Multiple Alignment

Outline

- Problem definition
- Can we use Dynamic Programming to solve MSA?
- Progressive Alignment
- ClustalW
- Scoring Multiple Alignments
  - Entropy
  - Sum of Pairs (SP) Score

Multiple Alignment versus Pairwise Alignment

- **Up until now we have only tried to align two sequences.**

Multiple Alignment versus Pairwise Alignment

- **Up until now we have only tried to align two sequences.**
- **What about more than two? And what for?**

Multiple Alignment versus 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

- One of the most essential tools in molecular biology
  - Finding highly conserved subregions or embedded patterns of a set of biological sequences
    - Conserved regions usually are key functional regions, prime targets for drug developments
  - Estimation of evolutionary distance between sequences
  - Prediction of protein secondary/tertiary structure
- Practically useful methods only since 1987 (D. Sankoff)
  - Before 1987 they were constructed by hand
  - Dynamic programming is expensive

Multiple Sequence Alignment (MSA)

- What is multiple sequence alignment?
- Given *k* sequences:

```
VTISCTGSSSNIGAGNHVKWYQQLPG
VTISCTGTSSNIGSITVNWYQQLPG
LRLSCSSSGFIFSSYAMYWVRQAPG
LSLTCTVSGTSFDDYYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG
ATLVCLISDFYPGAVTVAWKADS
AALGCLVKDYFPEPVTVSWNSG
VSLTCLVKGFYPSDIAVEWESNG
```

Multiple Sequence Alignment (MSA)

- An MSA of these sequences:

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--
```

Multiple Sequence Alignment (MSA)

- An MSA of these sequences:

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--
```

Conserved residues

Multiple Sequence Alignment (MSA)

- An MSA of these sequences:

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--
```

Conserved regions

Multiple Sequence Alignment (MSA)

- An MSA of these sequences:

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--
```

Patterns? Positions 1 and 3 are hydrophobic residues

Multiple Sequence Alignment (MSA)

- An MSA of these sequences:

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--
```

Conserved residues, regions, patterns

MSA Warnings

- MSA algorithms work under the assumption that they are aligning related sequences
- They will align ANYTHING they are given, even if unrelated
- If it just "looks wrong" it probably is

Generalizing the Notion of Pairwise Alignment

- Alignment of 2 sequences is represented as a 2-row matrix
- In a similar way, we represent alignment of 3 sequences as a 3-row matrix

```
A T _ G C G _
A _ C G T _ A
A T C A C _ A
```

- Score: more conserved columns, better alignment

Alignments = Paths in *k* dimensional grids

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

x coordinate

Alignment Paths

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

x coordinate

y coordinate

-

Alignment Paths

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

x coordinate

y coordinate

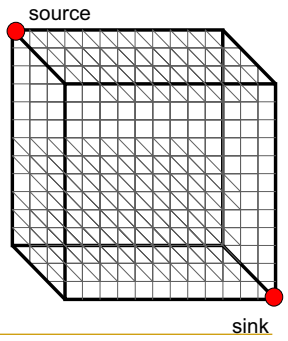
z coordinate

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

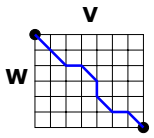
(0,0,0)→(1,1,0)→(1,2,1) →(2,3,2) →(3,3,3) →(4,4,4)

Aligning Three Sequences

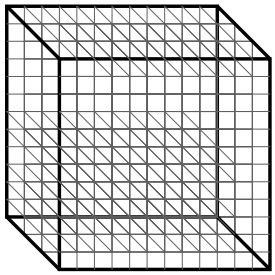
- Same strategy as aligning two sequences
- Use a 3-D matrix, with each axis representing a sequence to align
- For global alignments, go from source to sink



2-D vs 3-D Alignment Grid

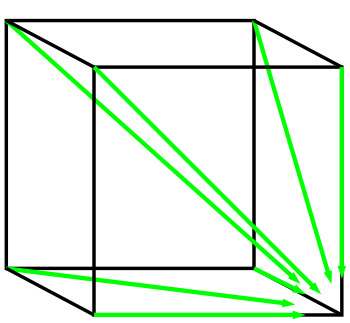


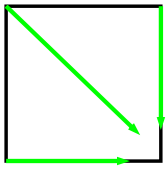
2-D alignment matrix



3-D alignment matrix

2-D cell versus 3-D Alignment Cell

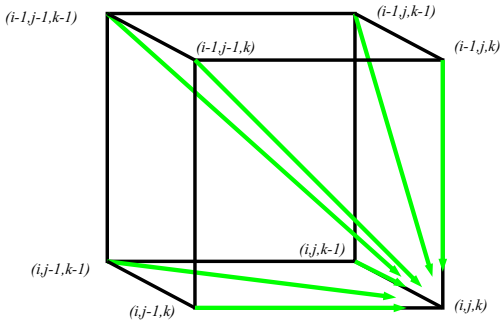




In **2-D**, 3 edges in each unit square

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

Architecture of 3-D Alignment Cell



Multiple Alignment: Dynamic Programming

- $s_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) \\ s_{i-1,j-1,k} + \delta(v_i, w_j, \_) \\ s_{i-1,j,k-1} + \delta(v_i, \_, u_k) \\ s_{i,j-1,k-1} + \delta(\_, w_j, u_k) \\ s_{i-1,j,k} + \delta(v_i, \_, \_) \\ s_{i,j-1,k} + \delta(\_, w_j, \_) \\ s_{i,j,k-1} + \delta(\_, \_, u_k) \end{cases}$

  cube diagonal: no indels

  face diagonal: one indel

  edge diagonal: two indels

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

Multiple Alignment: Running Time

- For 3 sequences of length n, the run time is $7n^3$; $O(n^3)$
- For k sequences, build a k-dimensional matrix, with run time $(2^k-1)(n^k)$; $O(2^k n^k)$
- Conclusion: dynamic programming approach for alignment between two sequences is easily extended to k sequences but it is impractical due to exponential running time

Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments

