Multiple Sequence Alignment

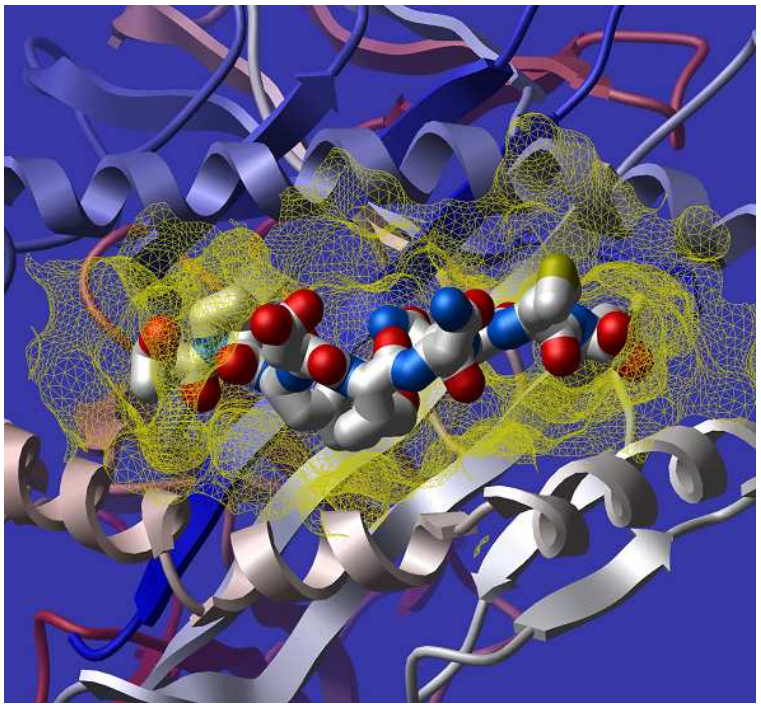
based on Ch. 6 from

Biological Sequence Analysis

by R. Durbin et al., 1998

Acknowledgements:

M.Sc. student Diana Popovici
M.Sc. student Oana Răţoi



[MHC class I with peptide]

MHC = Major Histocompatibility Complex

PLAN

1. Introduction: What a multiple alignment means
2. Scoring a multiple alignment
   - 2.1 general remarks
   - 2.2 sum of pair (SP) scores
   - 2.3 profiles
   - 2.4 position specific (minimum entropy) scores
3. Simultaneous multiple alignment by
   - 3.1 multidimensional dynamic programming;
   - 3.2 Carillo-Lipman/MSA algorithm
4. Heuristic multiple alignment methods
   - 4.1 Divide-et-Impera: Stoye et al.'s algorithm
   - 4.2 Progressive multiple alignment
     - Feng-Doolittle algorithm
     - Profile-based alignment: CLUSTALW
   - 4.3 Iterative refinement multiple alignment methods: Barton-Sternberg algorithm
5. Appendix: Protein structure

1 Introduction

Remember: The goal of biological sequence comparison is to discover functional (or structural) similarities.

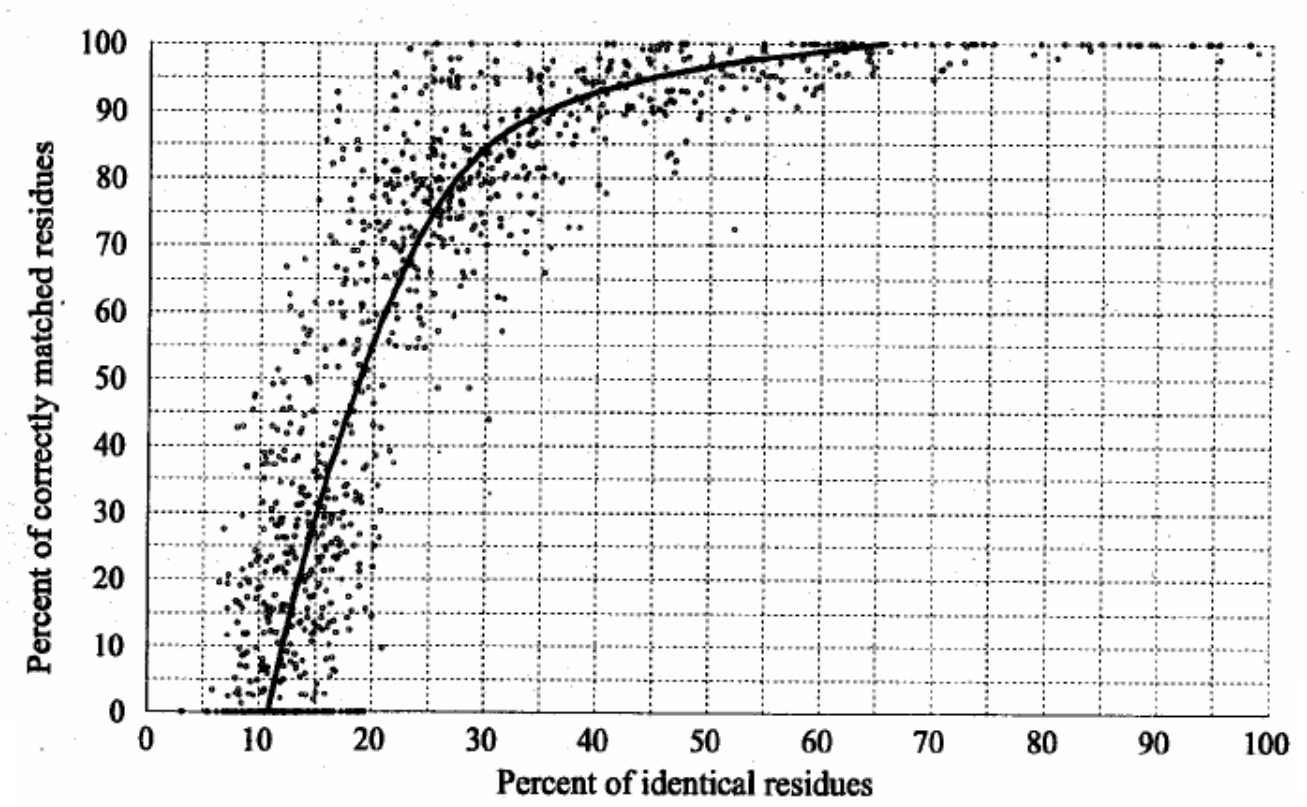
Unfortunately, if the sequence similarity is weak, pairwise alignment can fail to identify biologically related sequences (because weak pairwise similarities may fail the statistical test for significance). Indeed, similar proteins may not exhibit a strong sequence similarity.

The good news is that simultaneous comparison of many sequences often allows one to find similarities that are invisible in pairwise sequence comparison.

[Hubbard et al., 1996]: "Pairwise alignment whispers... multiple alignment shouts out loud."

Pair-wise alignment quality *versus* sequence identity

- Vogt et al., JMB 249, 816-831,1995



Biological sequences are typically grouped into functional families.

Biologists produce high quality multiple sequence alignments by hand using expert knowledge. Important factors are:

- Specific sorts of columns in alignments, such as highly conserved residues or buried hydrophobic residues;
- The influence of the secondary structure (α-helices, β-strands etc. in proteins) and the tertiary structure, the alternation of hydrophobic and hydrophilic columns in exposed β-strands, etc;
- Expected patterns of insertions and deletions, that tend to alternate with blocks of conserved sequence.
- Phylogenetic relationships between sequences, that dictate constraints on the changes that occur in columns and in the patterns of gaps.

A multiple alignment example:

seven globins

Adnotations:

At the top:
α-helices (A-H).

At the bottom:
highly conservative residues (uppercase letter), medium (lowercase letter), or low (dot).

