Algorithms and Tools in Bioinformatics

Algorithms: Multiple Sequence Alignment
(adapted from Prof. Stephan Winkler)

Julia Vetter
julia.vetter@fh-hagenberg.at

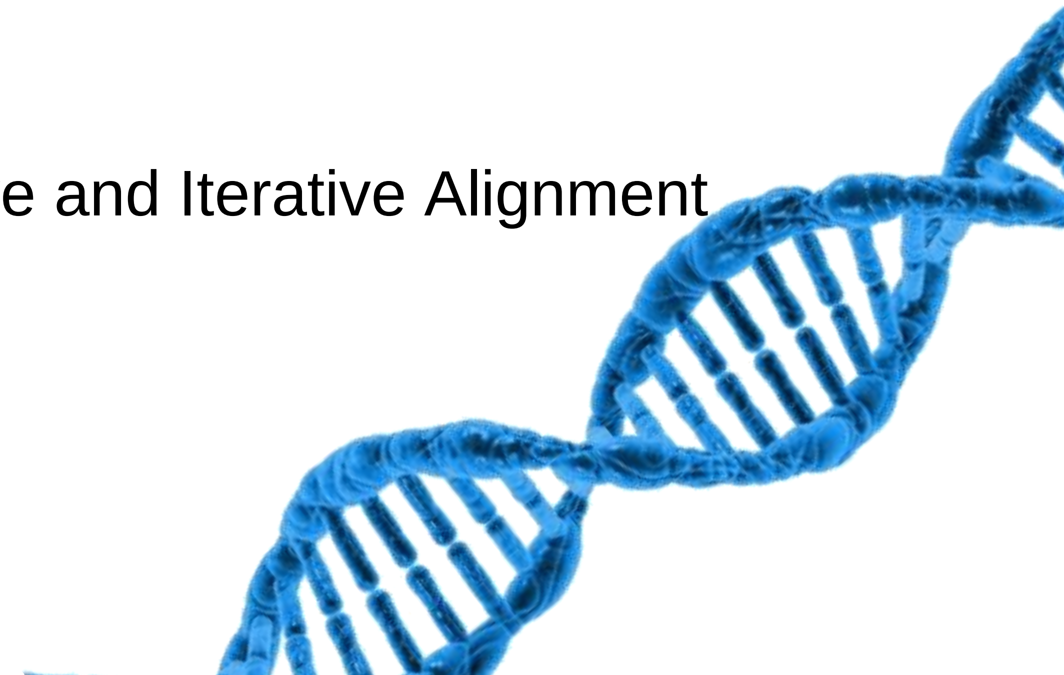
SS2024

(5) Multiple Sequence Alignment

Scoring

Strategies: Exact Algorithms, Progressive and Iterative Alignment

Profiles



Global Multiple Sequence Alignment (MSA)

- Multiple alignments must usually be inferred from primary sequences alone. Biologists produce high quality multiple sequence alignments by hand using expert knowledge of protein sequence evolution. Automatic multiple alignment methods must have a way to assign a score so that better multiple alignments get better scores.

- In a multiple sequence alignment, homologous residues among a set of sequences are aligned together in columns. **Homologous** is meant in both the structural and evolutionary sense. In practice an evolutionary correct alignment can be even more difficult to infer than a structural alignment. Except for trivial cases of highly identical sequences, it is not possible to unambiguously identify structurally and evolutionarily homologous positions and create a single correct multiple alignments. Family of proteins share perhaps only 30 % average pair wise sequence identity.

Global Multiple Sequence Alignment (MSA)

- Example:
  MSA of 4 sequences `MQPILLLV`, `MLRLL`, `MKILLL`, and `MPPVLILV`:

    - `MQPILLLV`
    - `MLR--LL-`
    - `MK-ILLL-`
    - `MPPVLILV`

Duality

- **Pairwise sequence comparison:**
  Find sequences with common sub-patterns that were not known to be biologically related

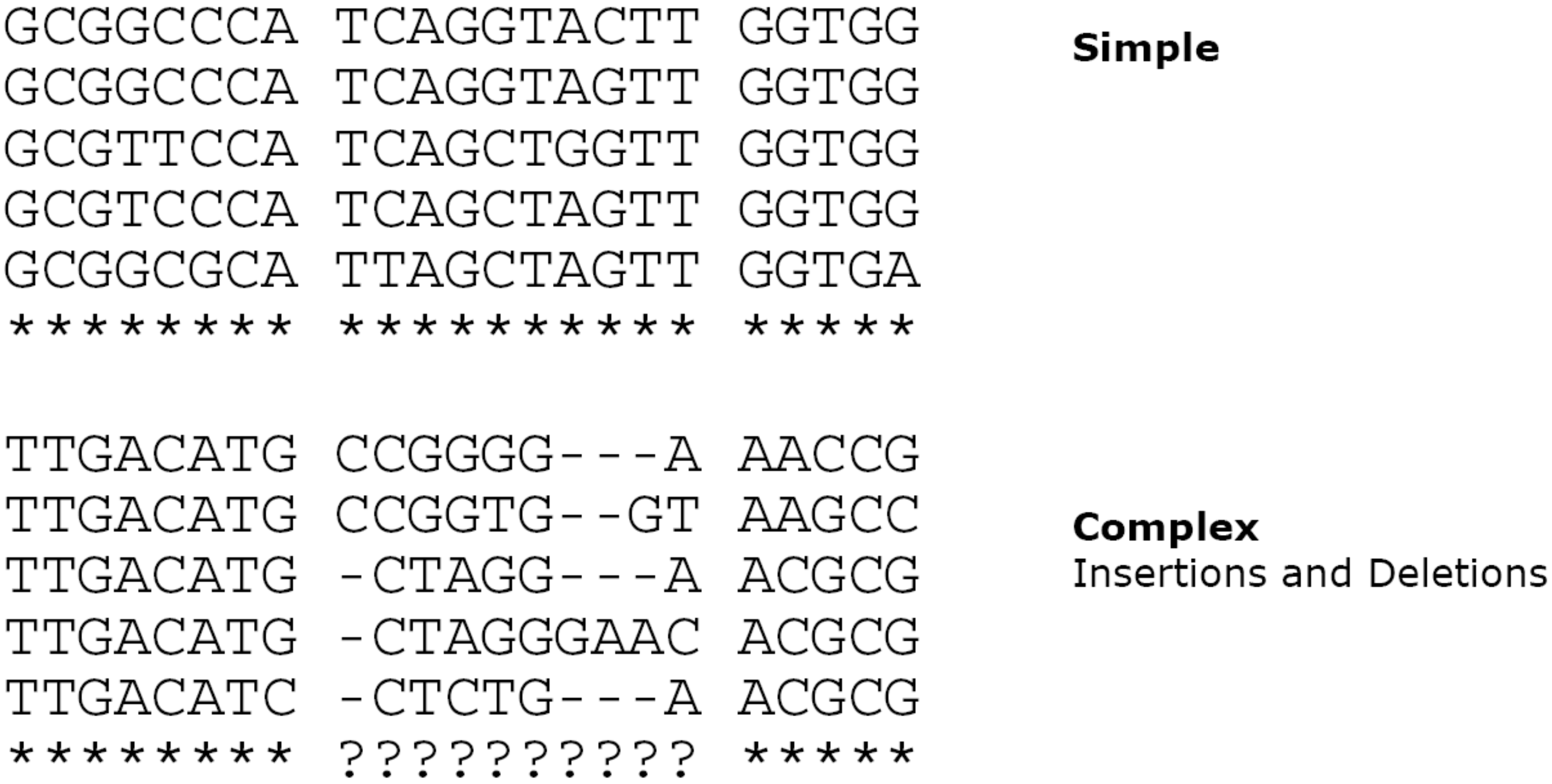
- **Multiple sequence alignment:**
  From known biological relationships deduce similarities in sequences

Motivation

Multiple sequence alignments are used for many reasons, including:

- to detect regions of variability or conservation in a family of proteins
- to provide stronger evidence than pairwise similarity for structural and functional inferences
- to extracting and representing biologically important commonalities from a set of biological sequences such as proteins/genes of the same family.
- because important commonalities may be faint or widely dispersed, they might not be apparent in pairwise comparisons.
- Commonalities are useful for revealing common history, function or 3 dimensional structure.
- Evolutionary related sequences may differ significantly and yet preserve the same 2D or 3D structure or the same active sites.
- Pairwise alignment of close sequences does not reveal important features: non-functional sites are homologue as well.
- Pairwise alignment of remote but related sequences does not reveal important features: weak homologies at functional sites are too close to noise.

Simple and Complex Alignments



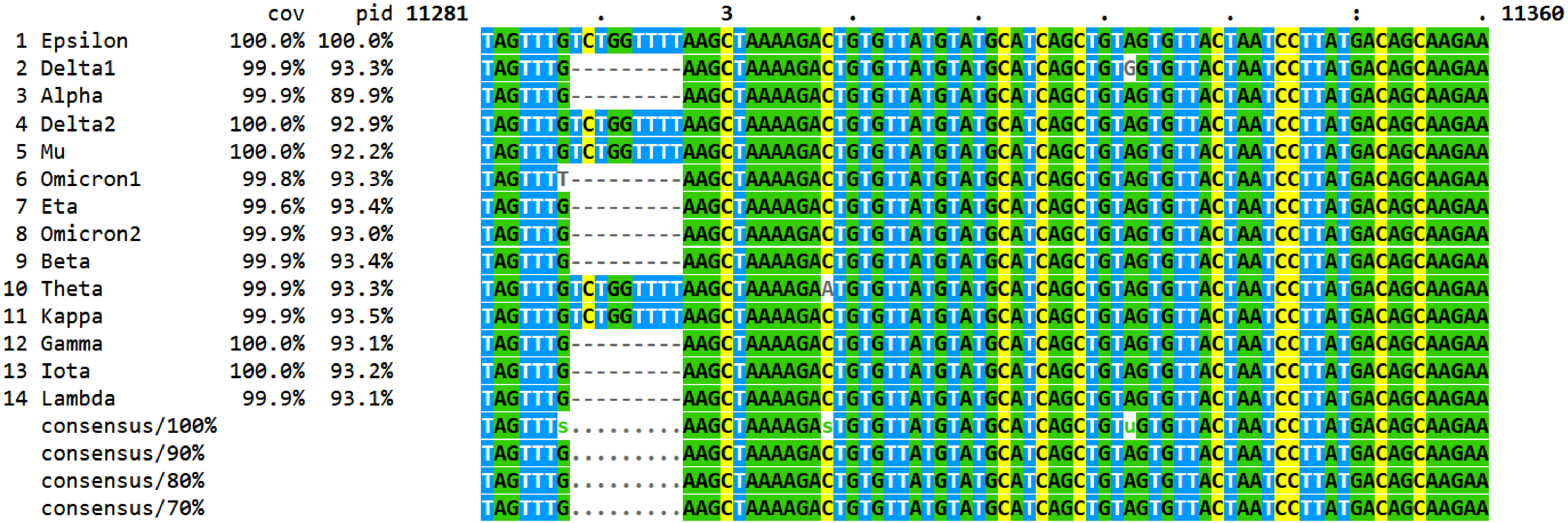
```
GCGGCCCA TCAGGTACTT GGTGG
GCGGCCCA TCAGGTAGTT GGTGG
GCGTTCCA TCAGCTGGTT GGTGG
GCGTCCCA TCAGCTAGTT GGTGG
GCGGCGCA TTAGCTAGTT GGTGA
******** ********** *****
```