```
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
```

Reverse Problem: Constructing Multiple Alignment from Pairwise Alignments

Given 3 arbitrary pairwise alignments:

```
x: ACGCTGG-C;   x: AC-GCTGG-C;   y: AC-GC-GAG
y: ACGC--GAC;   z: GCCGCA-GAG;   z: GCCGCAGAG
```

can we construct a multiple alignment that induces them?

Reverse Problem: Constructing Multiple Alignment from Pairwise Alignments

Given 3 arbitrary pairwise alignments:

```
x: ACGCTGG-C;   x: AC-GCTGG-C;   y: AC-GC-GAG
y: ACGC--GAC;   z: GCCGCA-GAG;   z: GCCGCAGAG
```

can we construct a multiple alignment that induces them?

NOT ALWAYS

Pairwise alignments may be inconsistent

Inferring Multiple Alignment from Pairwise Alignments

- From an optimal multiple alignment, we can infer pairwise alignments between all pairs of sequences, but they are not necessarily optimal
- It is difficult to infer a "good" multiple alignment from optimal pairwise alignments between all sequences

Combining Optimal Pairwise Alignments into Multiple Alignment

Can combine pairwise alignments into multiple alignment

Can ***not*** combine pairwise alignments into multiple alignment

Consensus String of a Multiple Alignment

```
- 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

C A G - C T A T C A C - G G
Consensus
String: C A G C T A T C A C G G
```

- The *consensus string* S_M derived from multiple alignment *M* is the concatenation of the consensus characters for each column of *M*.
  - The *consensus character* for column *i* is the character that minimizes the summed distance to it from all the characters in column *i*. (i.e., if match and mismatch scores are equal for all symbols, the majority symbol is the consensus character)

Profile Representation of Multiple Alignment

```
          -  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
```

Profile Representation of Multiple Alignment

```
          -  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
```

Earlier, we were aligning a **sequence against a sequence**

Can we align a **sequence against a profile?**

Can we align a **profile against a profile?**

Aligning alignments

- Given two alignments, can we align them?

```
x GGGCACTGCAT
y GGTTACGTC--     Alignment 1
z GGGAACTGCAG

w GGACGTACC--     Alignment 2
v GGACCT-----
```

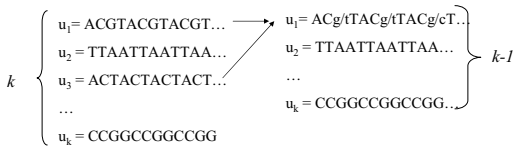
Aligning alignments

- Given two alignments, can we align them?
- Hint: use alignment of corresponding profiles

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

Multiple Alignment: Greedy Approach

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



Greedy Approach: Example

- Consider these 4 sequences

s1 GATTCA
s2 GTCTGA
s3 GATATT
s4 GTCAGC

Greedy Approach: Example (cont'd)

- There are $\binom{4}{2}$ = 6 possible alignments

