $\log p_i(a)/p(a)$



Position	Probe								Consensus	Profile																					
	8 - 37	38 - 67	68 - 98	130 - 159	160 - 188	189 - 214	216 - 246	247 - 276		A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Gap Opening	Gap Extension
1	V	T	T	T	.	.	K	E	T	15	-3	12	13	-8	11	4	10	12	2	6	11	9	7	6	14	32	12	-22	-8	25	25
2	V	G	G	Q	.	.	E	E	E	16	-8	22	24	-14	22	8	4	8	-1	2	15	8	16	3	15	13	9	-30	-12	25	25
3	Y	E	E	Q	.	Q	R	Q	Q	11	-16	23	27	-16	7	22	0	15	-1	3	17	8	35	14	8	7	0	-19	-11	25	25
4	K	K	K	L	A	D	T	R	K	12	-16	13	13	-15	5	8	4	31	0	11	16	8	15	21	12	15	5	-8	-12	100	100
5	R	P	N	P	G	L	V	P	P	17	-6	9	8	-12	11	9	6	8	3	7	12	28	11	12	17	12	13	-21	-15	100	100
6	Y	F	F	Y	Y	A	Y	F	Y	-5	20	-23	-19	57	-21	10	20	-20	29	10	-3	-24	-20	-18	-5	-5	8	38	58	100	100
7	I	P	T	E	P	V	L	V	V	15	-3	6	8	-2	8	4	20	3	14	14	5	17	6	4	12	16	25	-27	-8	100	100
8	C	C	C	C	C	C	C	C	C	32	142	-48	-49	-10	19	-7	29	-51	-73	-53	-25	9	-53	-27	78	28	30	-129	101	100	100
9	S	K	D	P	K	D	P	E	D	16	-13	25	22	-24	15	12	0	23	-8	1	19	21	20	16	20	15	2	-26	-18	100	100
10	F	E	S	H	K	.	R	H	H	6	-7	9	11	-4	2	21	3	11	2	3	13	6	13	15	11	7	2	-3	1	24	24
11	A	E	D	E	D	.	D	A	D	27	-15	40	37	-26	25	14	1	12	-8	-3	24	12	24	4	16	16	3	-42	-15	24	24
12	D	G	G	G	D	.	G	G	G	32	-3	43	30	-31	62	3	-7	6	-16	-9	26	14	17	-6	29	22	10	-53	-28	24	24
13	.	.	.	.	S	V	.	.	S	9	-1	5	5	0	6	4	9	5	5	6	7	5	4	5	12	8	10	-9	-1	24	24
14	C	C	C	C	C	C	C	C	C	32	142	-48	-49	-10	19	-7	29	-51	-73	-53	-25	9	-53	-27	78	28	30	-129	101	100	100
15	G	E	D	D	.	N	D	G	D	19	-13	40	31	-25	31	12	-1	11	-10	-5	27	9	21	3	18	15	4	-38	-16	28	28
16	A	K	L	K	.	R	R	K	K	9	-17	9	9	-15	0	11	2	35	-1	13	14	8	16	30	11	10	3	5	-14	28	28
17	A	G	R	R	S	K	S	C	S	15	3	9	8	-13	12	7	2	16	-7	2	13	13	8	20	27	12	4	-1	-10	100	100
18	Y	F	F	F	F	F	Y	F	F	-19	1	-46	-34	83	-33	3	41	-35	62	27	-19	-36	-36	-25	-10	-10	16	63	81	100	100
19	N	T	T	S	V	R	T	A	T	19	1	11	10	-9	14	4	10	11	1	5	15	12	6	8	19	33	12	-17	-9	100	100
20	K	S	T	L	G	H	T	M	T	13	-5	9	9	-3	10	7	9	11	7	10	11	8	8	7	15	22	10	-14	-6	100	100
23	N	L	K	P	K	K	A	K	K	13	-20	13	13	-20	5	9	1	42	-2	13	21	11	18	25	14	13	3	-10	-15	100	100
24	W	H	A	S	T	D	F	K	S	10	-7	7	6	-1	5	10	4	9	4	3	11	4	7	11	15	11	3	-6	2	100	100
25	K	H	N	R	W	Y	N	S	N	5	-8	9	7	-2	2	16	0	14	-1	1	23	3	9	17	15	6	-1	2	3	100	100
26	.	.	.	.	T	.	.	.	T	8	-3	5	5	-1	5	5	6	5	4	5	8	4	5	5	9	11	7	-9	0	25	25
27	.	.	.	.	L	.	.	.	L	7	-6	3	4	3	2	5	8	4	9	8	6	3	5	5	7	6	8	-7	1	25	25
28	.	.	.	.	Y	.	.	.	S	6	-1	3	3	4	2	7	6	3	5	5	7	2	4	4	7	6	6	-5	5	25	25
29	L	L	M	L	L	L	L	L	L	-1	-58	-32	-17	77	-28	-10	59	-17	107	92	-19	-16	-4	-19	-20	-3	61	23	18	100	100
30	Q	T	K	K	K	R	R	E	K	7	-18	15	16	-22	3	14	-1	39	-6	9	18	9	21	33	13	14	0	0	-18	100	100
31	A	R	K	R	H	D	S	R	R	9	-10	12	11	-16	3	18	-1	24	-6	5	15	12	17	33	15	8	0	10	-13	100	100
32	H	H	H	H	V	H	H	H	H	0	-12	28	28	-7	-7	104	-13	11	-8	-15	37	16	50	35	-4	-2	-11	-15	17	100	100
33	.	.	F	.	.	.	I	.	I	6	-3	1	2	6	1	5	12	2	10	8	5	1	3	3	7	6	10	-6	5	30	30
34	L	S	H	E	A	Q	Q	S	S	17	-7	17	18	-11	13	12	4	11	1	4	18	11	21	8	22	11	5	-15	-10	100	100
35	C	L	R	K	E	K	S	V	K	9	-7	7	9	-8	5	7	8	19	2	10	11	6	9	15	15	11	10	-10	-5	100	100
36	K	T	F	V	C	T	F	V	V	9	2	-4	-2	10	3	1	23	1	17	13	2	1	-4	-1	10	20	23	-12	10	100	100
37	H	H	H	H	H	H	H	H	H	-4	-17	40	39	-12	-13	152	-27	15	-17	-27	53	21	73	50	-9	-7	-26	-14	25	100	100
38									*	11	19	13	14	14	12	21	2	25	16	2	7	8	7	16	13	15	11	2	9		