Simple

```
TTGACATG CCGGGG---A AACCG
TTGACATG CCGGTG--GT AAGCC
TTGACATG -CTAGG---A ACGCG
TTGACATG -CTAGGGAAC ACGCG
TTGACATC -CTCTG---A ACGCG
******** ?????????? *****
```

Complex
Insertions and Deletions

Beispiel: MSA von SARS-CoV2 Sequenzen



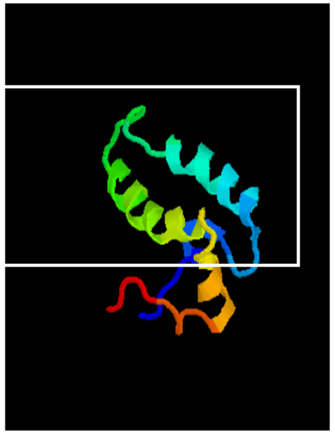
```
                     cov    pid  11281          .         3         .         .         .         .         :         . 11360
 1 Epsilon        100.0% 100.0%       TAGTTTGTCTGGTTTTAAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 2 Delta1          99.9%  93.3%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTGGTGTTACTAATCCTTATGACAGCAAGAA
 3 Alpha           99.9%  89.9%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 4 Delta2         100.0%  92.9%       TAGTTTGTCTGGTTTTAAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 5 Mu             100.0%  92.2%       TAGTTTGTCTGGTTTTAAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 6 Omicron1        99.8%  93.3%       TAGTTTT---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 7 Eta             99.6%  93.4%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 8 Omicron2        99.9%  93.0%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
 9 Beta            99.9%  93.4%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
10 Theta           99.9%  93.3%       TAGTTTGTCTGGTTTTAAGCTAAAAGAATGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
11 Kappa           99.9%  93.5%       TAGTTTGTCTGGTTTTAAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
12 Gamma          100.0%  93.1%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
13 Iota           100.0%  93.2%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
14 Lambda          99.9%  93.1%       TAGTTTG---------AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
   consensus/100%                     TAGTTTs.........AAGCTAAAAGAsTGTGTTATGTATGCATCAGCTGTuGTGTTACTAATCCTTATGACAGCAAGAA
   consensus/90%                      TAGTTTG.........AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
   consensus/80%                      TAGTTTG.........AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
   consensus/70%                      TAGTTTG.........AAGCTAAAAGACTGTGTTATGTATGCATCAGCTGTAGTGTTACTAATCCTTATGACAGCAAGAA
```

https://www.ebi.ac.uk/Tools/msa/clustalo/

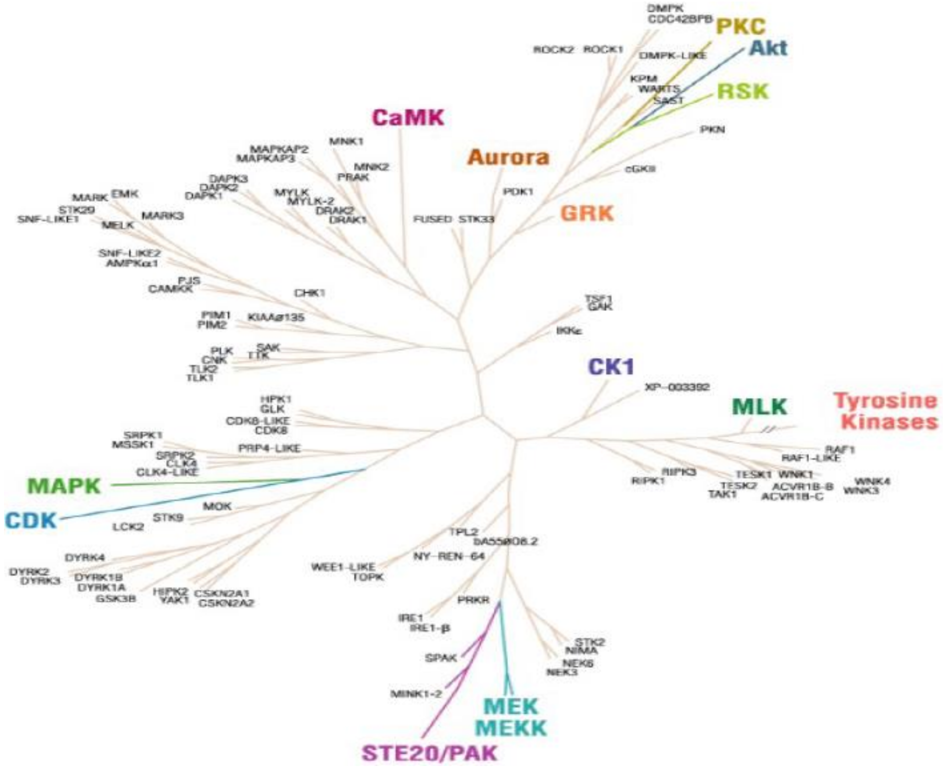
Scoring of multiple alignments problem

- Scoring system should take into account at least two important features:
  - The fact that some positions are more conserved than others – position- specific scoring
  - The fact that the sequences are not independent, but instead are related by a phylogenetic tree.

Escherichia coli DjlA protein



Homo sapiens DjlA protein



Similarity Measures

- For pairwise alignments, we aligned sequences to maximize the similarity score. Almost all alignment methods assume that the individual columns of an alignment are statistically independent.

- The scoring function can be written as:

$$S(m) = G + \Sigma_i S(m_i)$$

where m_i is column i of the multiple alignment m, $S(m_i)$ is the score for column i and G is a function for scoring the gaps that occur in the alignment.

Minimum Entropy Measure

- Definitions:
  - Let m_{ij} be a symbol in column *i* for sequence *j*.
  - Let c_{ia} be the observed counts for residue *a* in column *i*.
  - Let mi be the column $m_{i1}, \ldots, m_{iN}$ of aligned symbols in column *i* and let c_i be the count vector $c_i = c_{i1}, \ldots, c_{iK}$ of observed symbols in column *i* for an alphabet of *K* different residues.

- If we assume that residues within the column are independent, as well as being independent between columns, then the probability of a column mi is

$$P(mi) = \Pi_a (p_{ia})C_{ia}$$

  where p_{ia} is the probability of residue a in column i and can be estimated from counts cia

$$p_{ia} = c_{ia} / \Sigma_x c_{ix}$$

- Column score :

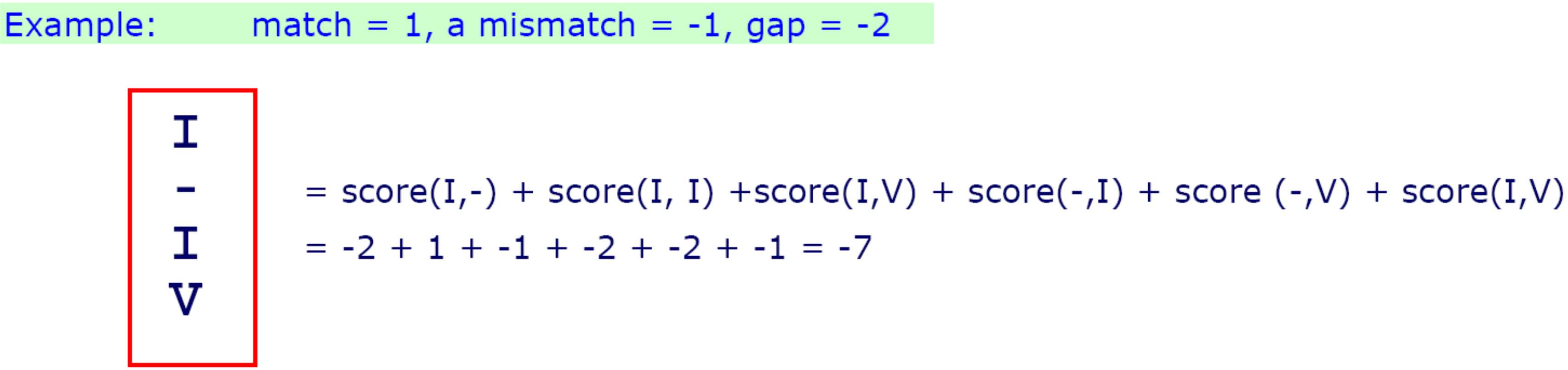
$$S(m_i) = - \Sigma_a c_{ia} \log p_{ia}$$

Sum-of-pairs (SP Scores Measure)

- With multiple sequences, it is not obvious which way is the best to score an alignment
- **Sum-of-pairs (SP)** is a commonly studied similarity measure for MSAs
- Each column is scored by summing the scores of all pairs of symbols in that column. The SP score for a column is defined as

where scores *s(a,b)* come fr $S(m_i) = \sum_{k<n} s(m_i^k m_i^n)$ such as PAM or BLOSUM matrix. For simple linear gaps costs, gaps are handled by defining *s(a,-)* = *s(-,a)*=*g* to be the gap cost and *s(-,-)* to be zero.