```
s2  GTCTGA                 s1  GATTCA--
s4  GTCAGC (score = 2)     s4  G-T-CAGC(score = 0)

s1  GAT-TCA                s2  G-TCTGA
s2  G-TCTGA (score = 1)    s3  GATAT-T (score = -1)

s1  GAT-TCA                s3  GAT-ATT
s3  GATAT-T (score  = 1)   s4  G-TCAGC (score = -1)
```

Greedy Approach: Example (cont'd)

s_2 *and* s_4 are closest; combine:

```
s2  GTCTGA ]
s4  GTCAGC ]  s2,4 GTCt/aGa/cA
                (profile)
```

new set of 3 sequences:

s_1 GATTCA
s_3 GATATT
$s_{2,4}$ GTCt/aGa/c

Progressive Alignment

- *Progressive alignment* is a variation of greedy algorithm with a somewhat more intelligent strategy for choosing the order of alignments.
- Progressive alignment works well for close sequences, but deteriorates for distant sequences
  - Gaps in consensus string are permanent
  - Use profiles to compare sequences

Star alignment

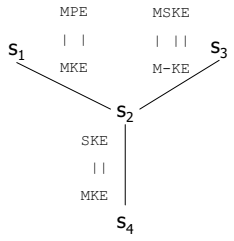
- Heuristic method for multiple sequence alignments
- Select a sequence *c* as the center of the star
- For each sequence $x_1, \ldots, x_k$ such that index $i \neq c$, perform a Needleman-Wunsch global alignment
- Aggregate alignments with the principle "once a gap, always a gap."

Choosing a center

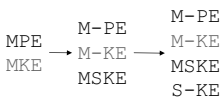
- Try them all and pick the one which is most similar to all of the sequences
- Let $S(x_i,x_j)$ be the optimal score between sequences x_i and x_j.
- Calculate all $O(k^2)$ alignments, and choose as x_c the sequence x_i that maximizes the following

$$\sum_{j \neq i} S(x_i,x_j)$$

Star alignment example



S_1: MPE
S_2: MKE
S_3: MSKE
S_4: SKE



Analysis

- Assuming all sequences have length n
- $O(k^2n^2)$ to calculate center
- Step i of iterative pairwise alignment takes $O((i \cdot n) \cdot n)$ time
  - two strings of length n and $i \cdot n$
- $O(k^2n^2)$ overall cost

ClustalW

- Most popular multiple alignment tool today
- 'W' stands for 'weighted' (different parts of alignment are weighted differently).
- Three-step process
  1.) Construct pairwise alignments
  2.) Build Guide Tree (by Neighbor Joining method)
  3.) Progressive Alignment guided by the tree
  - The sequences are aligned progressively according to the branching order in the guide tree

Step 1: Pairwise Alignment

- Aligns each sequence again each other 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	-

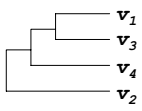
(.17 means 17 % identical)

Step 2: Guide Tree

- Create Guide Tree using the similarity matrix
  - ClustalW uses the neighbor-joining method
  - Guide tree roughly reflects evolutionary relations

Step 2: Guide Tree (cont'd)

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



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
- Following the guide tree, add in the next sequences, aligning to the existing alignment
- Insert gaps as necessary

```
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.