FIG. 2. Profile of the *Xenopus laevis* transcription factor TFIIIA zinc finger. The eight repeats of the zinc finger sequence that form the probe are shown descending vertically at the left, labeled with the positions where they occur in the complete sequence. Insertions made to align the sequences are shown as periods. The profile calculated by **PROFILEMAKE** is shown in the box. The rows correspond to the positions in the aligned sequences, and the columns contain the score for each possible amino acid residue when aligned at that position. The position-specific gap penalties are given in the two right-hand columns. The consensus sequence is shown immediately to the left of the box, and represents the highest scoring column at each row in the profile. In other words, the consensus residue is the amino acid that would receive the highest score when aligned with that position in the

Aligning a sequence to a profile

- Key in pairwise alignment is scoring two positions x,y: $\sigma(x,y)$
- For a letter x and a column C in a profile, $\sigma(x,C)$=value of x in col. C
- Invent a score for $\sigma(x,-)$
- Run the DP alg for pairwise alignment

Aligning alignments

- Given two alignments, how can we align them?
- Hint: use DP on the corresponding profiles.

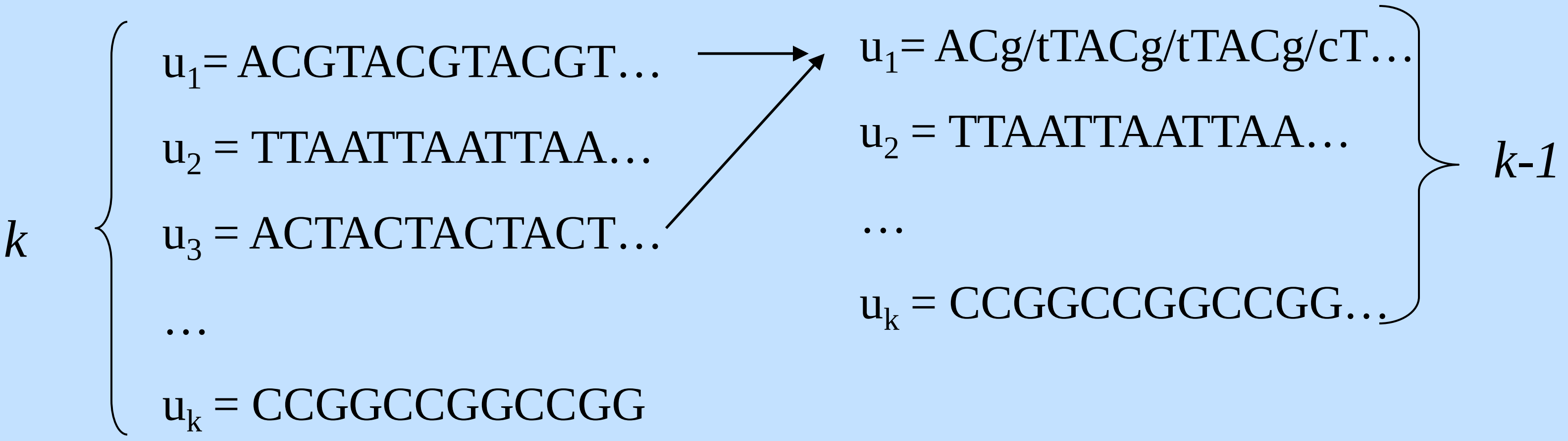
```
x GGGCACTGCAT                          x GGGCACTGCAT
y GGTTACGTC--    Alignment 1           y GGTTACGTC--
z GGGAACTGCAG                          z GGGAACTGCAG
                                       w GGACGTACC--
w GGACGTACC--    Alignment 2           v GGACCT-----
v GGACCT-----
```

Profile-profile scoring

- Fix a position in the alignment
  - p_i – prob (i in 1st profile); q_i – in 2nd profile
- Expected score: $\Sigma_{ij}\, p_i\, q_j\, s_{ij}$
- Euclidean distance
- Pearson correlation
- KL-divergence (relative entropy)
- …

Multiple Alignment: Greedy Heuristic

- Choose most similar pair of sequences and combine into a profile , thereby reducing alignment of *k* sequences to an alignment of *k-1* sequences/profiles. **Repeat**

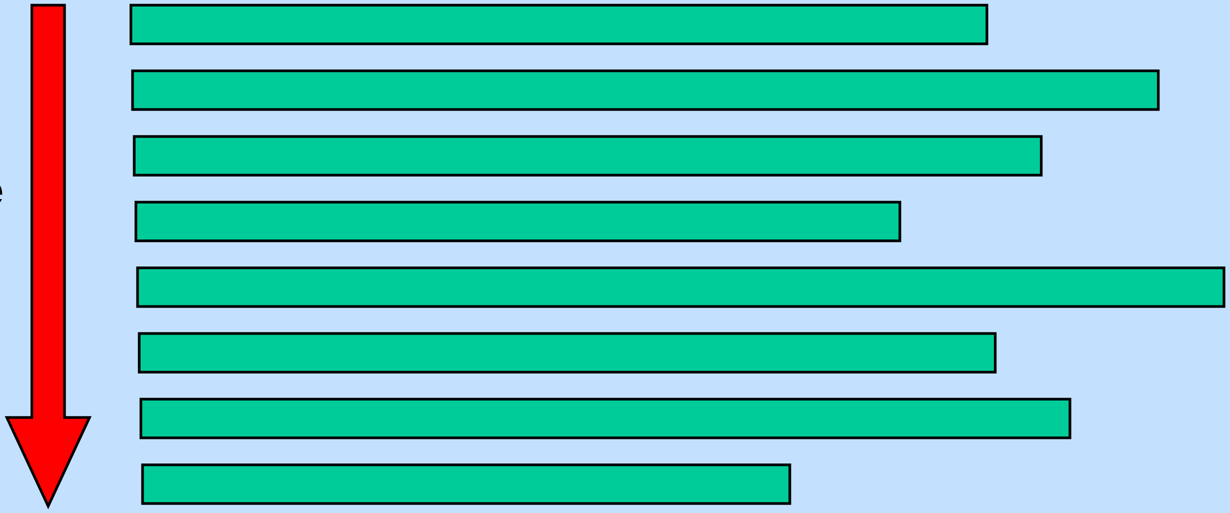


Progressive Alignment

- *A* variation of greedy algorithm with a somewhat more intelligent strategy for choosing the order of alignments.