Sum-of-pairs (SP Scores Measure)



Example: match = 1, a mismatch = -1, gap = -2

```
I
-
I
V
```

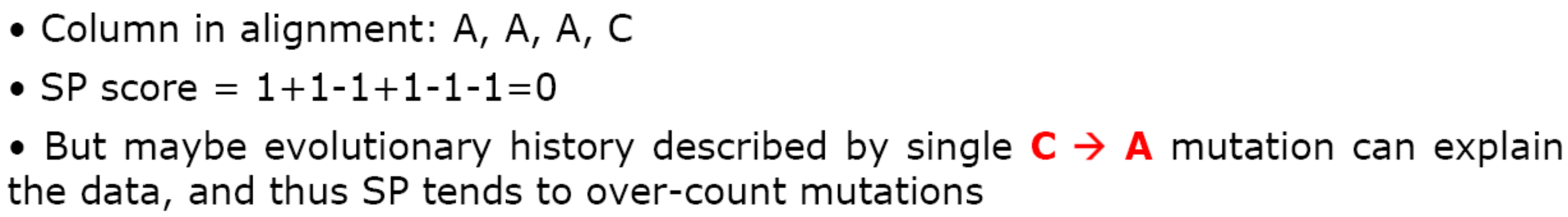
= score(I,-) + score(I, I) +score(I,V) + score(-,I) + score (-,V) + score(I,V)

= -2 + 1 + -1 + -2 + -2 + -1 = -7

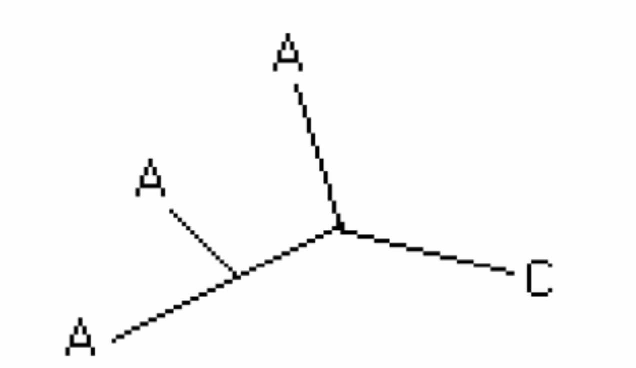
Sum-of-pairs (SP Scores Measure)

- The substitution matrices used by standard algorithms were PAM or BLOSUM matrices with an affined gap penalty

- ***Is SP a good measure?***

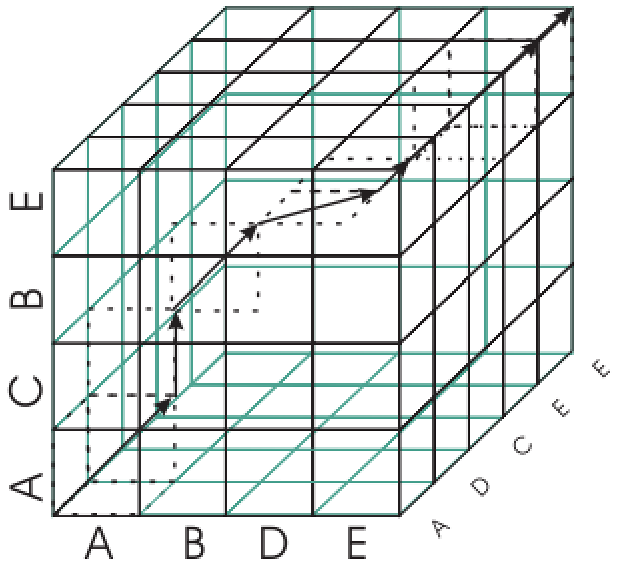


- Column in alignment: A, A, A, C
- SP score = 1+1-1+1-1-1=0
- But maybe evolutionary history described by single **C → A** mutation can explain the data, and thus SP tends to over-count mutations



Different strategies to perform Multiple Sequence Alignments

Exact Alignment Method



Progressive Alignment Method

Step-1:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--

Step-2:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--
Seq3: --AGGACTTG-C

Step-3:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--
Seq3: --AGGACTTG-C
Seq4: AACCAACATT--

Iterative Refinement Method

Step-1:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--

Step-2:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--
Seq3: --AGGACTTG-C

Step-3:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--
Seq3: --AGGACTTG-C
Seq4: AACCAACATT--

Step-4:

Seq1: AGCCAACCTGGC
Seq2: AATTAATGTA--
Seq3: --AGGACTT-GC
Seq4: AACCAACATT--

Optimal pairwise alignments (Review)

- Used dynamic programming

- If two length n sequences: (n+1) x (n+1) array

- Fill out each box in the array by considering what happens in the last column: 3 choices: align last letters from both sequences, align last letter from 1st sequence with gap, align last letter from 2nd sequence with gap

- O(n^2) algorithm

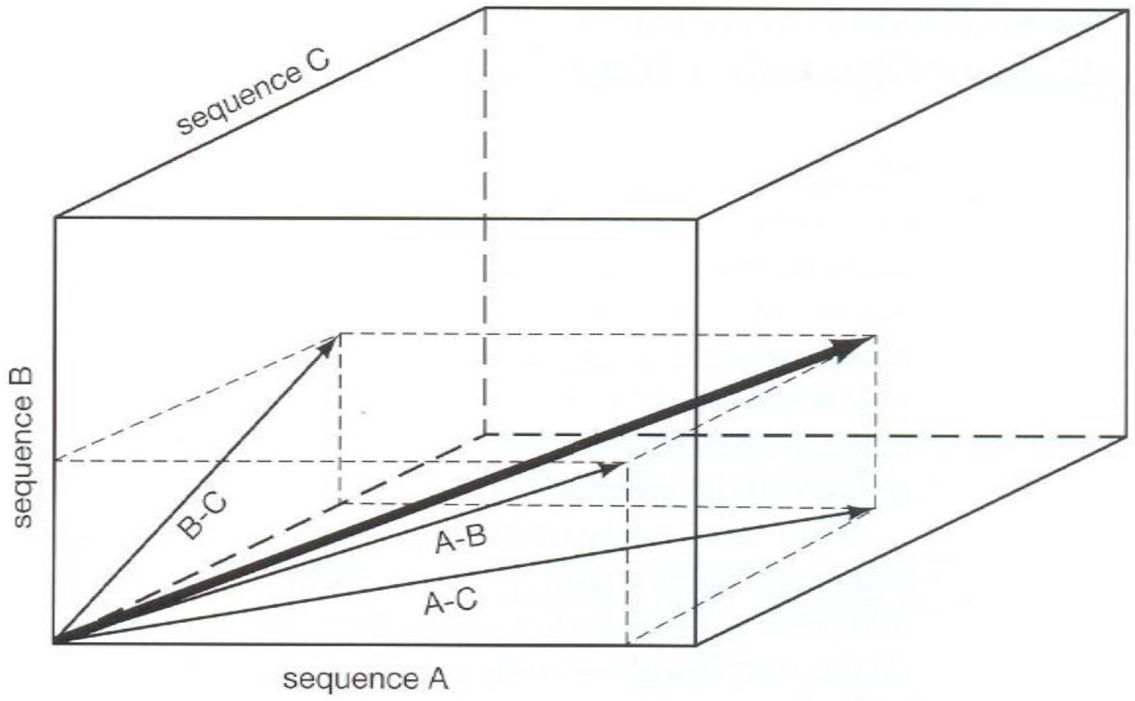
Finding optimal MSAs:
Multidimensional dynamic programming

Typical algorithms are:

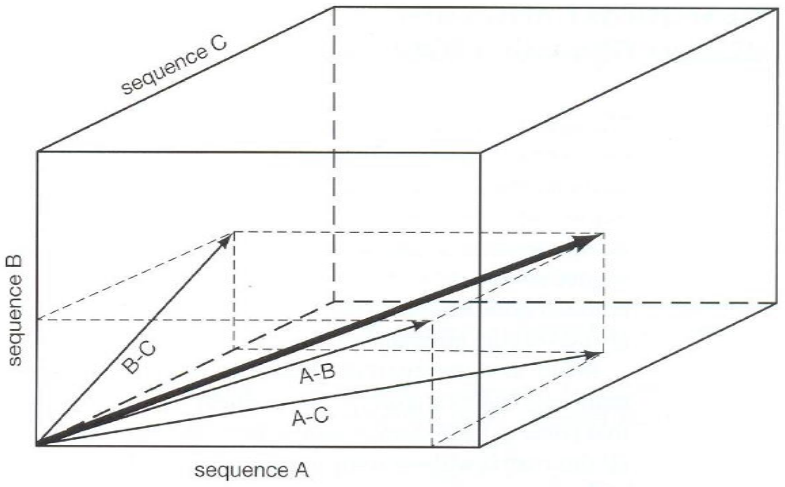
- Exact algorithms that compute the optimal alignment. They are NP complete unless there is additional information provided.
- Bound and error approximations of the exact ones.
- Heuristic algorithms that compute a high quality sub-optimal alignment in reasonable time:
  - Progressive alignments
  - Iterative alignments
  - Phylogenetic tree construction algorithms that yield a multiple alignment as a byproduct.
  - Algorithms that identifies significant motifs and produce a partial alignment as a byproduct.

Exact algorithms

- We can use dynamic programming to find optimal solutions. It is possible to generalize pairwise dynamic programming to the alignment of N sequences.



Exact algorithms



- Bsp: Betrache 2 Proteinsequenzen von 100 Aminosäuren Länge. Wenn es 100^2 Sekunden dauert, diese beiden Sequenzen erschöpfend zu alignieren, dann wird es 100^3 Sekunden dauern um 3 Sequenzen zu alignieren, 100^4 Sekunden für 4 Sequenzen und **1.90258 x 10^{34} Jahre** für 20 Sequenzen.

Exact algorithms

Define $\alpha_{i1,i2,..,iN}$ as the maximum score of an alignment up to the subsequences ending with $x_{i1}^{1},x_{i2}^{2},..,x_{iN}^{N}$

$$\alpha_{i1,i2,..,iN} = \max \begin{cases} \alpha_{i1-1,i2-1,..,iN-1} & + \quad s(x_{i1}^{1},x_{i2}^{2},..,x_{iN}^{N}) \\ \alpha_{i1,i2-1,..,iN-1} & + \quad s(-,x_{i2}^{2},.....,x_{iN}^{N}) \\ \alpha_{i1-1,i2,..,iN-1} & + \quad s(x_{i1}^{1},-,.....,x_{iN}^{N}) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ \alpha_{i1-1,i2-1,..,iN} & + \quad s(x_{i1}^{1},x_{i2}^{2},.....,-) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ \alpha_{i1,i2-1,..,iN} & + \quad s(-,x_{i2}^{2},........,-) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \end{cases}$$

If have N sequences of length n, array is of size $(n+1)^N$ and in considering last column, have 2^N-1 choices, e.g., align last letters from all sequences; align last letter from one sequence and gaps in all others, etc.