ClustalW: another example

S_1 ALSK
S_2 TNSD
S_3 NASK
S_4 NTSD

ClustalW example

S_1 ALSK
S_2 TNSD
S_3 NASK
S_4 NTSD

All pairwise alignments

	S_1	S_2	S_3	S_4
S_1	0	9	4	7
S_2		0	8	3
S_3			0	7
S_4				0

Distance Matrix

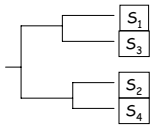
ClustalW example

S_1 ALSK
S_2 TNSD
S_3 NASK
S_4 NTSD

All pairwise alignments

	S_1	S_2	S_3	S_4
S_1	0	9	4	7
S_2		0	8	3
S_3			0	7
S_4				0

Neighbor Joining



Rooted Tree

Distance Matrix

ClustalW example

S_1 ALSK
S_2 TNSD
S_3 NASK
S_4 NTSD

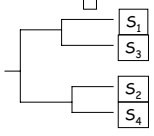
All pairwise alignments

	S_1	S_2	S_3	S_4
S_1	0	9	4	7
S_2		0	8	3
S_3			0	7
S_4				0

Multiple Alignment Steps

1. Align S_1 with S_3
2. Align S_2 with S_4
3. Align (S_1, S_3) with (S_2, S_4)

Neighbor Joining



Rooted Tree

Distance Matrix

ClustalW example

S_1 ALSK
S_2 TNSD
S_3 NASK
S_4 NTSD

All pairwise alignments

-ALSK
-TNSD
NA-SK
NT-SD

Multiple Alignment

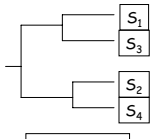
-ALSK
NA-SK
-TNSD
NT-SD

Multiple Alignment Steps

1. Align S_1 with S_3
2. Align S_2 with S_4
3. Align (S_1, S_3) with (S_2, S_4)

	S_1	S_2	S_3	S_4
S_1	0	9	4	7
S_2		0	8	3
S_3			0	7
S_4				0

Neighbor Joining



Rooted Tree

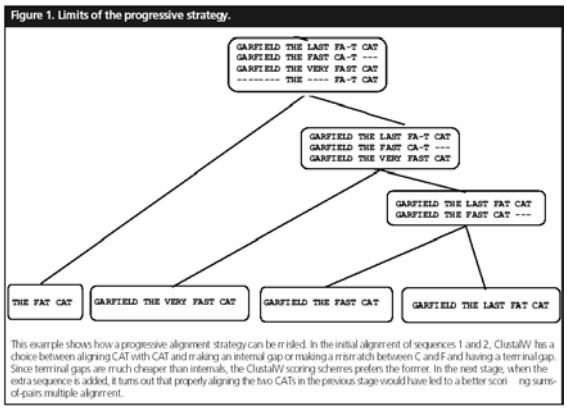
Distance Matrix

Other progressive approaches

- PILEUP
  - Similar to CLUSTALW
  - Uses UPGMA to produce tree

Problems with progressive alignments

- Depend on pairwise alignments
- If sequences are very distantly related, much higher likelihood of errors
- Care must be made in choosing scoring matrices and penalties



Iterative refinement in progressive alignment

Another problem of progressive alignment:

- Initial alignments are "frozen" even when new evidence comes

Example:

```
x:  GAAGTT
y:  GAC-TT   >   Frozen!

z:  GAACTG   >   Now clear that correct y = GA-CTT
w:  GTACTG
```

Evaluating multiple alignments

- Balibase benchmark (Thompson, 1999)
- De-facto standard for assessing the quality of a multiple alignment tool
- Manually refined multiple sequence alignments
- Quality measured by how good it matches the core blocks
- Another benchmark: SABmark benchmark
  - Based on protein structural families

Scoring multiple alignments

- Ideally, a scoring scheme should
  - Penalize variations in conserved positions higher
  - Relate sequences by a phylogenetic tree
    - Tree alignment
- Usually assume
  - Independence of columns
  - Quality computation
    - Entropy-based scoring
      - Compute the Shannon entropy of each column
    - Sum-of-pairs (SP) score

Multiple Alignments: Scoring

- Number of matches (multiple longest common subsequence score)
- Entropy score
- Sum of pairs (SP-Score)

Multiple LCS Score

- A column is a "match" if all the letters in the column are the same

AAA
AAA
AAT
ATC

- Only good for very similar sequences

Entropy

- Define frequencies for the occurrence of each letter in each column of multiple alignment
  - $p_A = 1$, $p_T=p_G=p_C=0$ (1st column)
  - $p_A = 0.75$, $p_T = 0.25$, $p_G=p_C=0$ (2nd column)
  - $p_A = 0.50$, $p_T = 0.25$, $p_C=0.25$ $p_G=0$ (3rd column)
- Compute entropy of each column

$$-\sum_{X=A,T,G,C} p_X \log p_X$$

AAA
AAA
AAT
ATC

Entropy: Example

$$entropy\begin{pmatrix}A\\A\\A\\A\end{pmatrix}=0$$ Best case

Worst case $$entropy\begin{pmatrix}A\\T\\G\\C\end{pmatrix}=-\sum\frac{1}{4}\log\frac{1}{4}=-4(\frac{1}{4}*-2)=2$$

Multiple Alignment: Entropy Score

Entropy for a multiple alignment is the sum of entropies of its columns:

$$\Sigma_{\text{ over all columns }} -\Sigma_{X=A,T,G,C} p_X \log p_X$$

Entropy of an Alignment: Example

column entropy:
$-(p_A\log p_A + p_C\log p_C + p_G\log p_G + p_T\log p_T)$

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

- Column 1 = -[1*log(1) + 0*log0 + 0*log0 +0*log0]
  = 0
- Column 2 = -[(1/4)*log(1/4) + (3/4)*log(3/4) + 0*log0 + 0*log0]
  = -[(1/4)*(-2) + (3/4)*(-.415)] = +0.811
- Column 3 = -[(1/4)*log(1/4)+(1/4)*log(1/4)+(1/4)*log(1/4) +(1/4)*log(1/4)]
  = 4* -[(1/4)*(-2)] = +2.0
- Alignment Entropy = 0 + 0.811 + 2.0 = +2.811

Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments

```
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
```

Sum of Pairs (SP) Scoring

- SP scoring is the standard method for scoring multiple sequence alignments.
- Columns are scored by a 'sum of pairs' function using a substitution matrix (PAM or BLOSUM)
- Assumes statistical independence for the columns, does not use a phylogenetic tree.