Note the two highly conserved histidines (H): they interact with the oxygene-binding *heme* group in the globine active side.

```
Helix                AAAAAAAAAAAAAAAA   BBBBBBBBBBBBBBBBCCCCCCCCCCC
HBA_HUMAN  ---------VLSPADKTNVKAAWGKVGA--HAGEYGAEALERMFLSFPTTKTYFPHF
HBA_HUMAN  --------VHLTPEEKSACTALWGKV----NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  ---------VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRF
GLB3_CHITP ----------LSADQISTVQASFDKVKG------DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSVAPLSAAEKTKIRSAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU --------GALTESQAALVKSSWEEFNA--NIPKHTHRFFILVLEIAPAAKDLFS-F
GLB1_GLYDI ---------GLSAAQRQVIAATWKDIAGADNGAGVGKDCLIKFLSAHPQMAAVFG-F
Consensus            Ls....  v a W kv .  .    g . L.. f . P .   F  F

Helix          DDDDDDDEEEEEEEEEEEEEEEEEE            FFFFFFFFFFF
HBA_HUMAN  -DLS-----HGSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL-
HBA_HUMAN  GDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHL---D--NLKGTFATLSELHCDKL-
MYG_PHYCA  KHLKTEAEMKASEDLKKHGVTVLTALGAILKK----K-GHHEAELKPLAQSHATKH-
GLB3_CHITP AG-KDLESIKGTAPFETHANRIVGFFSKIIGEL--P---NIEADVNTFVASHKPRG-
GLB5_PETMA KGLTTADQLKKSADVRWHAERIINAVNDAVASM--DDTEKMSMKLRDLSGKHAKSF-
LGB2_LUPLU LK-GTSEVPQNNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG-
GLB1_GLYDI SG----AS---DPGVAALGAKVLAQIGVAVSHL--GDEGKMVAQMKAVGVRHKGYGN
Consensus   .  t    .. . v..Hg KV. a   a...l   d   . a l. l   H  .

Helix        FFGGGGGGGGGGGGGGGGG    HHHHHHHHHHHHHHHHHHHHHHHHH
HBA_HUMAN  -RVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
HBA_HUMAN  -HVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH------
MYG_PHYCA  -KIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
GLB3_CHITP --VTHDQLNNFRAGFVSYMKAHT--DFA-GAEAAWGATLDTFFGMIFSKM-------
GLB5_PETMA -QVDPQYFKVLAAVIADTVAAG---------DAGFEKLMSMICILLRSAY-------
LGB2_LUPLU --VADAHFPVVKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
GLB1_GLYDI KHIKAQYFEPLGASLLSAMEHRIGGKMNAAAKDAWAAAYADISGALISGLQS-----
Consensus    v.   f  l . ..  ....   f    . aa. k..      l sky
```

Another multiple alignment example:

ten I-set immunoglobin superfamily domains

Adnotations:

At the top:
β-strands (a-g).
At the bottom:
identical residues (letter), or highly conservative residues (+).

```
structure:    ...aaaaa...bbbbbbbbbb.....ccccccCCC..C........ddd
1tlk          ILDMDVVEGSAARFDCKVEGY--PDPEVMWFKDDNP--VKESR----HFQ
AXO1_RAT      RDPVKTHEGWGVMLPCNPPAHY-PGLSYRWLLNEFPNFIPTDGR---HFV
AXO1_RAT      ISDTEADIGSNLRWGCAAAGK--PRPMVRWLRNGEP--LASQN----RVE
AXO1_RAT      RRLIPAARGGEISILCQPRAA--PKATILWSKGTEI--LGNST----RVT
AXO1_RAT      ----DINVGDNLTLQCHASHDPTMDLTFTWTLDDFPIDFDKPGGHYRRAS
NCA2_HUMAN    PTPQEFREGEDAVIVCDVVSS--LPPTIIWKHKGRD--VILKKDV--RFI
NCA2_HUMAN    PSQGEISVGESKFFLCQVAGDA-KDKDISWFSPNGEK-LTPNQQ---RIS
NCA2_HUMAN    IVNATANLGQSVTLVCDAEGF--PEPTMSWTKDGEQ--IEQEEDDE-KYI
NRG_DROME     RRQSLALRGKRMELFCIYGGT--PLPQTVWSKDGQR--IQWSD----RIT
NRG_DROME     PQNYEVAAGQSATFRCNEAHDDTLEIEIDWWKDGQS--IDFEAQP--RFV
consensus :   ........G..+.+.C.+.........+.W........+.........++

structure:    ddd.....eeeeee.......ffffffff.......gggggggggggg.
1tlk          IDYDEEGNCSLTISEVCGDDDAKYTCKAVNSL-----GEATCTAELLVET
AXO1_RAT      SQTT----GNLYIARTNASDLGNYSCLATSHMDFSTKSVFSKFAQLNLAA
AXO1_RAT      VLA-----GDLRFSKLSLEDSGMYQCVAENKH-----GTIYASAELAVQA
AXO1_RAT      VTSD----GTLIIRNISRSDEGKYTCFAENFM-----GKANSTGILSVRD
AXO1_RAT      AKETI---GDLTILNAHVRHGGKYTCMAQTVV-----DGTSKEATVLVRG
NCA2_HUMAN    VLSN----NYLQIRGIKKTDEGTYRCEGRILARG---EINFKDIQVIVNV
NCA2_HUMAN    VVWNDDSSSTLTIYNANIDDAGIYKCVVTGEDG----SESEATVNVKIFQ
NCA2_HUMAN    FSDDSS---QLTIKKVDKNDEAEYICIAENKA-----GEQDATIHLKVFA
NRG_DROME     QGHYG---KSLVIRQTNFDDAGTYTCDVSNGVG----NAQSFSIILNVNS
NRG_DROME     KTND----NSLTIAKTMELDSGEYTCVARTRL-----DEATARANLIVQD
consensus :   ..........L.+..+...+.+.Y.C.................+.+.+..
```

What can be done?

Manual multiple alignment is tedious.

Automatic multiple sequence alignment methods are a topic of extensive research in bioinformatics.

Very similar sequences will generally be aligned unambiguously (a simple program can get the alignment right).

For cases of interest (e.g. a family of proteins with only 30% average pairwise sequence identity), there is no objective way to define an unambiguously correct alignment.

In general, an automatic method must assign a score so that better multiple alignments get better scores.

2 Scoring a multiple alignment

2.1 General remarks

A score system for multiple alignment should take into account that:

- the sequences are not independent, but instead related by a *phylogenetic tree* (see Ch. 7);
- some positions are more conserved than others, thus requiring position-specific scoring.

Complex scoring

Goal: Specify a complete probabilistic model of molecular sequence evolution.

Given the correct phylogenetic tree for the sequences to be aligned, the probability for a multiple alignment is the product of the probabilities of all the evolutionary events necessary to produce that alignment via ancestral intermediate sequences times the prior probability for the root ancestral sequence.

The probabilities of evolutionary events would depend on the evolutionary times along each branch of the tree, as well as position-specific structural and functional constraints imposed by natural selection, so that the key residues and structural elements would be conserved.

High-probability alignments would then be good structural and evolutionary alignments under this model.

Unfortunately, we do not have enough data to parametrise such a complex evolutionary model.

Simplifying assumptions

- Partly or (as we did in the previous chapter) entirely ignore the phylogenetic tree.
- Consider that individual columns of an alignment are statistically independent, which leads to

$$S(m) = \sum_i S(m_i)$$

  - Note: most multiple alignment methods use affine gap scoring functions, so succesive gap residues are in fact not treated independently.
- For simplicity, in the sequel we will focus on definitions of $S(m_i)$ for scoring a column of aligned residues with no gaps.

2.2 Sum of Pairs (SP) scores

- As already stated, we assume the statistical independence of columns.
- Columns are scored by a "sum of pairs" (SP) function.
  The SP score for a column is defined as: $S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$, where scores $s(a, b)$ come from a substitution matrix such as BLOSUM or PAM.

Drawbacks:

- There is no probabilistic justification of the SP score.
- Each sequence is scored as if it descended from N-1 other sequences instead of a single ancestor. Evolutionary events are over-counted, a problem which increases as the number of sequences increases (see next slide).

  Altschul, Carroll & Lipman[1989] proposed a weighting scheme designed to partially compensate for this defect in SP scores.

A problem with SP scores: Example

- Consider an alignment of N sequences which all have leucine (L) at a certain position. The score of an L aligned to L is 5 (BLOSUM), so the score of the column is 5xN(N-1)/2, where N(N-1)/2 is the number of symbol pairs in the column.
- If there were one glycine (G) in the column and N-1 Ls, the score would be 9x(N-1) less, because a G-L pair scores -4 and N-1 pairs are affected.
- So, the SP score for a column with one G is worse than the score for a column of all Ls by a fraction of $\frac{9(N-1)}{5N(N-1)/2} = \frac{18}{5N}$.
- Notice the inverse dependence on N: the relative difference in score between the correct alignment and the incorrect alignment decreases with the number of sequences in the alignment. This is counter-intuitive, because the relative difference ought to increase with the more evidence we have for a conserved leucine.