Progressive alignment

Align sequences (pairwise) in some (greedy) order

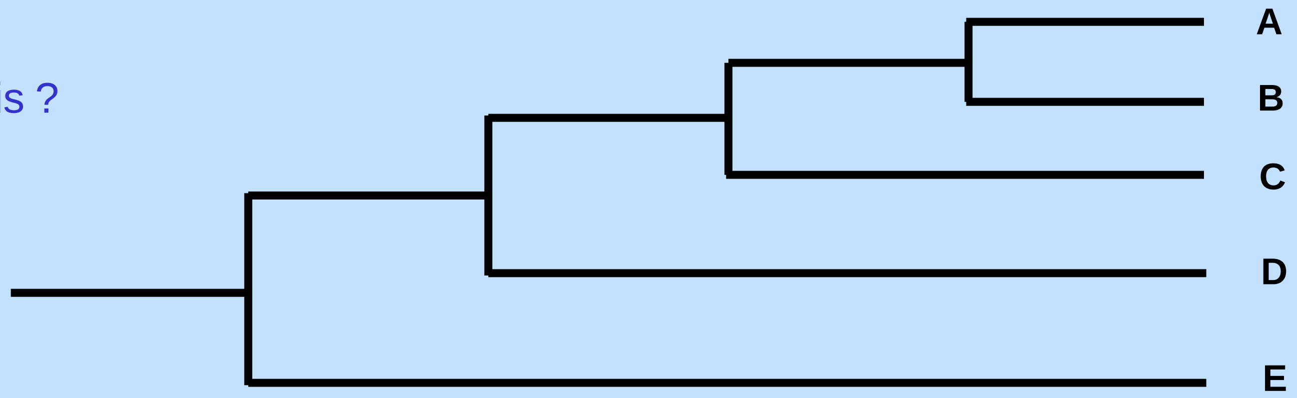


Decisions

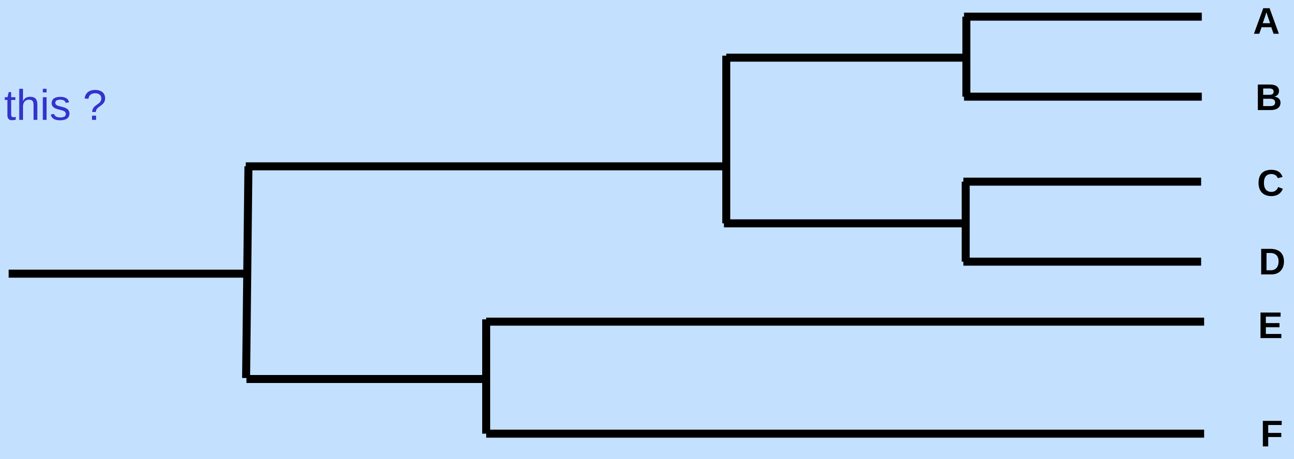
(1) Order of alignments
(2) Alignment of sequence to group (only), or allow group to group
(3) Method of alignment, and scoring function

Guide tree

this ?



or this ?



Feng & Doolittle (1987)

Overview

(1) Calculate distances between all pairs of sequences by pairwise alignment; convert raw alignment scores to approximate pairwise "distances".

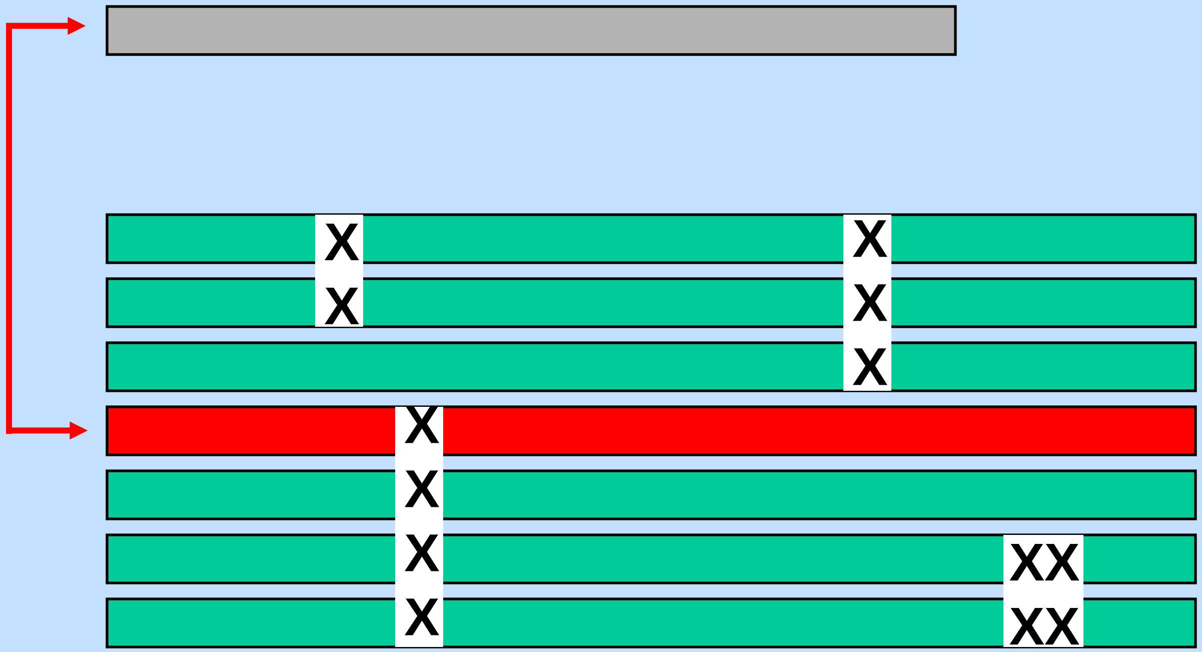
(2) Construct guide tree from the distance matrix by using a tree building (clustering) alg.