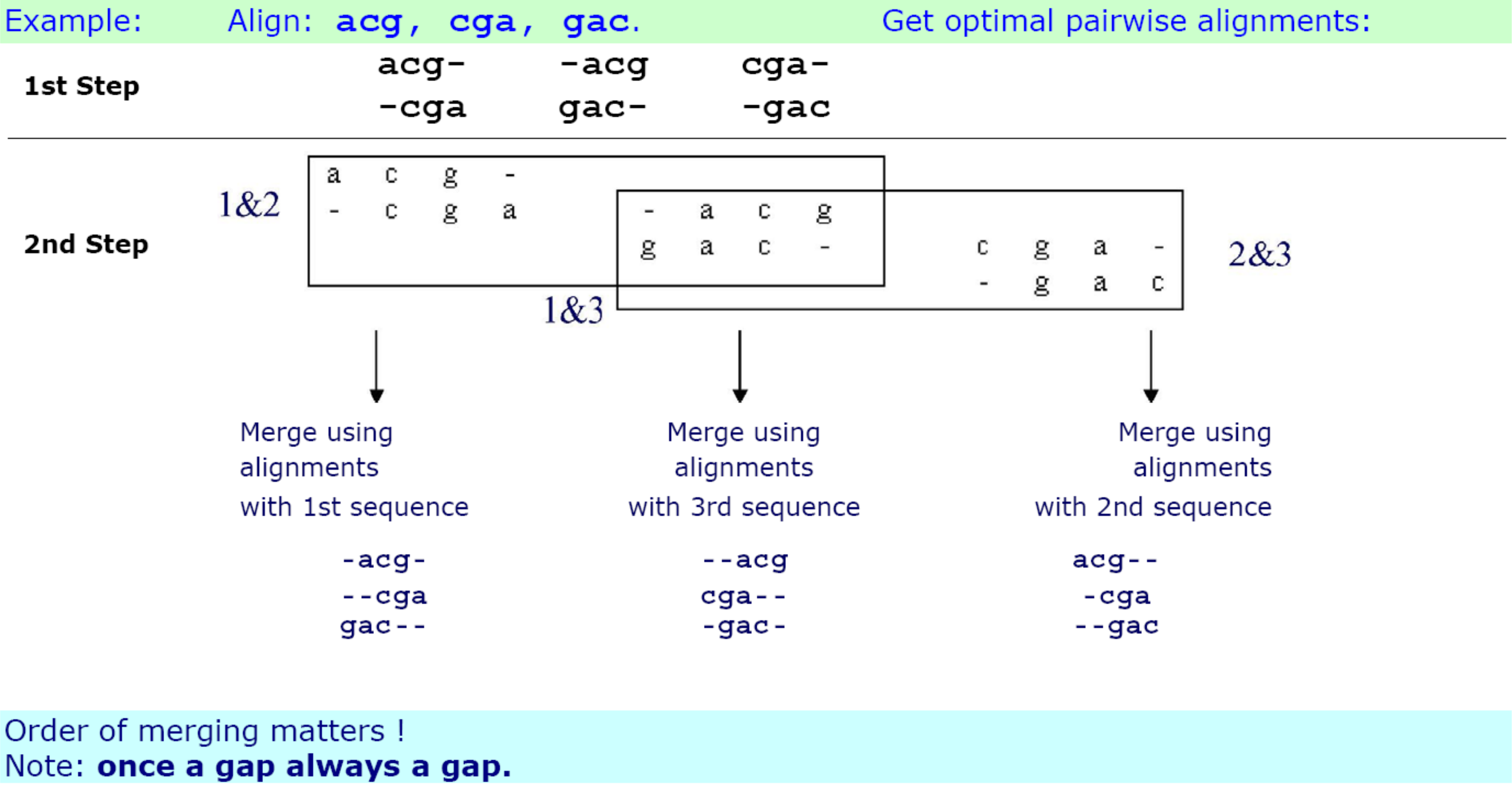
The memory complexity such algorithm is $O(n^N)$ and the time complexity is $O(2^N n^N)$.

Running time is exponential in the number of sequences !

Impractical → MSA packages use heuristics

Progressive alignment (Heuristic)

- Basic idea: compute pairwise alignments and merge alignments consistently



Example: Align: **acg, cga, gac.** Get optimal pairwise alignments:

1st Step

```
acg-    -acg    cga-
-cga    gac-    -gac
```

2nd Step

```
1&2        1&3        2&3
a c g -    - a c g    c g a -
- c g a    g a c -    - g a c
```

Merge using alignments with 1st sequence

```
-acg-
--cga
gac--
```

Merge using alignments with 3rd sequence

```
--acg
cga--
-gac-
```

Merge using alignments with 2nd sequence

```
acg--
-cga
--gac
```

Order of merging matters !
Note: **once a gap always a gap.**

Progressive Alignment Algorithm: Feng-Doolittle

- **Step 1:** Calculate a diagonal matrix of N(N-1)/2 distances between all pairs of N sequences by standard pairwise alignment, converting alignment scores to approximate pairwise distances e.g. determine degrees of similarity between each pair.

$$\textit{Distance} = \mathrm{D} = (\mathrm{score} - \mathrm{score}_{\mathrm{random}})/(\mathrm{score}_{\mathrm{max}} - \mathrm{score}_{\mathrm{random}})$$

or

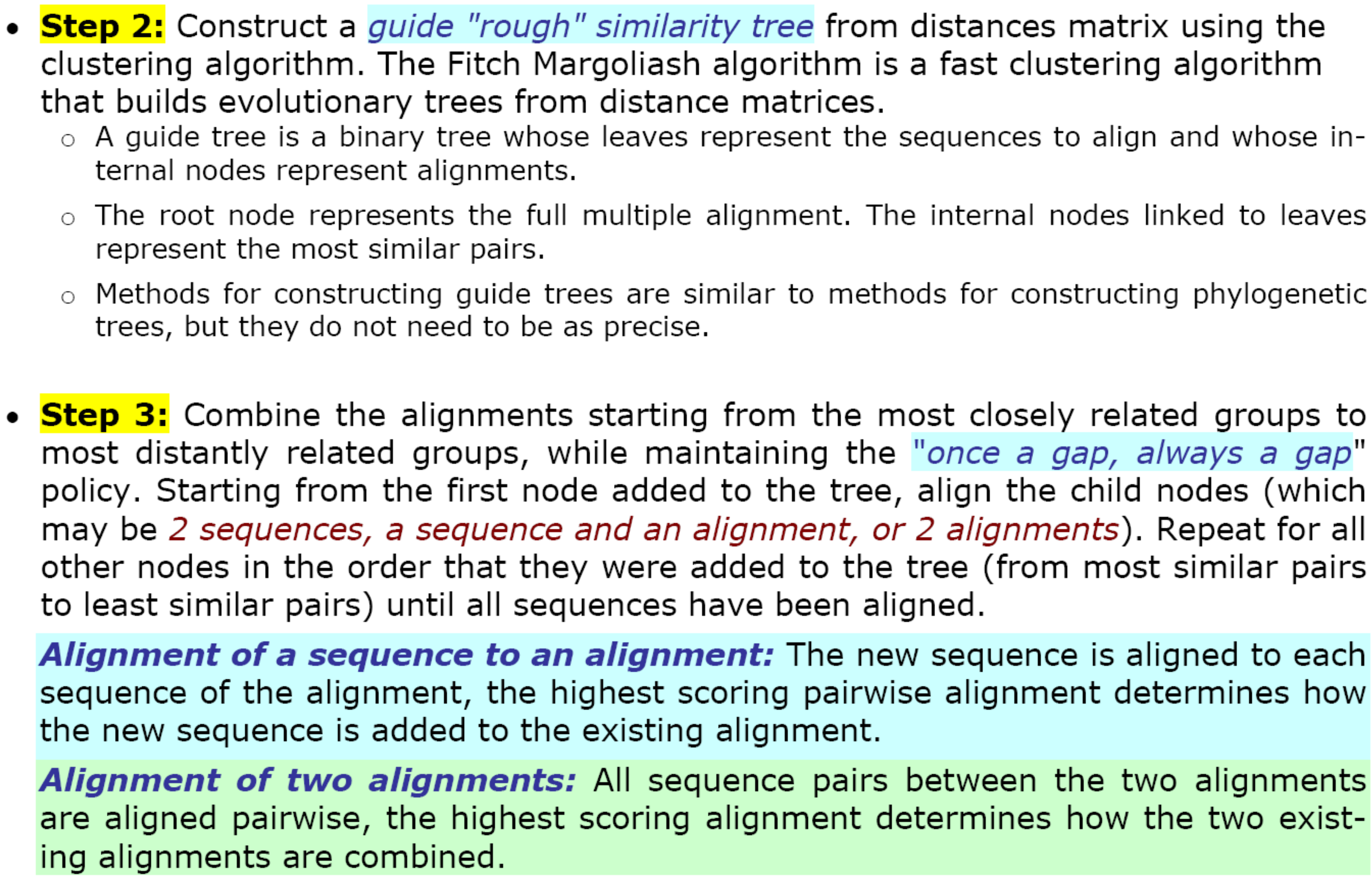
$$\textit{Distance} = \mathrm{D} = \text{number of mismatches / number of columns with no gaps}$$

More exactly, the formula for converting an alignment similarity score to a distance D is

$$\mathrm{D} = -\log((\mathrm{score} - \mathrm{score}_{\mathrm{random}})/(\mathrm{score}_{\mathrm{max}} - \mathrm{score}_{\mathrm{random}}))$$

where $\mathrm{score}_{\mathrm{max}}$ is the average of the scores obtained by aligning each sequence to itself, and $\mathrm{score}_{\mathrm{random}}$ is the expected score for aligning to random sequences of the same length and amino acid composition. A formula or random shuffling of the 2 sequences can estimate it.