Sum of Pairs Score(SP-Score)

- Consider pairwise alignment of sequences
  a_i and a_j
  imposed by a multiple alignment of k sequences
- Denote the score of this suboptimal (not necessarily optimal) pairwise alignment as
  $s^*(a_i, a_j)$
- Sum up the pairwise scores for a multiple alignment:
  $s(a_1,\ldots,a_k) = \Sigma_{i,j}\, s^*(a_i, a_j)$

Computing SP-Score

Aligning 4 sequences: 6 pairwise alignments

Given a_1,a_2,a_3,a_4:

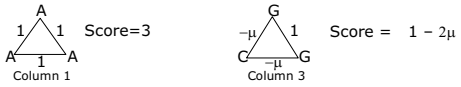
$$s(a_1\ldots a_4) = \Sigma s^*(a_i,a_j) = s^*(a_1,a_2) + s^*(a_1,a_3) + s^*(a_1,a_4) + s^*(a_2,a_3) + s^*(a_2,a_4) + s^*(a_3,a_4)$$

SP-Score: Example

a_1 ATG–C–AAT
. A–G–CATAT
a_k ATCCCATTT

$$S(a_1\ldots a_k) = \sum_{i,j} S^*(a_i,a_j) \longleftarrow \binom{n}{2} \text{ Pairs of Sequences}$$

May also calculate the scores column by column:



Example

- Compute Sum of Pairs Score of the following multiple alignment with match = 3, mismatch = -1, S(X,-) = -1, S(-,-) = 0

```
X: G T A C G
Y: T G C C G
Z: C G G C C
W: C G G A C
  -2 6-2 6 2
```

Sum of pairs = -2+6-2+6+2 = 10

Multiple alignment tools

- Clustal W (Thompson, 1994)
  - Most popular
- PRRP (Gotoh, 1993)
- HMMT (Eddy, 1995)
- DIALIGN (Morgenstern, 1998)
- T-Coffee (Notredame, 2000)
- MUSCLE (Edgar, 2004)
- Align-m (Walle, 2004)
- PROBCONS (Do, 2004)

Table 1. Some recent and less recent available methods for MSAs.

Name	Algorithm	URL
MSA	Exact	http://www.ibc.wustl.edu/ibc/msa.html
DCA	Exact (requires MSA)	http://bibiserv.techfak.uni-bielefeld.de/dca
OMA	Iterative DCA	http://bibiserv.techfak.uni-bielefeld.de/oma
ClustalW, ClustalX	Progressive	ftp://ftp-igbmc.u-strasbg.fr/pub/clustalW or clustalX
MultAlin	Progressive	http://www.toulouse.inra.fr/multalin.html
DiAlign	Consistency-based	http://www.gsf.de/biodv/dialign.html
CorrAlign	Consistency-based	http://www.daimi.au.dk/~ocaprani
T-Coffee	Consistency-based/progressive	http://igs-server.cnrs-mrs.fr/~cnotred
Praline	Iterative/progressive	jhering@nimr.mrc.ac.uk
IterAlign	Iterative	http://giotto.Stanford.edu/~luciano/iteralign.html
Prrp	Iterative/Stochastic	ftp://ftp.genome.ad.jp/pub/genome/saitama-cc/
SAM	Iterative/Stochastic/HMM	rph@cse.ucsc.edu
HMMER	Iterative/Stochastic/HMM	http://hmmer.wustl.edu/
SAGA	Iterative/Stochastic/GA	http://igs-server.cnrs-mrs.fr/~cnotred
GA	Iterative/Stochastic/GA	czhang@watnow.uwaterloo.ca

from: C. Notredame, "Recent progresses in multiple alignment: a survey", *Pharmacogenomics* (2002) 3(1)

Useful links

http://cnx.org/content/m11036/latest/

http://www.biokemi.uu.se/Utbildning/Exercises/ClustalX/index.shtm

http://bioinformatics.weizmann.ac.il/~pietro/Making_and_using_protein_MA/

http://homepage.usask.ca/~ctl271/857/paper1_overview.shtml

http://journal-ci.csse.monash.edu.au/ci/vol04/mulali/mulali.html