(3) Starting from first node added to the tree, align the child nodes (which may be two sequences, a sequence and an alignment, or two alignments). Repeat for all other nodes in the order that they were added to tree, until all sequences have been aligned.

Feng & Doolittle (1987)

sequence-to-group

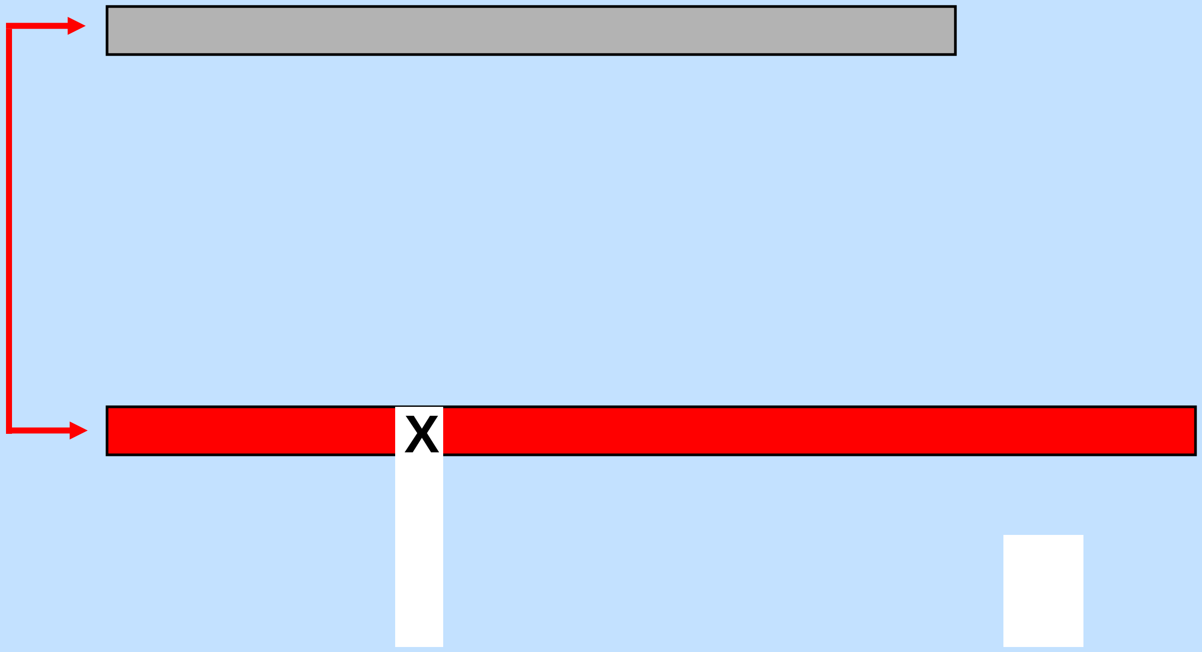
Best pairwise alignment determines alignment to group



Feng & Doolittle (1987)

sequence-to-group

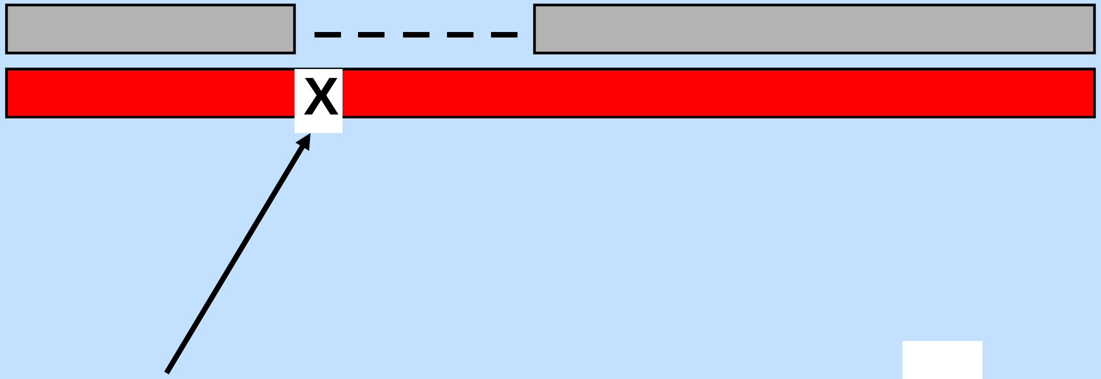
Best pairwise alignment determines alignment to group