Progressive Alignment Algorithm: Feng-Doolittle



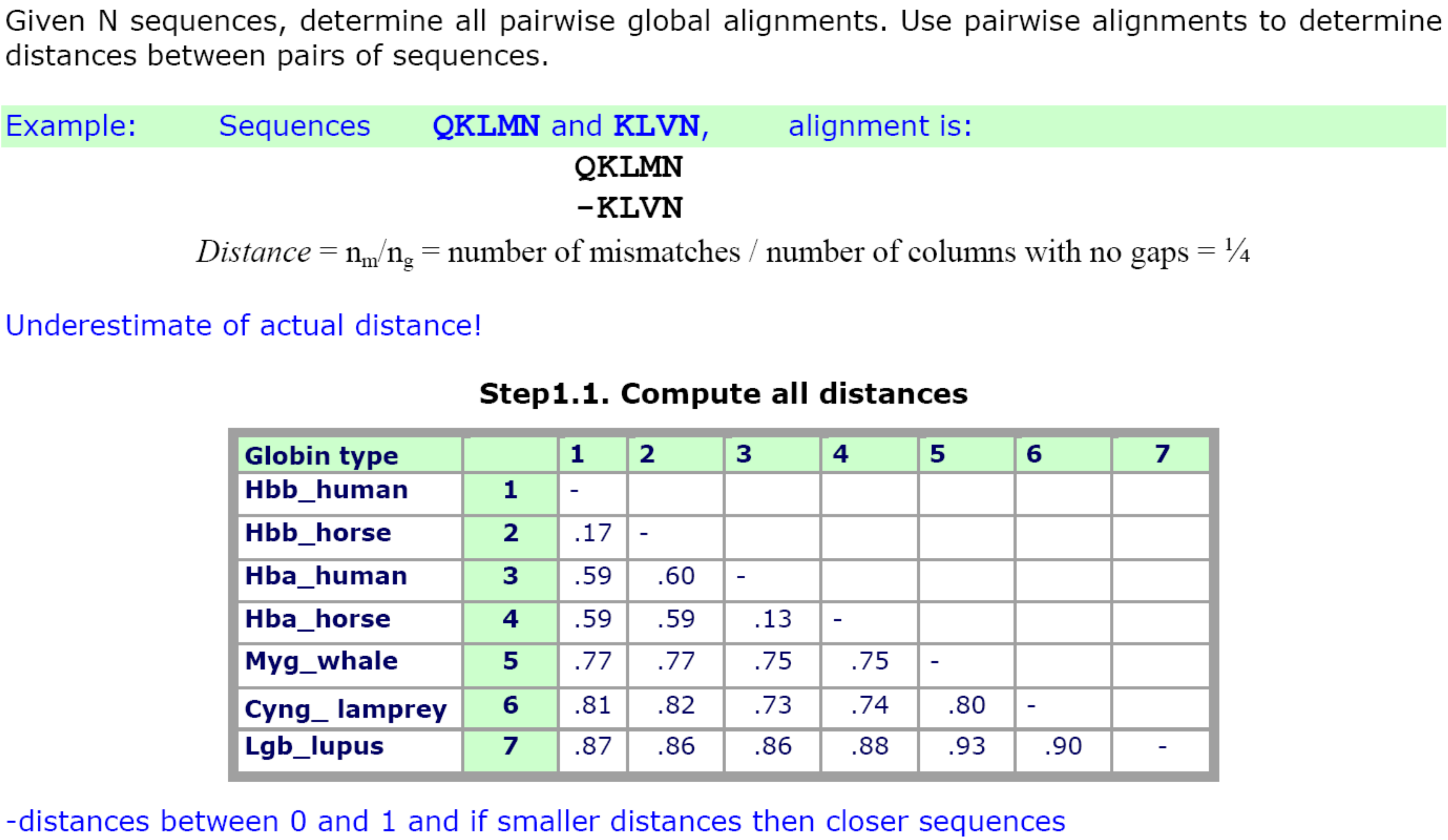
- **Step 2:** Construct a *guide "rough" similarity tree* from distances matrix using the clustering algorithm. The Fitch Margoliash algorithm is a fast clustering algorithm that builds evolutionary trees from distance matrices.
    - A guide tree is a binary tree whose leaves represent the sequences to align and whose internal nodes represent alignments.
    - The root node represents the full multiple alignment. The internal nodes linked to leaves represent the most similar pairs.
    - Methods for constructing guide trees are similar to methods for constructing phylogenetic trees, but they do not need to be as precise.

- **Step 3:** Combine the alignments starting from the most closely related groups to most distantly related groups, while maintaining the "*once a gap, always a gap*" policy. Starting from the first node added to the tree, align the child nodes (which may be *2 sequences, a sequence and an alignment, or 2 alignments*). Repeat for all other nodes in the order that they were added to the tree (from most similar pairs to least similar pairs) until all sequences have been aligned.

Alignment of a sequence to an alignment: The new sequence is aligned to each sequence of the alignment, the highest scoring pairwise alignment determines how the new sequence is added to the existing alignment.

Alignment of two alignments: All sequence pairs between the two alignments are aligned pairwise, the highest scoring alignment determines how the two existing alignments are combined.

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 1



Given N sequences, determine all pairwise global alignments. Use pairwise alignments to determine distances between pairs of sequences.

Example: Sequences **QKLMN** and **KLVN**, alignment is:

```
QKLMN
-KLVN
```

$Distance = n_m/n_g$ = number of mismatches / number of columns with no gaps = ¼

Underestimate of actual distance!

Step1.1. Compute all distances

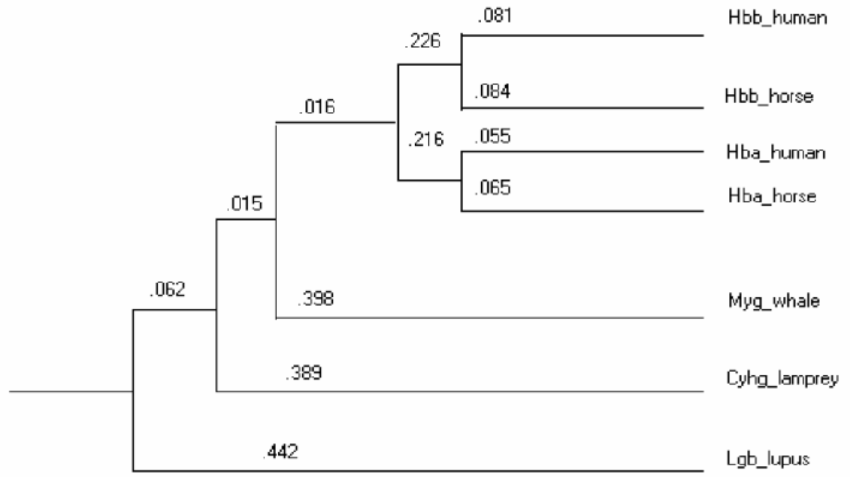
Globin type		1	2	3	4	5	6	7
Hbb_human	1	-						
Hbb_horse	2	.17	-					
Hba_human	3	.59	.60	-				
Hba_horse	4	.59	.59	.13	-			
Myg_whale	5	.77	.77	.75	.75	-		
Cyng_ lamprey	6	.81	.82	.73	.74	.80	-	
Lgb_lupus	7	.87	.86	.86	.88	.93	.90	-

-distances between 0 and 1 and if smaller distances then closer sequences

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 2

1. Distance matrix is fed into clustering algorithm that will build a tree relating these sequences (neighbor -joining)
2. Ideally, path length in tree between sequences is equal to distance in matrix (cannot always maintain this)

Example: Neighbor Joining Tree



Note: Figure not drawn to scale

distance between **Hbb_human** and **Hbb_horse** tree is 0.081 + 0.084 = 0.165 which is close to 0.17 from matrix

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 3

1. Start from the most closely related groups to most distantly related groups (start from tips to root in tree), while maintaining the "once a gap, always a gap" policy.

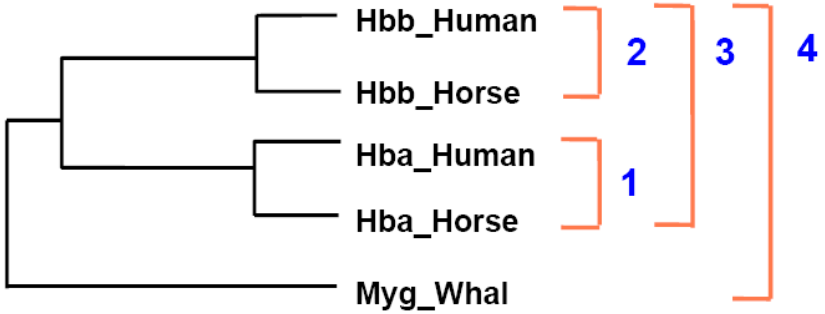
   – **Sequence – sequence** alignments are done with the usual pairwise dynamic programming algorithm.

   – A **sequence** is added to an existing **group** by aligning it pairwise to each sequence in the group in turn. The highest scoring pairwise alignment determines how the sequence will be aligned to the group.

   – For aligning a **group** to a **group**, all sequence pairs between the two groups are tried. The best pairwise sequence alignment determines the alignment of the two groups.