Aligning 2 MAs using SP scoring with linear gaps

(A case study)

The gap scores can be included in the SP score by setting

$s(-, a) = s(a, -) = -g$ **and** $s(-, -) = 0$**.**

Here an alignment of two MAs will be done so that gaps are inserted in whole columns, so the alignment within each one of the two MAs is not changed.

Assuming that we have two MAs, one containing sequence 1 to n, and the other containing sequence $n+1$ to N, the global alignment score is:

$$\begin{aligned}\sum_i S(m_i) = \quad & \sum_i \sum_{k<l\leq N} s(m_i^k, m_i^l) = \\ & \sum_i \sum_{k<l\leq n} s(m_i^k, m_i^l) + \sum_i \sum_{n<k<l\leq N} s(m_i^k, m_i^l) + \\ & \sum_i \sum_{k\leq n, n<l\leq N} s(m_i^k, m_i^l)\end{aligned}$$

Aligning 2 MAs using SP scoring with linear gaps (cont'd)

Note that the first two sums are unaffected by the global alignment, since adding columns of gap characters to a MA adds 0 to the score ($s(-,-) = 0$).

Therefore the **optimal alignment** of the two MAs can be obtained by only **optimising the last sum** with the cross terms. This can be done exactly **like standard pairwise alignment**, where columns are scored against columns by adding pair scores.

Obviously, one of the MAs can consist of a single sequence only, which corresponds to **aligning a single sequence to a MA**.

Remark

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.

- **The degree of sequence conservation at each position should be taken into account and mismatches at highly conserved positions penalized more stringently than mismatches at variable positions.**
- **Gap penalties might be reduced where lots of gaps occur in the cluster alignment, and increased where no gaps occur.**

2.3 Profiles

(following [Gusfield, 1999])

Definition:

Given a multiple sequence alignment, a **profile** for that alignment is a matrix that specifies for each column the frequency with which each character appears in that column. (Also called **weight matrix**, or **position-specific score matrix**.)

Example:

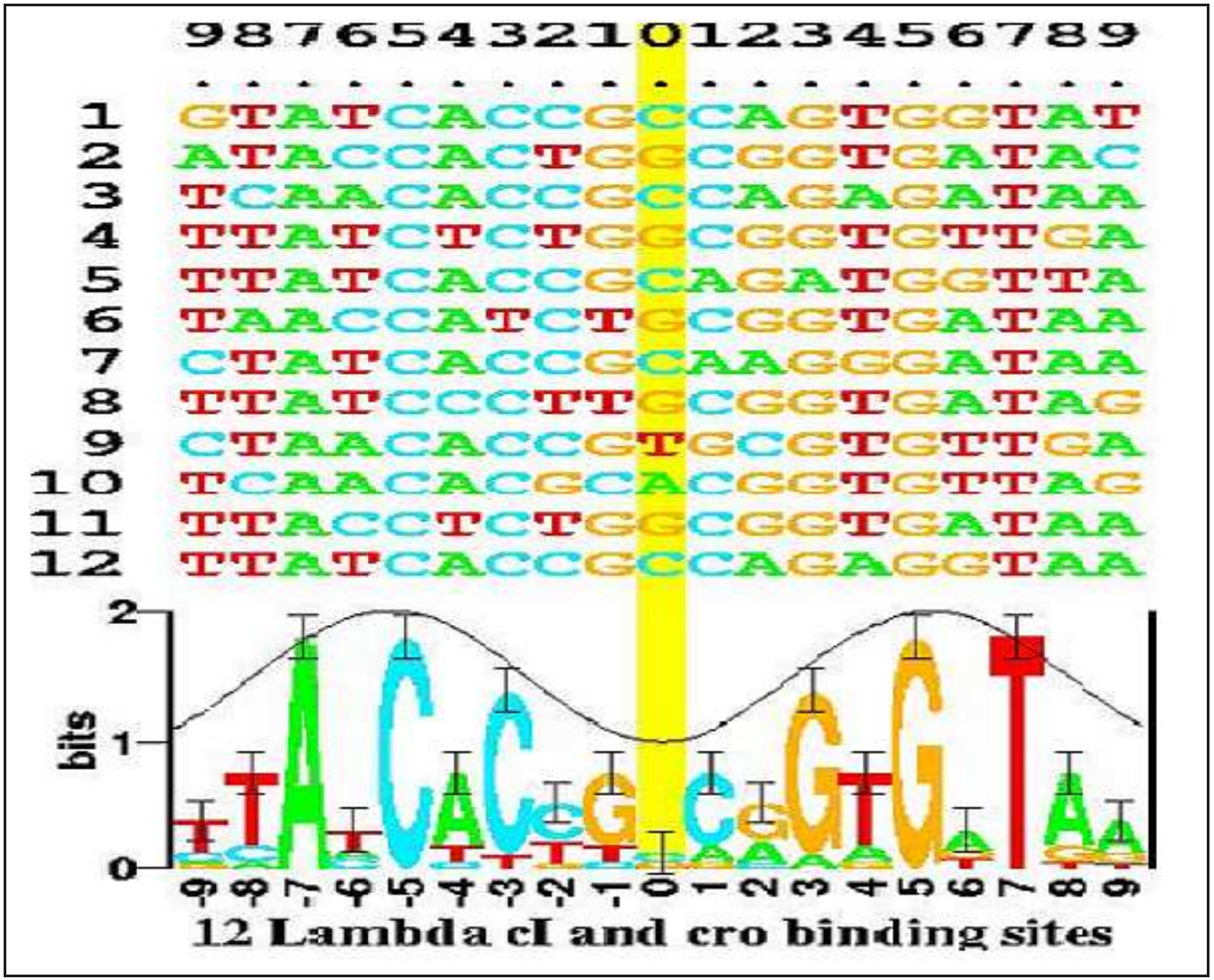
A multiple sequence alignment and the profile generated from it:

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

Consensus string for this profile: ***abcba***

Representing a profile as a logo



(http://www-lmmb.ncifcrf.gov/ toms/sequencelogo.html)

How to score a string-to-profile alignment: Illustrating the idea

Given an aligment of the string *aabbc* to the column positions of the previous sequence alignment:

$$\begin{array}{cccccc} a & a & b & - & b & c \\ 1 & - & 2 & 3 & 4 & 5 \end{array}$$

assuming that $s(a,a) = 2,\ s(a,b) = -1,\ s(a,c) = -3,\ s(a,-) = -1$**,**

the first two columns in the above string-to-profile alignment contribute to the overall alignment score with $0.75 \times 2 - 0.25 \times 3 - 1$**.**

How to score a string-to-profile alignment

using dynamic programming:

Initialisation: $V(0,j) = \sum_{k \le j} S(-,k)$ **and** $V(i,0) = \sum_{k \le i} s(x_k,-)$

Recursion:

$$V(i,j) = \max \begin{cases} V(i-1,j-1), +S(x_i,j) \\ V(i-1,j), +s(x_i,-) \\ V(i,j-1), +S(-,j) \end{cases}$$

where

- $s(a,b)$ **is the score of aligning characters** a **and** b **in the pure string alignment problem**
- $p(b,j)$ **denotes the frequency of the character** b **appearing in the column** j **of the profile**
- $S(a,j) = \sum_b [s(a,b) \times p(b,j)]$
- $V(i,j)$ **denotes the value of the optimal alignment of the substring** $x_1 \ldots x_i$ **with the first** j **columns of the given profile.**

Complexity: $\mathcal{O}(\sigma nm)$**, where** σ **is the size of the alphabet,** n **is the length of the sequence, and** m **is the number of columns in the profile.**

Remark

It is straightforward to formalize optimal profile to profile alignment and to obtain the recurrence relations to compute it.

2.4 Position specific (minimum entropy) scores

Notations

- m **is a multiple alignment;**
  m_i **the column** $m_i^1, \ldots, m_i^N$**of aligned symbols in column** i**;**
  m_i^j **the symbol in column** i **for sequence** j**;**
- c_{ia} **is the observed counts for residue a in column** i**;**
  $c_{ia} = \sum_i \delta\left(m_i^j = a\right)$ **where** $\delta(m_i^j = a)$ **is 1 if** $m_i^j = a$ **and 0 otherwise**
- c_i **the count vector** $c_i^1, \ldots, c_i^K$ **of observed symbols in column** i **for an alphabet of K different residues**

- We assume that residues within the column are independent, as well as between columns.

Definition

- The probability of a column m_i is:
  $P(m_i) = \prod_a p_{ia}^{c_{ia}}$
  where p_{ia} is the probability of residue a in column i.