Feng & Doolittle (1987)

sequence-to-group

Best pairwise alignment determines alignment to group

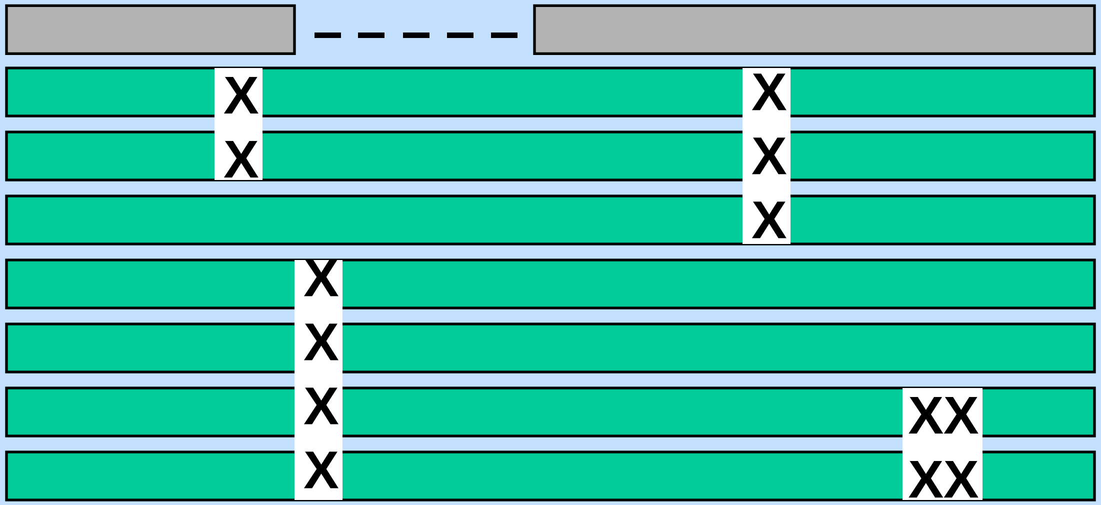


A space in the grey seq in this column is encouraged because it has no cost

Feng & Doolittle (1987)

sequence-to-group

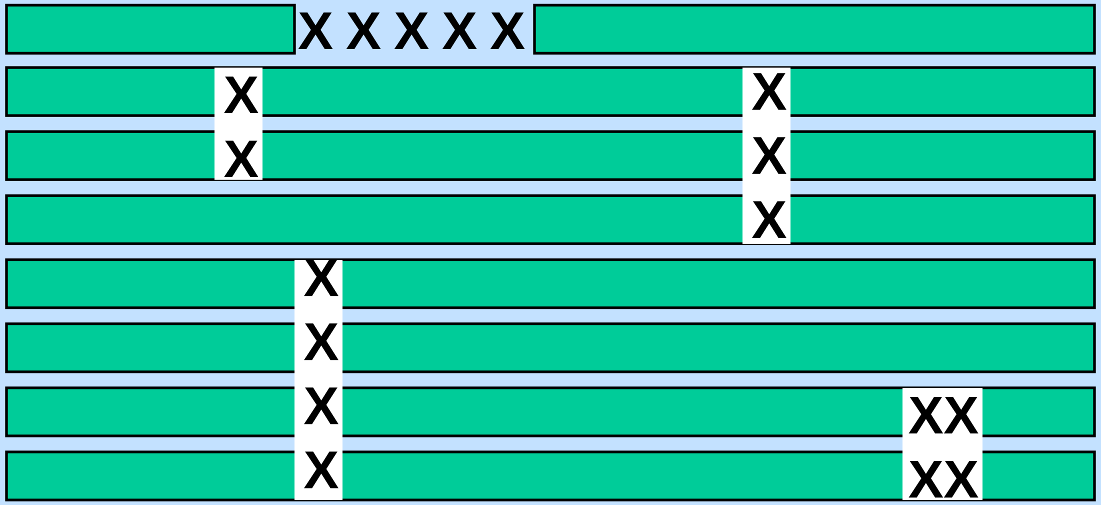
Best pairwise alignment determines alignment to group



Feng & Doolittle (1987)

sequence-to-group

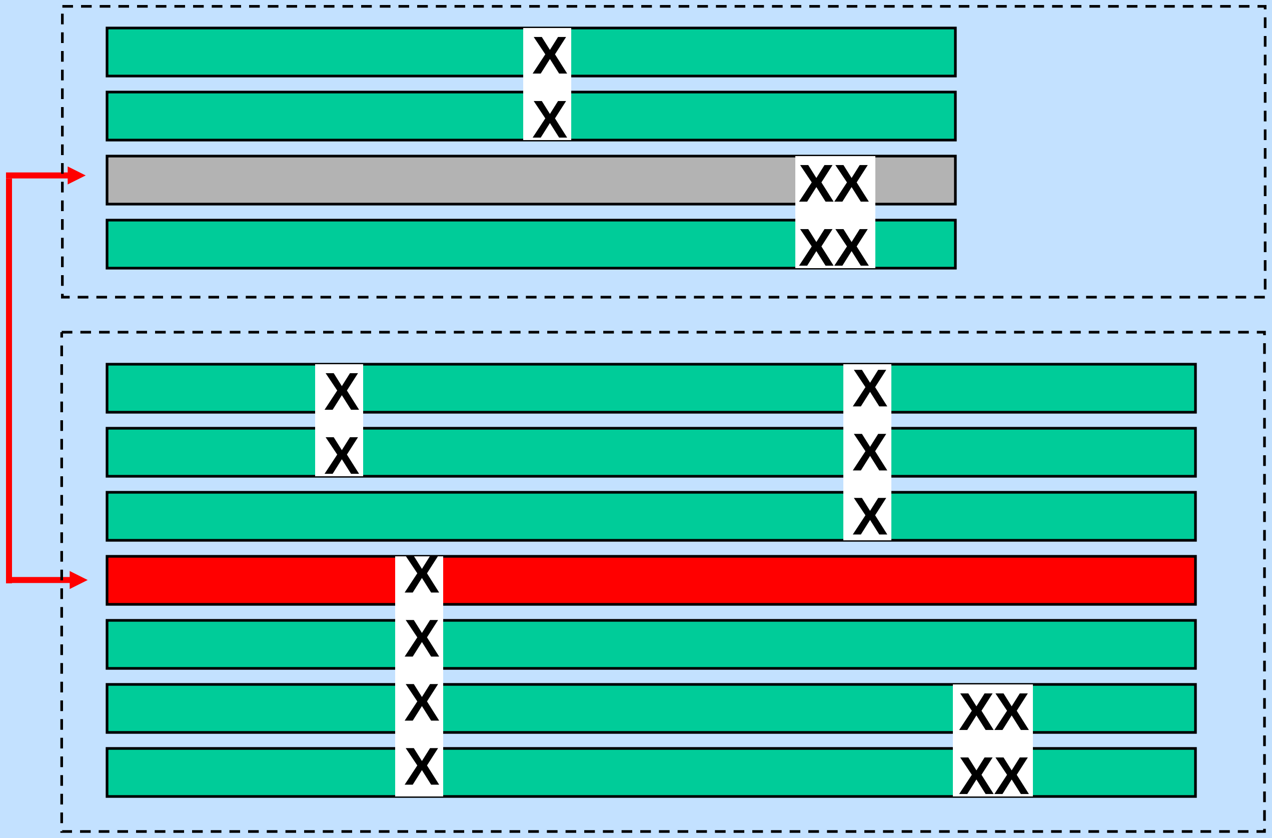
Best pairwise alignment determines alignment to group