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 3

Example: First align **hba_human** & **hba_horse**; then **hbb_human** & **hbb_horse**; then **hba's** with **hbb's**; then add to that alignment **whale, lamprey** and **lupus** in turn



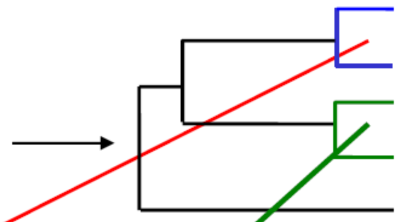
alpha-helices

```
Hbb_Human   1   1   PEEKSAVTALWGKVN--VDEVGG   ] 2 ] 3 ] 4
Hbb_Horse   2   2   GEEKAAVLALWDKVN--EEEVGG
Hba_Human   3   3   PADKTNVKAAWGKVGAHAGEYGA   ] 1
Hba_Horse   4   4   AADKTNVKAAWSKVGGHAGEYGA
Myg_Whale   5   5   EHEWQLVLHVWAKVEADVAGHGQ
```

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 3

Progressive Alignment   Step 3.1.

```
gctcgatacgatacgatgactagcta
gctcgatacaagacgatgacagcta
gctcgatacacgatgactagcta
gctcgatacacgatgacgagcga
ctcgaacgatacgatgactagct
```

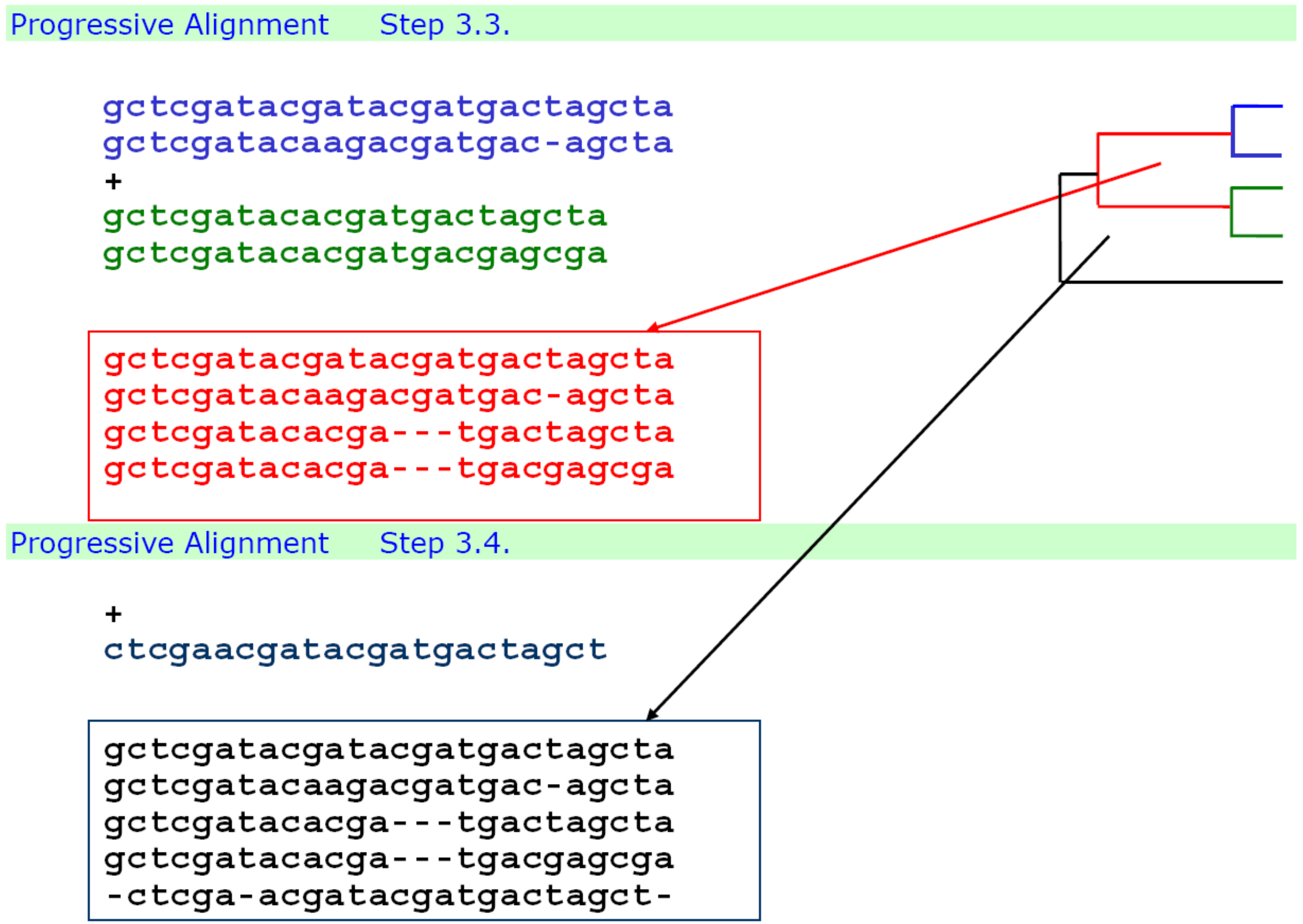


```
gctcgatacgatacgatgactagcta
gctcgatacaagacgatgac-agcta
```

Progressive Alignment   Step 3.2.

```
gctcgatacacgatgactagcta
gctcgatacacgatgacgagcga
```

Progressive Alignment Algorithm: Feng-Doolittle; An Example, Step 3



Progressive Alignment Step 3.3.

```
gctcgatacgatacgatgactagcta
gctcgatacaagacgatgac-agcta
+
gctcgatacacgatgactagcta
gctcgatacacgatgacgagcga
```

```
gctcgatacgatacgatgactagcta
gctcgatacaagacgatgac-agcta
gctcgatacacga---tgactagcta
gctcgatacacga---tgacgagcga
```

Progressive Alignment Step 3.4.

```
+
ctcgaacgatacgatgactagct
```

```
gctcgatacgatacgatgactagcta
gctcgatacaagacgatgac-agcta
gctcgatacacga---tgactagcta
gctcgatacacga---tgacgagcga
-ctcga-acgatacgatgactagct-
```

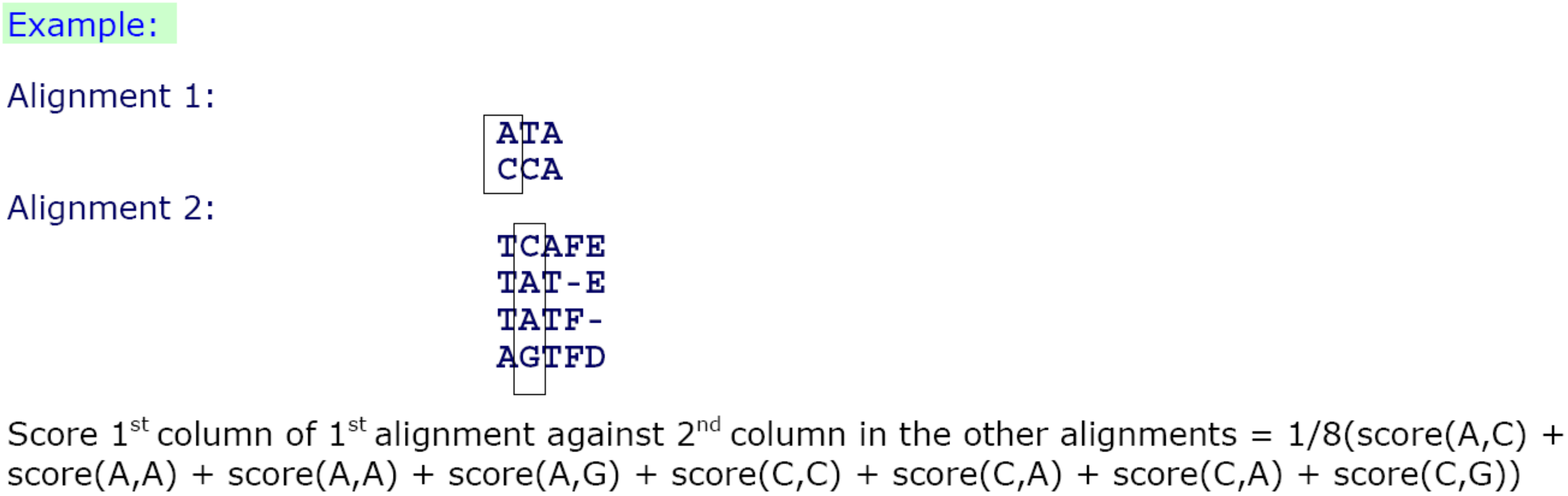
Aligning pairs of alignments

- Can be solved optimally using dynamic programming.

- Similarity between columns in two alignments is now the average similarity between the sequences.

- The problem is that all alignments are determined by pairwise sequence alignments. Once an aligned group has been built up it is advantageous to use position specific information from the group’s multiple alignment to align a new sequence to it. It also makes sense to apply profiles in progressive multiple sequence alignment.

Aligning pairs of alignments



Example:

Alignment 1:

```
ATA
CCA
```

Alignment 2:

```
TCAFE
TAT-E
TATF-
AGTFD
```

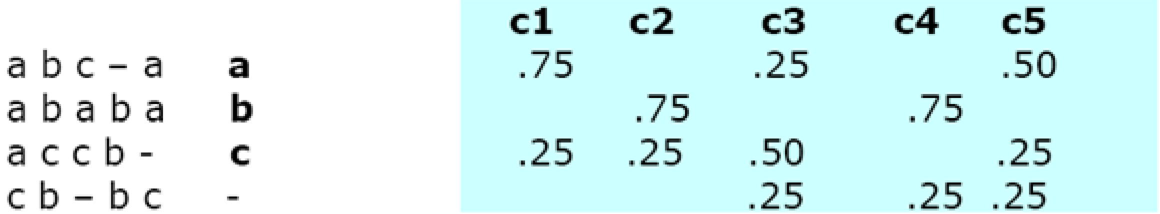
Score 1st column of 1st alignment against 2nd column in the other alignments = 1/8(score(A,C) + score(A,A) + score(A,A) + score(A,G) + score(C,C) + score(C,A) + score(C,A) + score(C,G))

Profiles: Protein Family Representation

- The representation of a protein family by means of a regular expression is called a motif, a signature, a motif signature, or a pattern, and sometimes, depending on the constraints on the motif, is also called anchor, block, region, identity segment, consensus pattern, etc.
- Examples of motifs to characterize protein families are stores in the PROSITE database (that also characterizes certain families by profiles instead of motifs!).

Profiles: Protein Family Representation

- Two classical models are used for characterizing a family of proteins: statistical models and regular expressions.
  - Statistical models are mainly profiles and hidden Markov models (HMMs).
  - A profile gives the relative frequency of each amino acid in each column of a multiple alignment. A profile is sometimes called a weight matrix.