- We define a column score as:
  $S(m_i) = -\log P(m_i) = -\sum_a c_{ia} \log p_{ia}$

  The column score is an entropy measure.
  A conserved column would score 0.

- The maximum likelihood estimate for the parameter p_{ia} is
  $p_{ia} = \frac{c_{ia}}{\sum_{a'} c_{ia'}}$.

Remarks

1. Profile HMMs (see Durbin et al., 1998, Ch. 5) extend this entropy-based score by probabilistically modeling insertions and deletions in the multiple alignment.

2. In return for giving up the evolutionary tree and assuming independence between sequences, we gain the ability to straightforwardly estimate a position specific model of both residue probabilities in columns and insertions and deletions.

3 Simultaneous multiple sequence alignment by Multidimensional dynamic programming

Assumption:

- the columns of an alignment are statistically independent
- gaps are scored with a linear gap cost $\gamma = gd$ for a gap of length g and some gap cost d.

  Note: Extension to affine gap costs is possible but the formalism becomes tedious.

Therefore the overall score for an alignment can be computed as the sum of the scores for each column i: $S(m) = \sum_i S(m_i)$.

3.1 Extending the Needleman-Wunsch algorithm

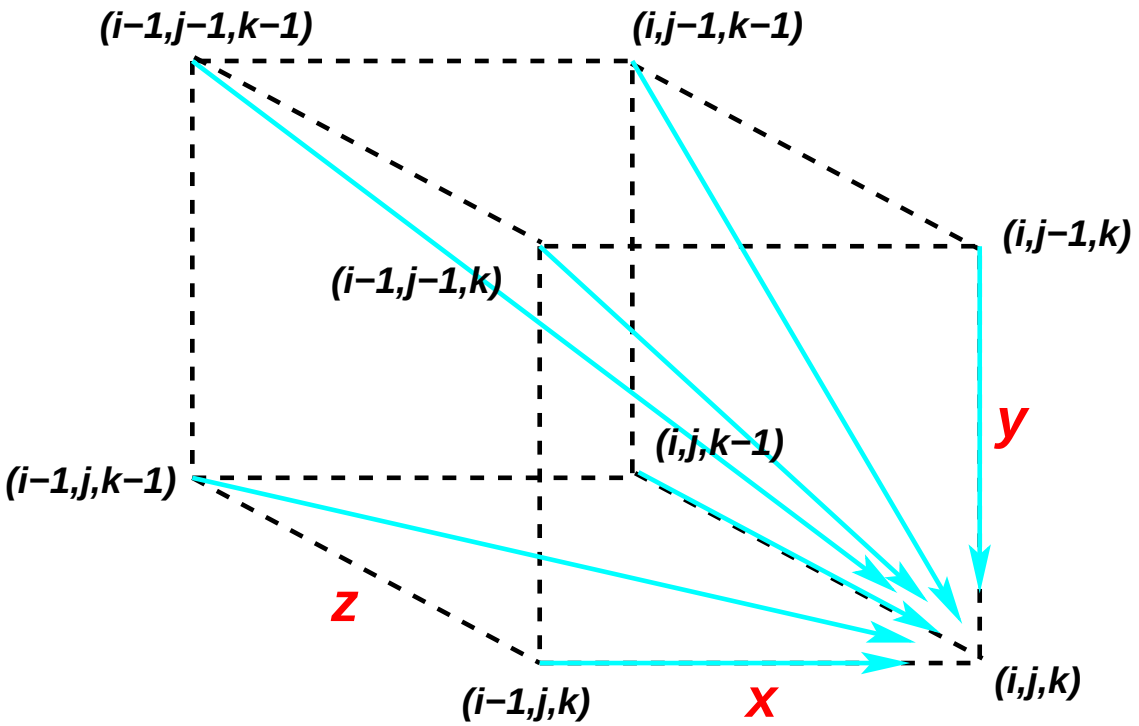
Define $\alpha_{i_1,i_2,...i_N}$ as the maximum score of an alignment up to the subsequences ending with $x^1_{i1}, ..., x^N_{iN}$.

Recurrence relation:

$$\alpha_{i_1,...,i_N} = max \begin{cases} \alpha_{i_1-1,i_2-1,...,i_N-1} & + S(x^1_{i_1}, x^2_{i_2}, \ldots, x^N_{i_N}), \\ \alpha_{i_1,i_2-1,...,i_N-1} & + S(-, x^2_{i_2}, \ldots, x^N_{i_N}), \\ \alpha_{i_1-1,i_2,...,i_N-1} & + S(x^1_{i_1}, -, \ldots, x^N_{i_N}), \\ & \vdots \\ \alpha_{i_1-1,i_2-1,...,i_N} & + S(x^1_{i_1}, x^2_{i_2}, \ldots, -), \\ \alpha_{i_1,i_2,...,i_N-1} & + S(-, -, \ldots, x^N_{i_N}), \\ & \vdots \\ \alpha_{i_1,i_2-1,...,i_N} & + S(-, x^2_{i_2}, \ldots, -), \\ & \vdots \end{cases}$$

Note: **The functional form of the column score $S(m_i)$ is left unspecified. For instance it could be calculated using an evolutionary model (e.g. [Sankoff, 1975], described in Durbin et al, 1998, Ch. 7), or SP (sum of pairs) scores.**

The dynamic programming matrix for 3 sequences



Note

Using the notation
$\Delta_i \cdot x = x$ **if** $\Delta_i = 1$, **and** $\Delta_i \cdot x = -$ **if** $\Delta_i = 0$,
the recursion relation becomes:

$$\alpha_{i_1,\ldots,i_N} = \max_{\Delta_1+\ldots+\Delta_N>0} \{\alpha_{i_1-\Delta_1,\ldots,i_N-\Delta_N} + S(\Delta_1 x^1_{i_1}, \ldots, \Delta_N x^N_{i_N})\}$$

Complexity

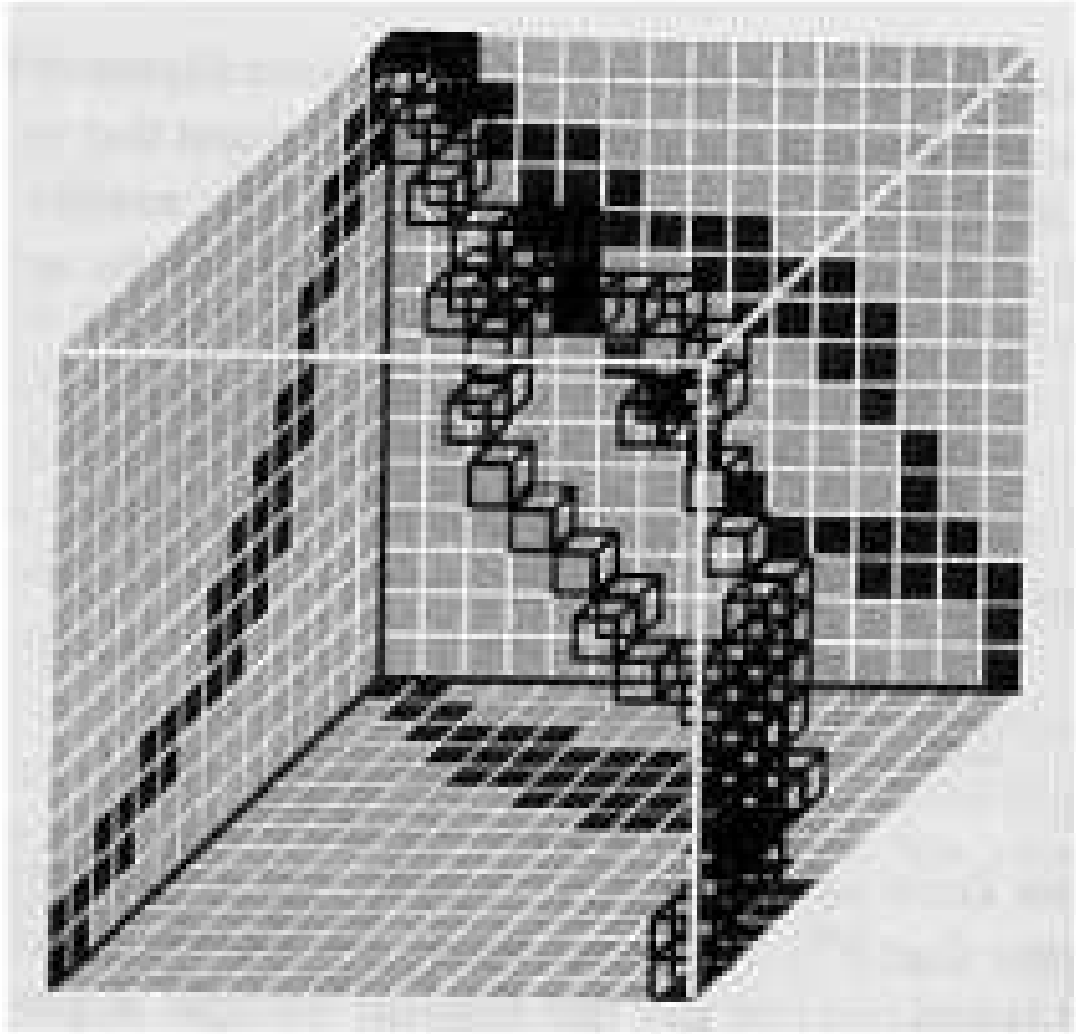
In general, if we assume that the sequences are roughly the same length $\bar{L}$, **the memory complexity of the (naive) dynamic programming algorithm for multiple sequence alignment is** $O(\bar{L}^N)$, **and the time complexity is** $O(2^N \bar{L}^N)$, **therefore in practice it is almost unusable.**