Feng & Doolittle (1987)

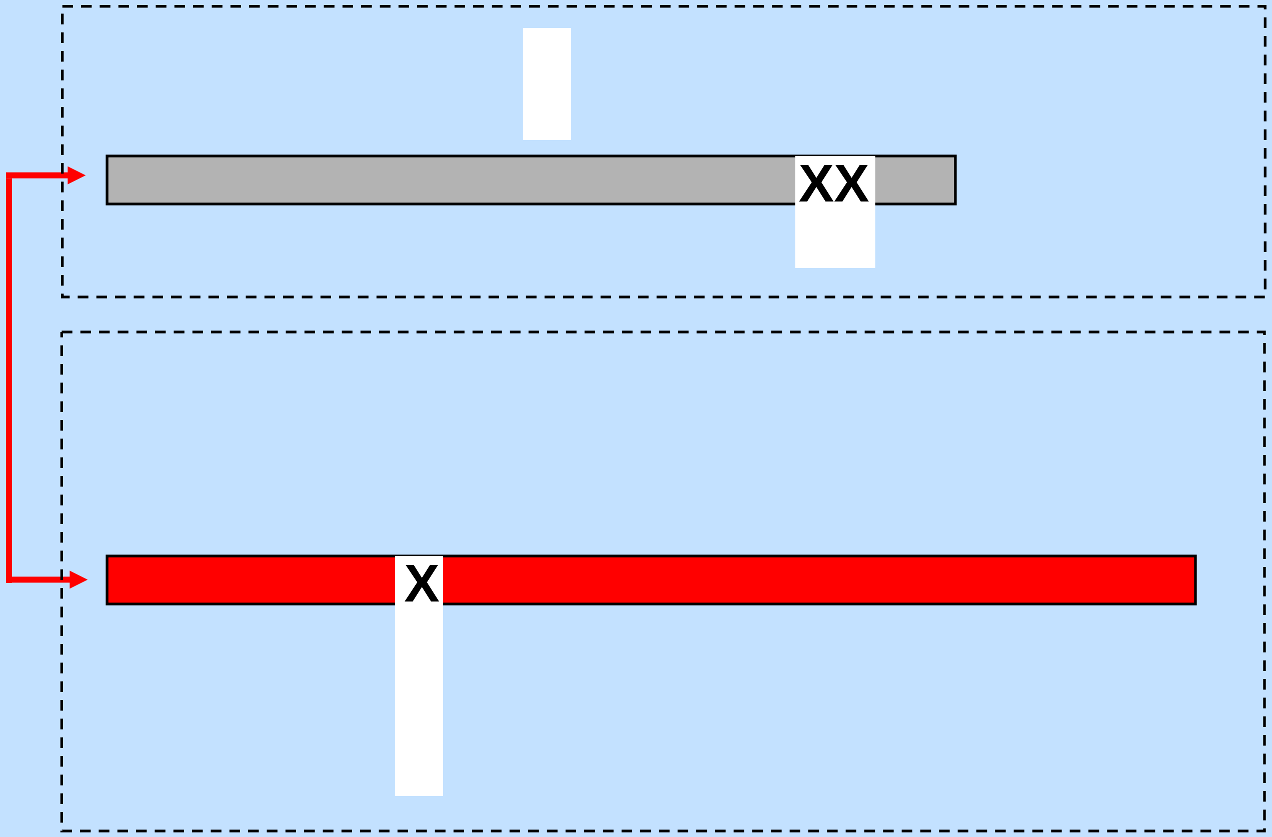
group-to-group

Best pairwise alignment determines alignment of groups



Feng & Doolittle (1987)

group-to-group

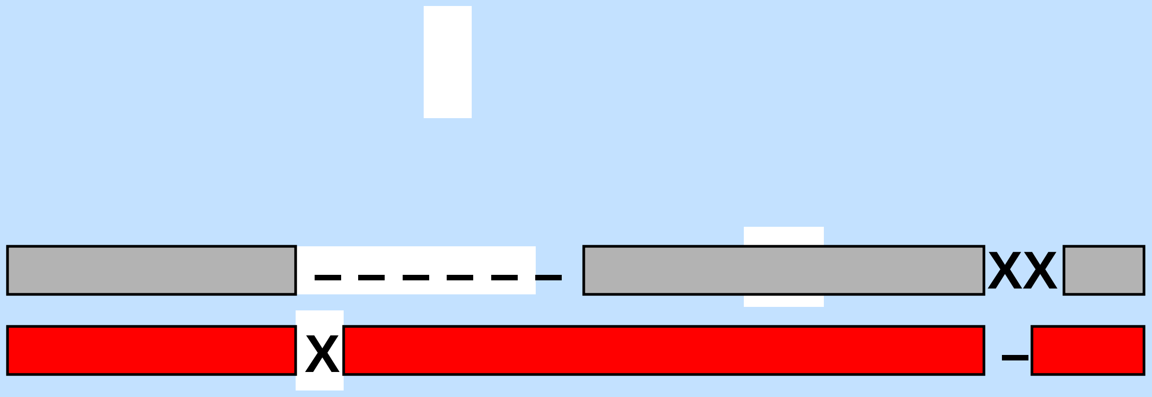


Best pairwise alignment determines alignment of groups

Feng & Doolittle (1987)