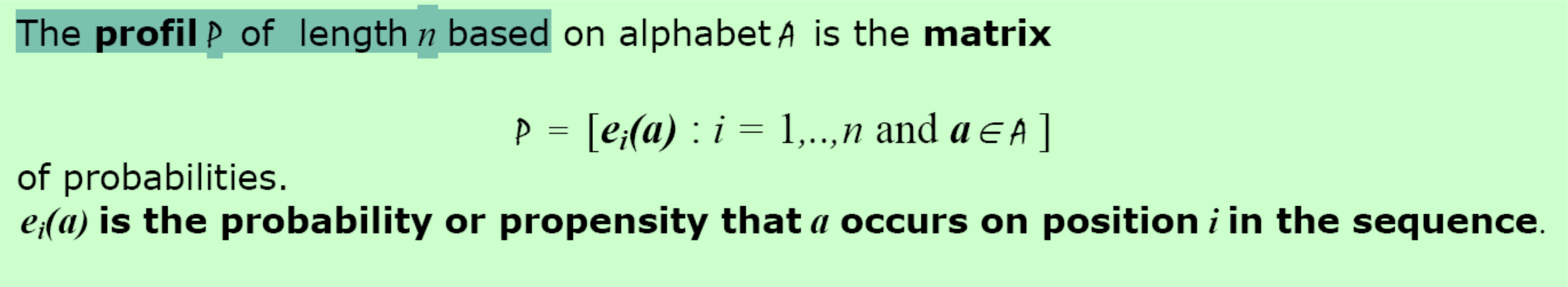


		c1	c2	c3	c4	c5
a b c – a	**a**	.75		.25		.50
a b a b a	**b**		.75		.75	
a c c b -	**c**	.25	.25	.50		.25
c b – b c	-			.25	.25	.25

- Often the values ood ratios. That is, *p(y,j)*, the relative frequency of *y* in column *j*, is replaced by *p(y,j)/p(y)*, where *p(y)* is the relative frequency of y in all the sequences of the multiple alignment.

Profile analysis framework

- Given subsequences which belong to a particular family
- Identify whether a new sequence belongs to that family
- Idea:
  - Align sequences
  - Create “profile” (probabilistic approach)
  - Test new sequences



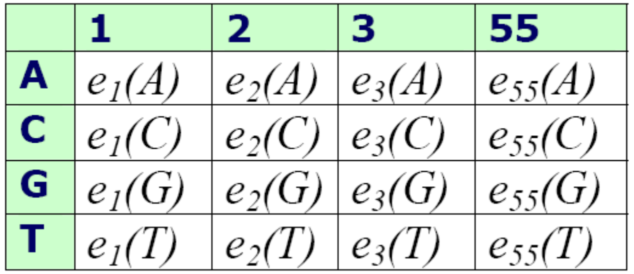
The **profil** $\mathcal{P}$ of length n based on alphabet $\mathcal{A}$ is the **matrix**

$$\mathcal{P} = [\boldsymbol{e_i(a)} : i = 1,..,n \text{ and } \boldsymbol{a} \in \mathcal{A}\ ]$$

of probabilities.

$e_i(a)$ is the probability or propensity that a occurs on position i in the sequence.

Profile analysis framework



	1	2	3	55
A	$e_1(A)$	$e_2(A)$	$e_3(A)$	$e_{55}(A)$
C	$e_1(C)$	$e_2(C)$	$e_3(C)$	$e_{55}(C)$
G	$e_1(G)$	$e_2(G)$	$e_3(G)$	$e_{55}(G)$
T	$e_1(T)$	$e_2(T)$	$e_3(T)$	$e_{55}(T)$

Step 1: Align members of family

```
LEVK
LDIR
LEIK
LDVE
```

l positions, *l* =4 here

Step 2: Compute $f_{i,j}$ = % of column *j* that is amino acid *i* *and*

b_i = % of "background" that is amino acid *i*; and finally

$p_{i,j} = f_{ij} / b_i$

Example: $p_{E,2} = (2/4) / (1/20) = 10$, assuming uniform background

Profile analysis framework

Intuition: $p_{i,j}$ is “**propensity**” for position **(> 1 is favorable, < 1 is unfavorable**);

E is 10x more likely in 2nd position than at random Step 2 gives a $20 \times l$ array of propensities

Step 3: Now to score an l long sequence, say LEVE, compute

$$p_{L,1} \times p_{E,2} \times p_{V,3} \times p_{E,4}$$

– If this is greater than some cutoff, then say “member of the family” otherwise not.
In practice, compute

$$\log(p_{L,1} \times p_{E,2} \times p_{V,3} \times p_{E,4}) = \log(p_{L,1}) + \log(p_{E,2}) + \log(p_{V,3}) + \log(p_{E,4})$$

So set

$$\mathrm{Score(i,j)}= log(p_{i,j})$$

Profile analysis framework

- Example: New sequence LEVEER; find out if it contains motif

- Score each l-long window:
  LEVE, EVEE, VEER
  - Score of LEVE = score L,1+score E,2+score V,3+score E,4
  - Score of EVEE = score E,1+score V,2+score E,3+score E,4
  - Score of VEER = score V,1+score E,2+score E,3+score R,4

- If any of these larger than cutoff, have found motif & position in sequence

- Profiles: Simple probabilistic interpretation of profiles (important in terms of assumptions and for future topics)

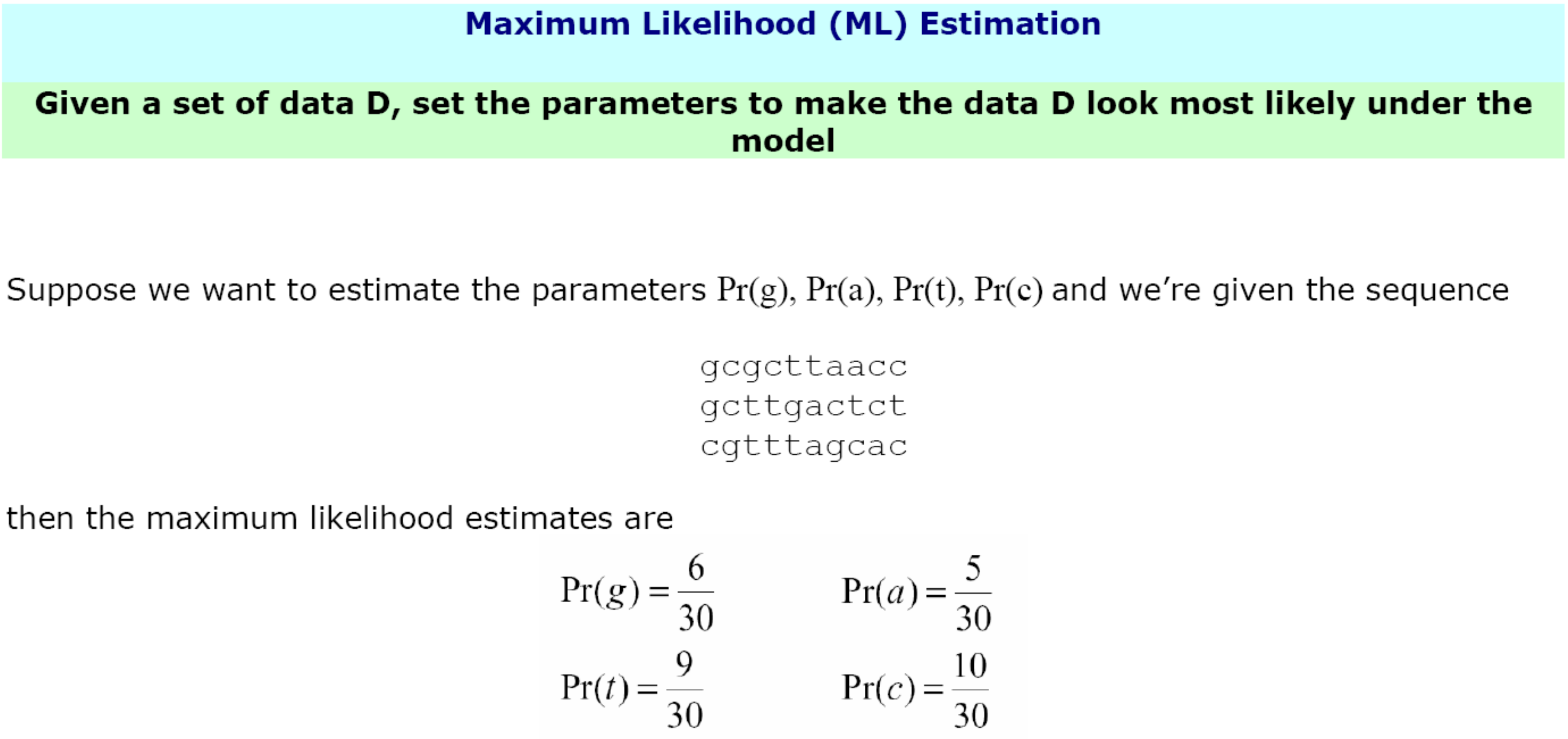
Estimating parameters

Given some data, how can we determine the probability parameters of our model?

Maximum Likelihood (ML) Estimation:

Given a set of data D, set the parameters to make the data D look most likely under the model

Estimating parameters



Maximum Likelihood (ML) Estimation

Given a set of data D, set the parameters to make the data D look most likely under the model

Suppose we want to estimate the parameters Pr(g), Pr(a), Pr(t), Pr(c) and we're given the sequence

```
gcgcttaacc
gcttgactct
cgtttagcac
```

then the maximum likelihood estimates are

$$\Pr(g) = \frac{6}{30} \qquad \Pr(a) = \frac{5}{30}$$

$$\Pr(t) = \frac{9}{30} \qquad \Pr(c) = \frac{10}{30}$$

Estimating parameters

Suppose instead we saw the following sequences

```
gcgcttggcc
gcttggctct
cgttttgctc
```

then the maximum likelihood estimates are

$$\Pr(g) = \frac{9}{30} \qquad \boxed{\Pr(a) = \frac{0}{30}}$$

$$\Pr(t) = \frac{11}{30} \qquad \Pr(c) = \frac{10}{30}$$

Do we really want to set this to 0? Maybe we just got unlucky …

Estimating Parameters: Alternate Approach

- Instead of estimating parameters strictly from the data, we could use Laplace estimates (also known as “add-one rule”)

$$\Pr(a) = \frac{n_a + 1}{\sum_i (n_i + 1)} \xleftarrow{\text{pseudocount}}$$

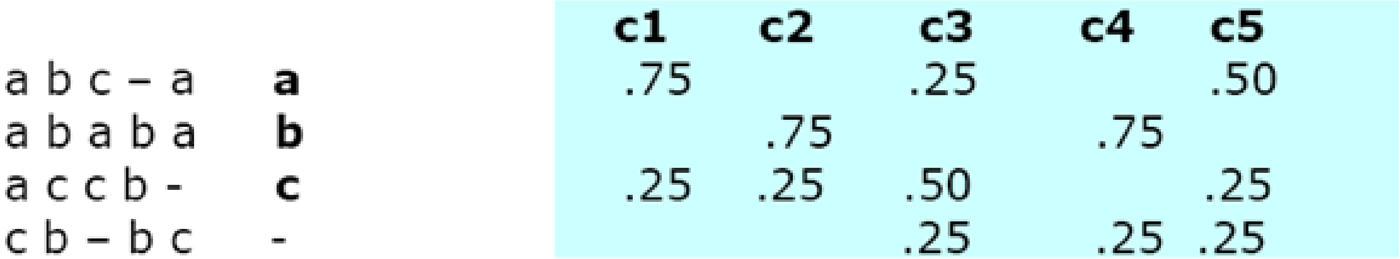
gcgcttggcc
gcttggctct
cgttttgctc

$$\Pr(a) = \frac{0+1}{34}$$

$$\Pr(c) = \frac{10+1}{34}$$

- Bayesian interpretation for this “hack”
- Using Laplace estimates with the sequences - Now nothing is zeroed out.