3.2 Carillo-Lipman Algorithm (1988)

Implementation: MSA by Lipman, Altschul & Kececioglu (1989)

- This algorithm reduces the volume of the multidimensional dynamic programming matrix.
- MSA can optimally align up to five to seven protein sequences of reasonable length (200-300 residues).
- Assumption: the score of a multiple alignment is the sum of the scores of all pairwise alignments defined by the multiple alignment

Illustrating the idea of Carillo-Lipman algorithm for 3 sequences



- **The score of a complete alignment a is defined as $S(a) = \sum_{k<l} S(a^{kl})$, where a^{kl} denotes the pairwise alignment between sequences k and l.**
- **Let $\hat{a}^{kl}$ be the optimal pairwise alignment of k, l. Obviously, $S(a^{kl}) \leq S(\hat{a}^{kl})$.**
- **Assume that we have a lower bound $\sigma(a)$ on $S(a)$, the score of the optimal multiple alignment a, i.e. $\sigma(a) \leq S(a)$.**

  (We can obtain a good bound $\sigma(a)$ by any fast heuristic multiple alignment algorithm, for instance progressive alignment algorithms, to be introduced in the sequel.)
- **Due to the sum of pairs (SP) score definition, we have:**

  $$S(a) = \sum_{k'<l'} S(a^{k'l'}) \leq S(a^{kl}) - S(\hat{a}^{kl}) + \sum_{k'<l'} S(\hat{a}^{k'l'})$$

  and thus $\sigma(a) \leq S(a^{kl}) - S(\hat{a}^{kl}) + \sum_{k'<l'} S(\hat{a}^{k'l'})$
- **Therefore we can set a lower bound on $S(a^{kl})$:**

  $S(a^{kl}) \geq \beta^{kl}$**, where** $\beta^{kl} = \sigma(a) + S(\hat{a}^{kl}) - \sum_{k'<l'} S(\hat{a}^{k'l'})$

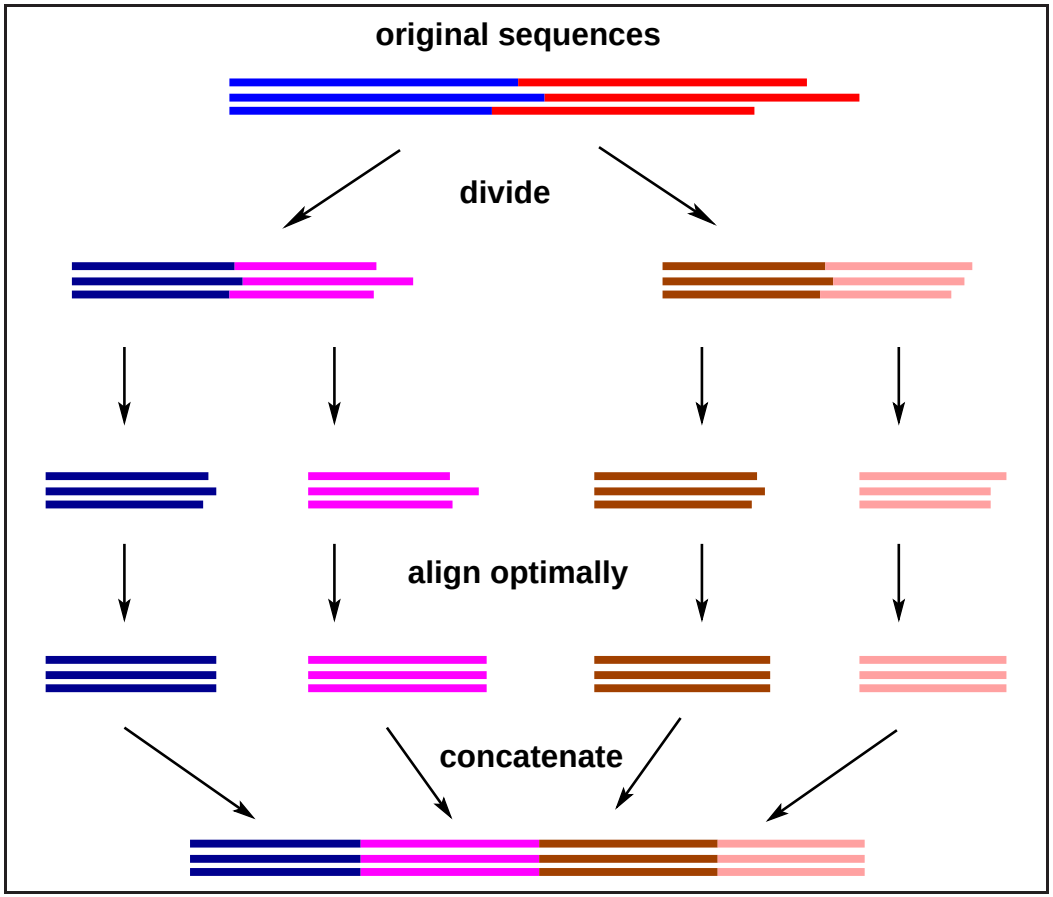
- **The N(N-1)/2 optimum pairwise alignments $\hat{a}^{lk}$ are each calculated and scored by standard pairwise alignment.**
- **Note: The higher the β^{kl} bounds are, the smaller the volume of multidimensional dynamic programming matrix that must be calculated.**
- **For each pair k, l we can find the complete set B^{kl} of coordinate pairs (i_k, i_l) such that the best alignment of x^k to x^l through (i_k, i_l) scores more than β^{kl}. This set is calculated in $O(\bar{L}^2)$ time by multiplying the Viterbi scores (for prefixes and reversed suffixes) for each cell of the complete pairwise dynamic programming table.**
- **The costly multidimensional dynamic programming algorithm can then be restricted to evaluate only cells in the intersection of these B^{kl} sets: i.e. cells $(i_1, i_2, \ldots, i_N)$ for which (i_k, i_l) is in B^{kl} for all k, l.**

Example: pr. 6.2 in [Borodovsky, Ekisheva, 2006]