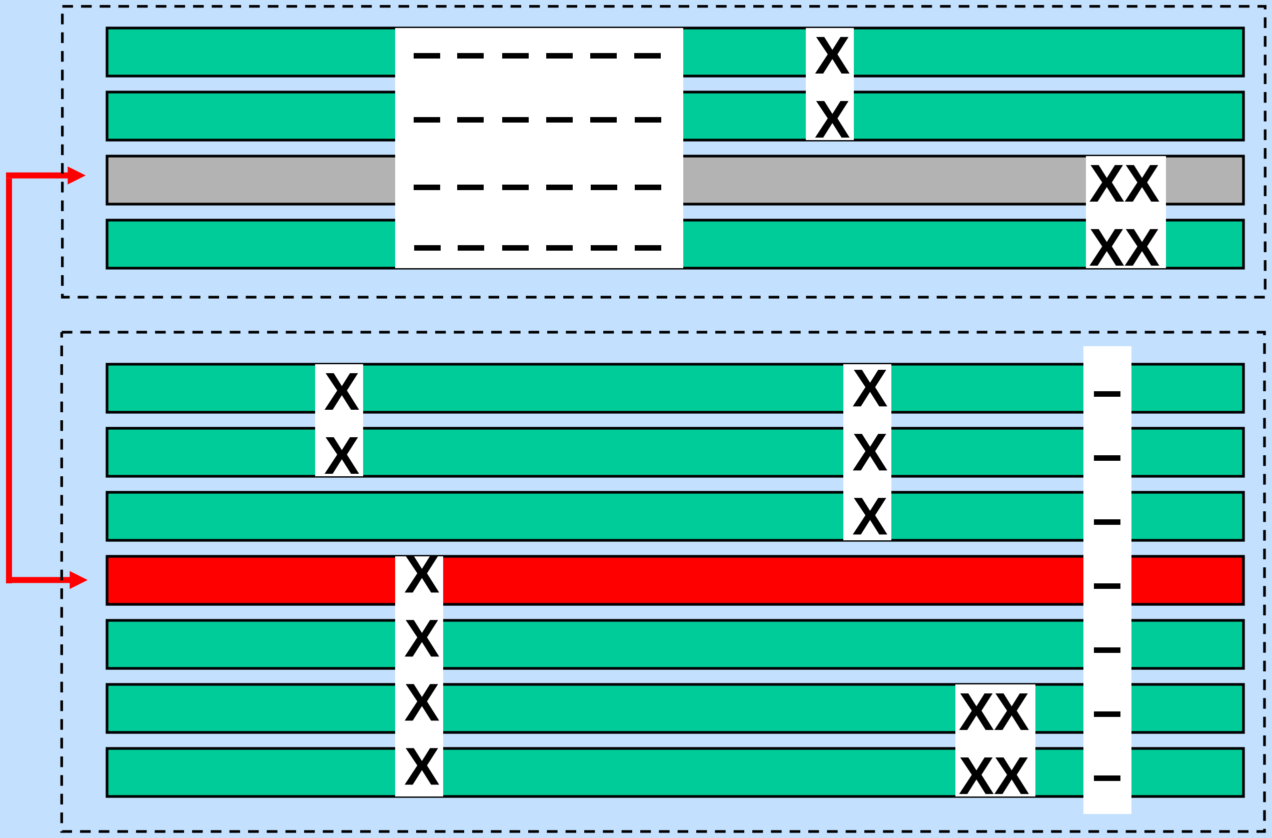
group-to-group

Best pairwise alignment determines alignment of groups



Feng & Doolittle (1987)

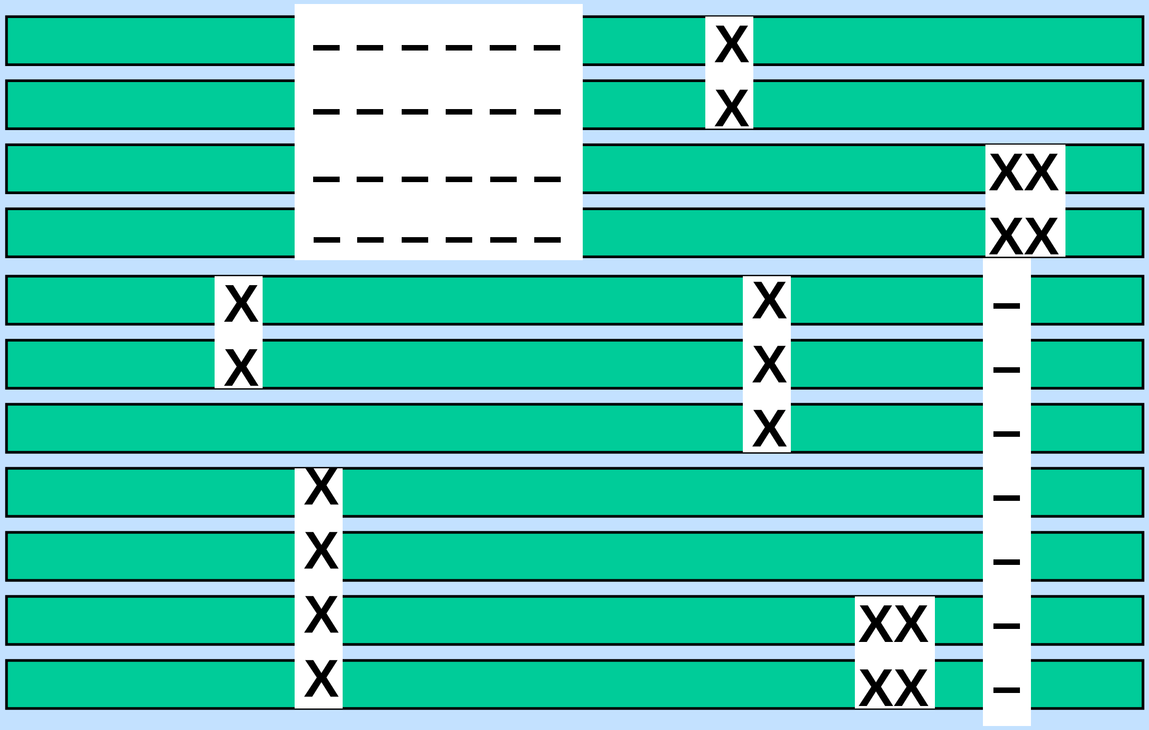
group-to-group



Best pairwise alignment determines alignment of groups

Feng & Doolittle (1987)