Estimating Parameters: Alternate Approach



a b c – a
a b a b a
a c c b -
c b – b c

	c1	c2	c3	c4	c5
a	.75		.25		.50
b		.75		.75	
c	.25	.25	.50		.25
-			.25	.25	.25

In the example we aligned four sequences and concluded that in column 1 the probability to observe 'a' was 0.75, 'c' 0.25, 'b' and '-' 0.

- It is possible that truly 'b' and '-' do not appear in column 1 but it is also possible that they appear with a small probability.
- It is even possible that they appear with a relatively high probability and we were especially unlucky with our choice of the four sequences we aligned!
- The classical methods to deal with this problem of limited sampling of the “real” alignment are “pseudocounts”: each symbol is given a small probability at least to appear, say 1% or 0.5%.

Profile alignments

- In Feng & Doolittle's Algorithm all alignments are determined by pairwise sequence alignments.
- Once a group of sequence has been aligned together, it is normally more informative to replace a representative sequence by a profile.
- The weight of adding gaps in new sequences is also more accurately modeled by the profile: positions where many gaps were already inserted are aligned with new gaps at low cost.
- Many progressive alignment algorithms use profiles for combining existing alignments with another alignment or a new sequence.
- The definition of the scoring function used in profile-sequence or profile-profile alignments may differ.
- A SP score often scores aligned amino acids, whereas gaps are handled in more diverse manners.

Profile alignments

- To align sequences with a profile or to align two profiles is a common operation of many multiple alignment algorithms. It also provides an easy and interesting introduction to these algorithms.

- **Sequence to profile alignment:**
  - Given a profile P and a new sequence q, we want to know how well q, or a subsequence of q, fits P. Since space is a legal symbol in P, a fit of q to P should allow spaces.
  - We need a score notion to adapt sequence alignment algorithms.
  - Let us denote by A the alphabet of possible symbols, $p(y,j)$ the probability or the logodds of y in A in column j of the profile, and $s(a,b)$ a substitution matrix extended to the case of a space aligned with an amino acid $(s(-, -)= 0, s(-, a)=-g)$, i.e. a linear gap penalty model.
  - The score to align y with column j of the profile is the weighted substitution matrix score …

Profile alignments

- The score to align *y* with column *j* of the profile is the weighted substitution matrix score:

$$S(y,j) = \sum_{x \text{ in } A} s(y,x)p(x,j)$$

Algorithm:

- The time complexity is *O(amn)*, m the length of q, n the length of P, a=|A| the size of the alphabet (a=21 for proteins). Space complexity is O(mn).

Profile alignments

- **Profile to profile alignment:**
  - Profile to profile alignments are obtained by successively adding entire columns of spaces in the profiles, i.e. columns where the probability to find a space symbol is 1.
  - Again, we need to define a scoring function for aligning column i of the first profile *P1* with column j of the second profile *P2*:

$$T(i,j)=\sum_{x,y \text{ in } A} p_1(x,i)\, p_2(y,j)\, s(x,y).$$

  - It is interesting to note that *T(i,j)* is the sum of pairs score of the *N1* and *N2* sequences that yielded profiles *P1* and *P2* restricted to columns *i* and *j* and scaled by *1/(N1N2)*.

Profile alignments

Algorithm:

- The time complexity is $O(a^2 n_1 n_2)$ and its space complexity is $O(n_1 n_2)$.

ClustalW Package

- ClustalW is a popular heuristic package (1994) for computing MSAs
- Based on progressive alignment

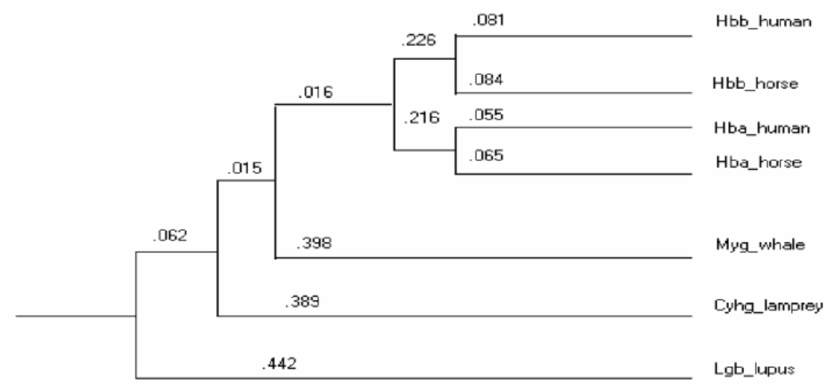
- CLUSTALW Progressive Alignment:
  - Step 1: Calculate a diagonal matrix of N(N-1)/2 distances between all pairs of N sequences by standard pairwise alignment, converting alignment scores to approximate pairwise distances e.g. determine degrees of similarity between each pair (scores ➔ evolutionary distances).
  - Step 2: Construct a guide similarity tree from distances matrix using the neighbor joining clustering algorithm.
  - Step 3: Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment i.e. progressively align at nodes using profiles.

  Sequences are weighted to compensate for biased representation in large subfamilies.

Weighting Sequences

- Note that when aligning alignments, we are just averaging over all sequences
- If some very closely related sequences are evailable, this is problematic (duplicate information)
- Use tree to weight sequences with highly diverged sequences getting larger weights
- Use length from root to sequences to compute weights to increased weights for more divergent species
- If two or more sequences share a branch, length of branch is split amongst sequences to reduced weight for related sequences
- Use these weights when scoring alignments of alignments

Weighting Sequences



Note: Figure not drawn to scale

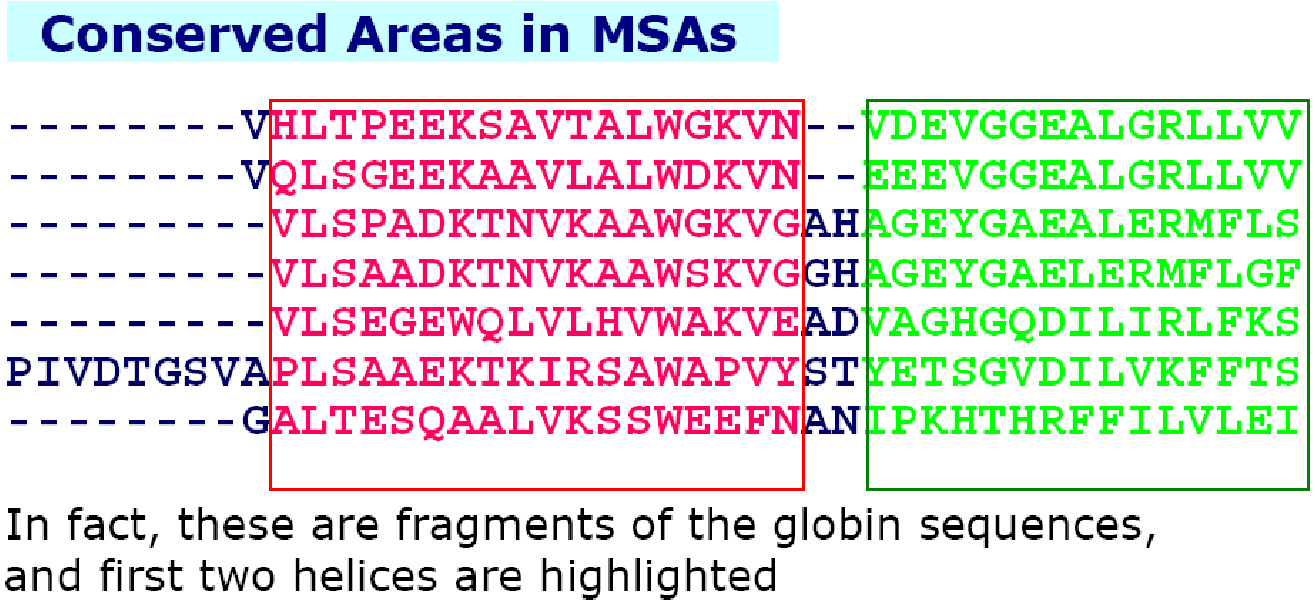
Example: Lgb_lupus: weight of .442 and Hba_human: weight of .055 + .216/2 + .016/4 + .015/5 + .062/6 = .194

Caveats for MSAs and ClustalW

- Progressive alignment says nothing about the optimum MSA (sum-of-pairs or any other measure).
- Initial errors from "once a gap, always a gap" are propagated/compounded
- More than one optimum pairwise alignment possible, yet we are committing ourselves to only one at the outset
- Order in which we add sequences to the alignment (e.g. based on the guide tree) changes alignment.
- If any pair of sequences is less than 25% identical, then the alignments are prone to be bad.
- In general, one needs to correct some alignments manually.

Using MSAs to search for other sequences

- Once we have a MSA, we may want to search for other similar sequences (more sensitivity than pairwise searches)
- Often blocks of conserved regions are observed, sometimes called motifs
- These blocks (or even entire alignments) can be used to make probabilistic profiles that search for similar sequences



Conserved Areas in MSAs

```
--------VHLTPEEKSAVTALWGKVN--VDEVGGEALGRLLVV
--------VQLSGEEKAAVLALWDKVN--EEEVGGEALGRLLVV
---------VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLS
---------VLSAADKTNVKAAWSKVGGHAGEYGAELERMFLGF
---------VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKS
PIVDTGSVAPLSAAEKTKIRSAWAPVYSTYETSGVDILVKFFTS
--------GALTESQAALVKSSWEEFNANIPKHTHRFFILVLEI
```

In fact, these are fragments of the globin sequences, and first two helices are highlighted