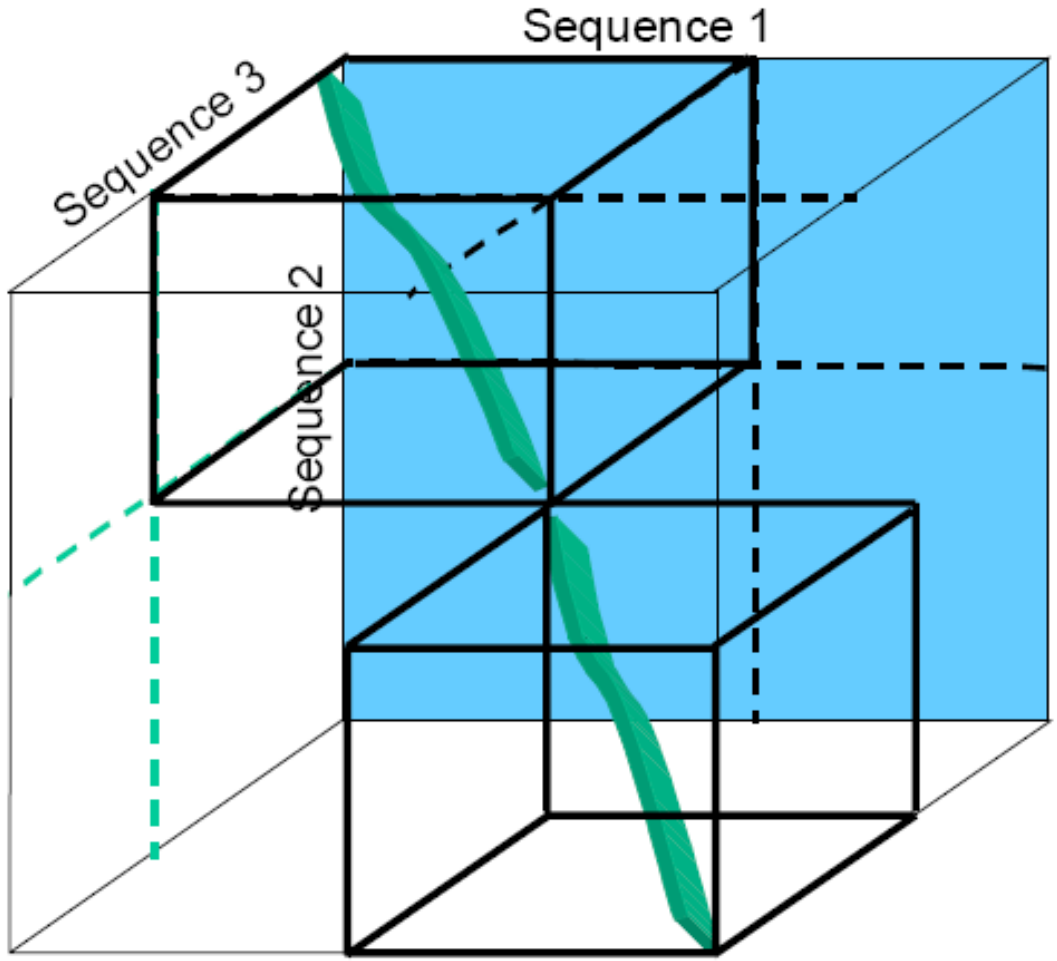
4 Heuristic multiple alignment methods

4.1 Divide et Impera [Stoye at al., 1997]

- each sequence is cut in two behind a suitable cut position somewhere close to its mid-point
- therefore the problem of aligning one family of (long) sequences is divided into the two problems of aligning two families of (shorter) sequences
- this procedure is re-iterated until the sequences are sufficiently short
- optimal alignment by MSA
- finally, the resulting short alignments are concatenated



So, in effect ...



Acknowledgement:

This slide and the previous one are from the Sequence Analysis Master Course, Centre for Integrative Bioinformatics, Vrije Universiteit, Amsterdam

4.2 Progressive multiple alignment methods

These (greedy) methods are the most commonly used approach to multiple sequence alignment. The general idea:

- Most progressive alignment algorithms use a "guide tree", a binary tree whose leaves represent sequences and whose interior nodes represent alignments.
  (The methods for constructing guide trees can be "quick and dirty" versions of those for phylogenetic trees.)
- Main heuristic: first align the most similar pairs of sequences, using a pairwise alignment method. Then walk up the tree and compute at each interior node the alignment of (alignments of) sequences associated with the direct descendants of that node.
- The root node will represent a complete multiple alignment of the input sequences.

Progressive alignment methods use no global scoring function of alignment correctness.

4.2.1 Progressive MA algorithm (Feng-Doolittle, 1987)

- **The guide tree is constructed using the clustering algorithm by Fitch & Margoliash (1967), starting from a distance matrix obtained by converting pairwise alignment scores to (approximate) pairwise distances:**

$$D = -\log S_{\mathit{eff}} = -\log \frac{S_{\mathit{obs}} - S_{\mathit{rand}}}{S_{\mathit{max}} - S_{\mathit{rand}}}$$

where

S_{obs} **is the observed pairwise alignment score;**

S_{max} **is the maximum score, the average of the score of aligning either sequence to itself;**

S_{rand} **is the expected score for aligning two random shufflings of the two sequences (or by an approximate calculation given in [Feng & Doolittle, 1996]).**

Note: The effective score S_{eff} can be viewed as a normalized percentage similarity; it is expected to decay exponentially towards 0 with increasing evolutionary distance, hence the $-\log$ to make the measure more approximately linear with evolutionary distance.

Feng-Doolittle algorithm (Cont'd)

- **Sequence to group alignments:**

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

- **Group to group alignments:**

  All sequence pairs between the two groups are tried; the best pairwise sequence alignment determines the alignment of the two groups.

- After an alignment is completed, gap symbols are replaced with a **neutral X character**. The cost for aligning an X with anything (including a gap symbol) is 0, hence a desirable effect (**"once a gap always a gap"**) is obtained: gaps (tend to) occur in the same columns in subsequent pairwise alignments.

Example: pr. 6.3 in [Borodovsky, Ekisheva, 2006]

4.2.2 Profile-based progressive alignment: The CLUSTALW algorithm

[Thompson, Higgins, Gibson, 1994]

- Construct a distance matrix of all N(N-1)/2 pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity scores to evolutionary distances using the model of Kimura [1983].
- Construct a guide tree by using the Neighbour-Joining clustering algorithm [Saitou & Nei, 1987], see Durbin et al, 1998, Ch. 7.
- Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.

Additional heuristics
contributing to CLUSTALW's accuracy

- In order to compensate for biased representation in large subfamilies, individual sequences are weighted according to the branch length in the Neighbour-Joining tree.
- The substitution matrix is chosen on the basis of the similarity expected of the alignment, e.g. BLOSUM80 for closely related sequences, and BLOSUM50 for distant sequences.
- Position-specific gap-open profile penalties are multiplied by a modifier that is a function of the residues observed at the position.
- Both gap-open and gap-extend penalties are increased if there are no gaps in a column but gaps occur nearby in the alignment. This rule tries to force all the gaps to occur in the same places in an alignment.
- In the progressive alignment stage, if the score of an alignment is low, the guide tree may be adjusted on the fly to defer the low-scoring alignment until later in the progressive alignment phase when more profile information has been accumulated.

4.3 Iterative refinement methods for multiple sequence alignment

A **problem** with the previous heuristic alignment methods:

The subalignments are 'frozen', i.e. once a group of sequences has been aligned, their alignment to each other cannot be changed at a later stage as more data arrive.

Example: align (x, y), (z, w), (xy, zw)

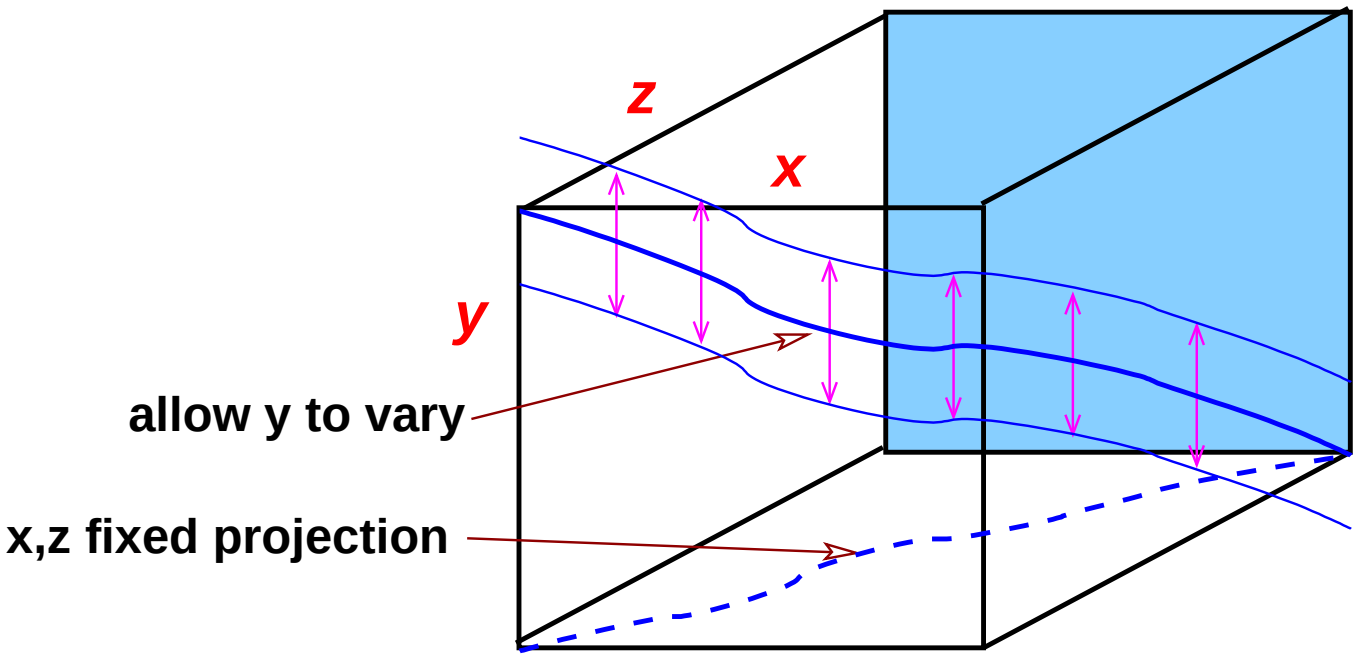
x :	GAAGTT	frozen!
y :	GAC−TT	
z :	GAACTG	
w :	GTACTG	now clearly we have to correct $y =$ GA−CTT

Basic idea
for iterative refinement MA methods

- An initial alignment is generated.
- Then one sequence (or a set of sequences) is taken out and realigned to a profile of the remaining aligned sequences. If a meaningful(!) score is being optimised, this either increases the overall score or results in the same score.