group-to-group

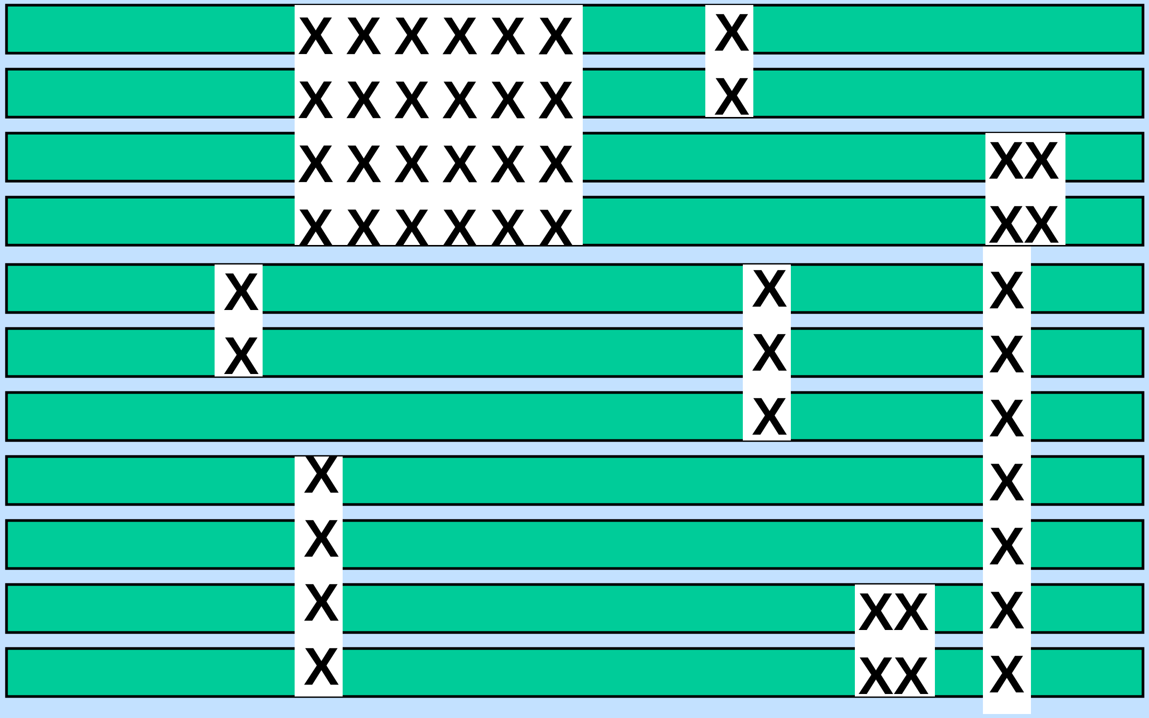


Best pairwise alignment determines alignment of groups

Feng & Doolittle (1987)

group-to-group

Best pairwise alignment determines alignment of groups



Feng & Doolittle (1987)

"Once a gap, always a gap".

Encourages gaps to occur in same columns in subsequent alignments.

This early version of progressive alignment is based on pairwise alignments. Later methods use profile-profile alignments.

ClustalW Thompson, Higgins, Gibson 94

- Popular multiple alignment tool today
- Three-step process

  1.) Construct pairwise alignments

  2.) Build guide tree

  3.) Progressive alignment guided by the tree

Step 1: Pairwise Alignment

- Aligns each pair of sequences, giving a similarity matrix
- Similarity = exact matches / sequence length (percent identity)

	v_1	v_2	v_3	v_4
v_1	-			
v_2	.17	-		
v_3	.87	.28	-	
v_4	.59	.33	.62	-

(.17 means 17 % identical)

Step 2: Guide Tree

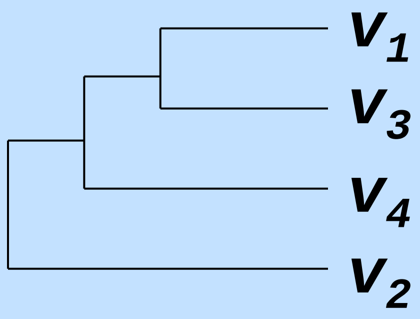
- Use the similarity method to create a guide tree by applying some clustering method*

- Guide tree roughly reflects evolutionary relations

  - *ClustalW uses the neighbor-joining method (to be described later in the course)

Step 2: Guide Tree (cont'd)

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & - & & & \\ v_2 & .17 & - & & \\ v_3 & .87 & .28 & - & \\ v_4 & .59 & .33 & .62 & - \end{array}$$



Calculate:

$v_{1,3}$ = alignment (v_1, v_3)

$v_{1,3,4}$ = alignment$((v_{1,3}),v_4)$

$v_{1,2,3,4}$ = alignment$((v_{1,3,4}),v_2)$

Step 3: Progressive Alignment

- Start by aligning the two most similar sequences
- Using the guide tree, add in the most similar pair (seq-seq, seq-prof or prof-prof)
- Insert gaps as necessary
- Many ad-hoc rules: weighting, different matrices, special gap scores….

```
FOS_RAT         PEEMSVTS-LDLTGGLPEATTPESEEAFTLPLLNDPEPK-PSLEPVKNISNMELKAEPFD
FOS_MOUSE       PEEMSVAS-LDLTGGLPEASTPESEEAFTLPLLNDPEPK-PSLEPVKSISNVELKAEPFD
FOS_CHICK       SEELAAATALDLG----APSPAAAEEAFALPLMTEAPPAVPPKEPSG--SGLELKAEPFD
FOSB_MOUSE      PGPGPLAEVRDLPG-----STSAKEDGFGWLLPPPPPPP-----------------LPFQ
FOSB_HUMAN      PGPGPLAEVRDLPG-----SAPAKEDGFSWLLPPPPPPP-----------------LPFQ
                .   . :   ** .     :..  *:.*   *   . *                   **:
```

Dots and stars show how well-conserved a column is.

Multiple Alignment: History

1975 Sankoff
Formulated multiple alignment problem and gave dynamic programming solution
1988 Carrillo-Lipman
Branch and Bound approach for MSA
1990 Feng-Doolittle
Progressive alignment
1994 Thompson-Higgins-Gibson-ClustalW ← ***>40K citations!***
Most popular multiple alignment program
1998 Morgenstern et al.-DIALIGN
Segment-based multiple alignment
2000 Notredame-Higgins-Heringa-T-coffee
Using the library of pairwise alignments
2004 MUSCLE
2005 ProbCons
2011 Clustal Omega
……

Still a lot to be done!