  Another sequence is chosen and realigned, and so on, until the alignment does not change.
- The procedure is guaranteed to converge to a local maximum of the score provided that all the sequences are tried, and a maximum score exists simply because the sequence space is finite.

Illustrating the idea of iterative multiple alignment



Acknowledgement:

This slide is from Serafim Batzoglou, Bioinformatics Course, Stanford University.

Barton-Sternberg algorithm (1987)

- **Find the two sequences with the highest pairwise similarity and align them using standard pairwise dynamic programming alignment.**

  Find the sequence that is most similar to a profile of the alignment of the first two, and align it to them by profile-sequence alignment. Repeat until all sequences have been included in the MA.

- **Remove the sequence x_1 and realign it to a profile of the other aligned sequences by profile-sequence alignment. Repeat this step for the sequences $x_2, \ldots, x_N$.**

- **Repeat the previous realignment step for a fixed number of times, or until the alignment score converges.**

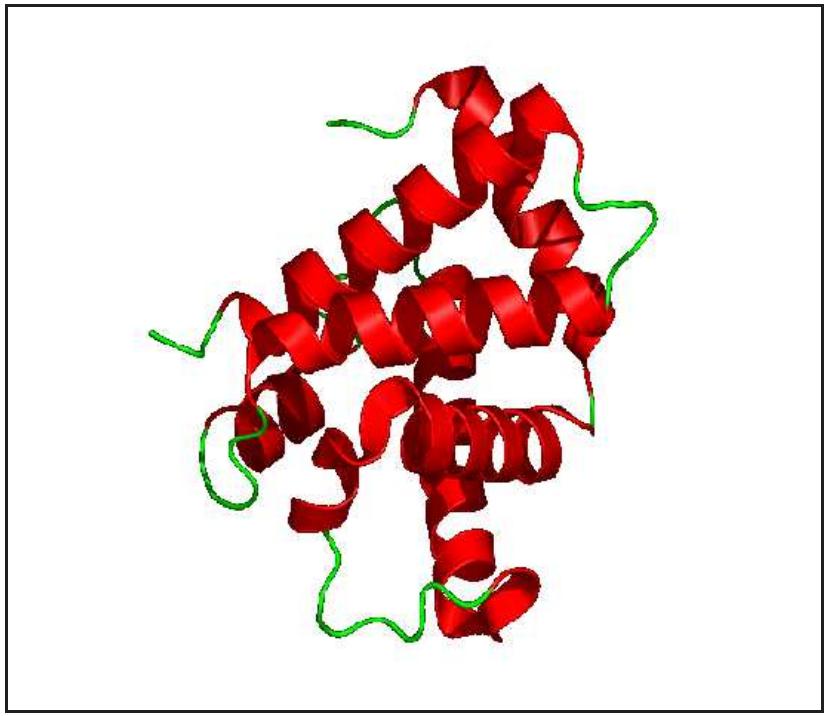
APPENDIX: Protein Structure

slides by Pemille Haste Andersen and Thomas Blicher

Center for Biological Sequence Anlysis
Technical University of Denmark

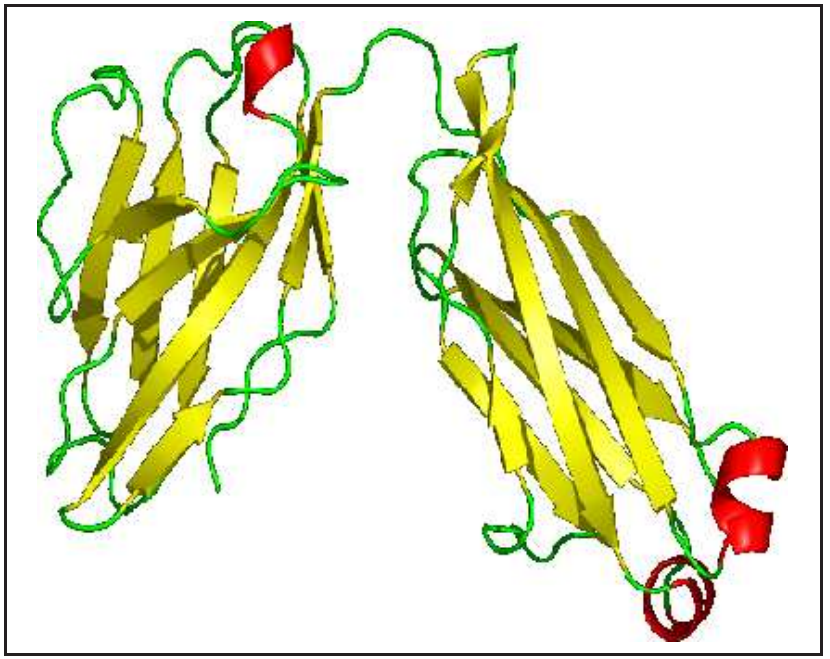
Major protein classes (SCOP database)

all α



hemoglobin (1BAB)

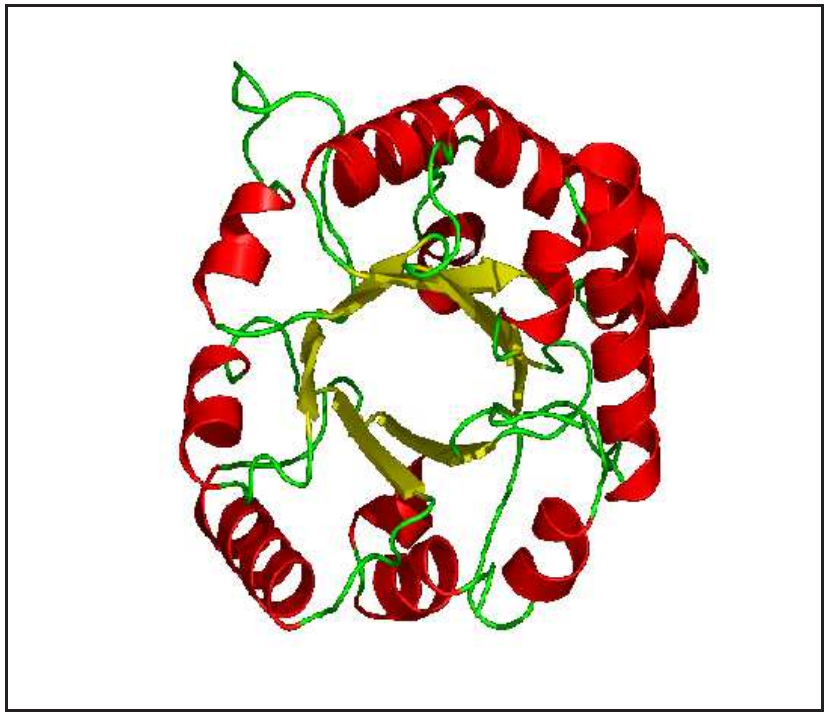
all β



immunoglobulin (8FAB)

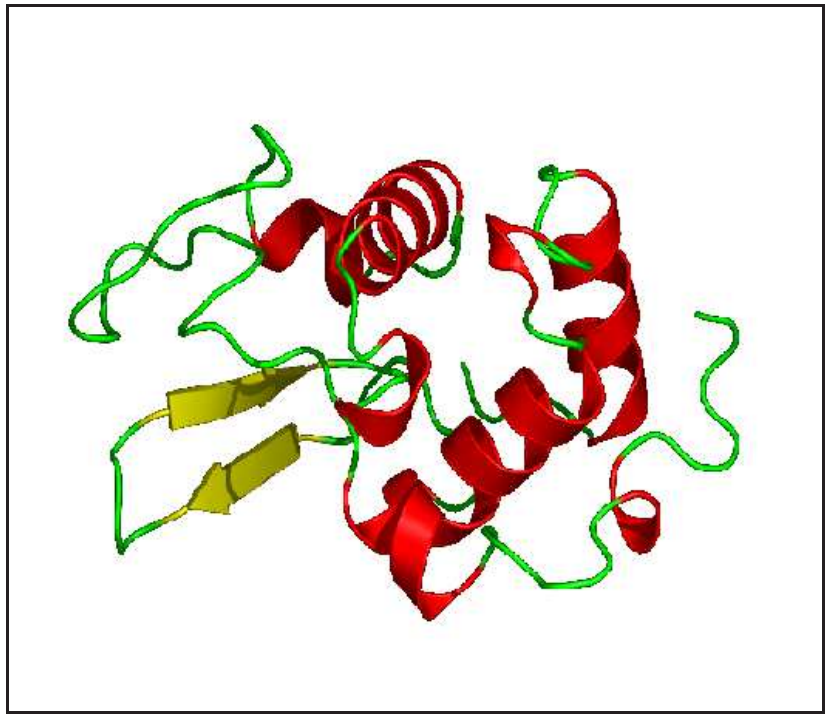
Major protein classes (cont'd)

α/β



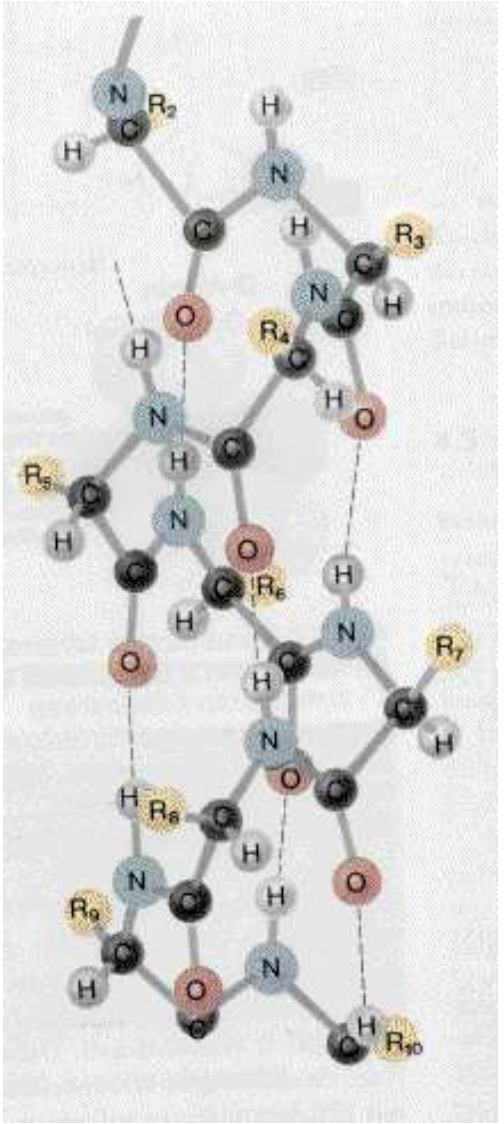
triose phosphate isomerase (1HTI)

$\alpha + \beta$

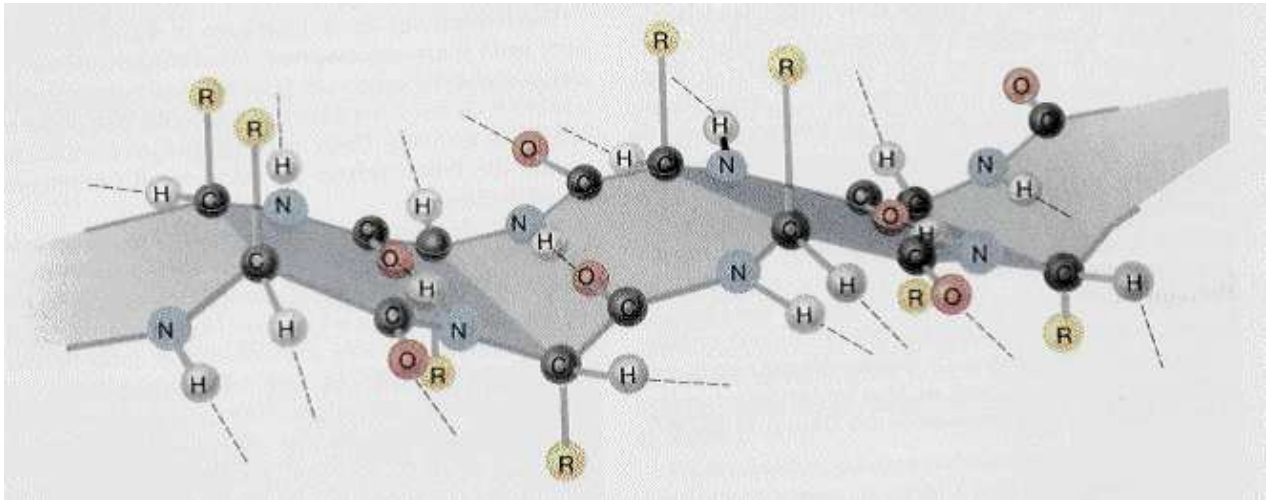


lysozyme (1JSF)

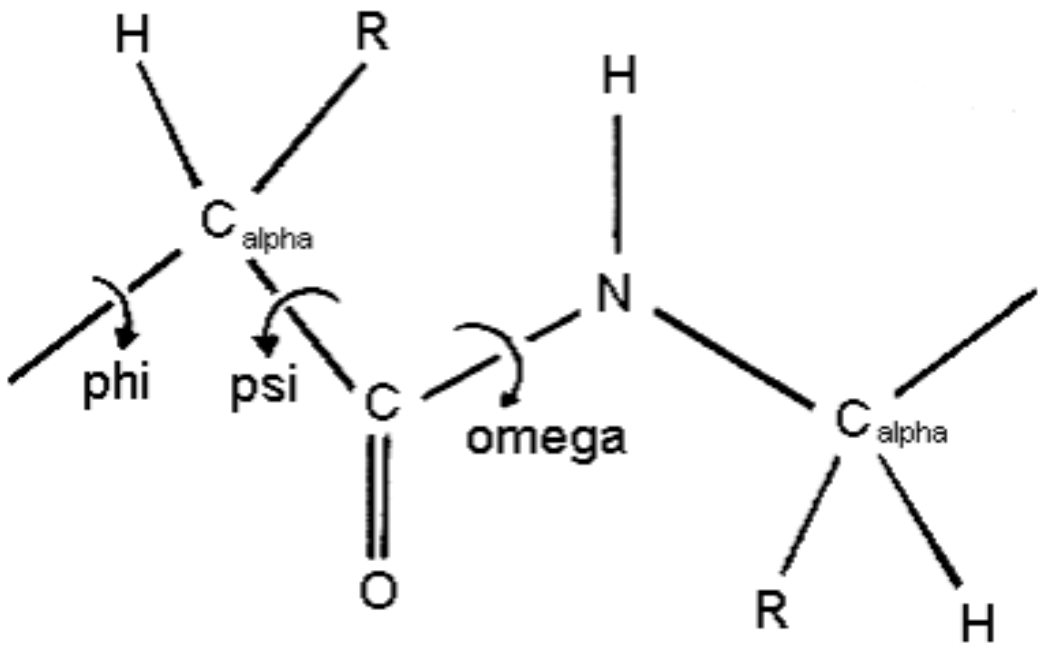
α-helix



β-strand



Dihedral angles in proteins



phi – dihedral angle about N–C_{alpha} bond
psi – dihedral angle about C_{alpha}–C bond
omega – dihedral angle about C–N (peptide) bond

Main secondary structure elements

Helices	***phi*** **(deg)**	***psi*** **(deg)**	**H-bond pattern**
right-handed alpha-helix	−57.8	−47.0	**i+4**
pi-helix	−57.1	−69.7	**i+5**
3_{10} helix	−47.0	−4.0	**i+3**

(***omega*** **is 180 deg in all cases)**

Beta Strands	***phi*** **(deg)**	***psi*** **(deg)**	***omega*** **(deg)**
	−120	120	180