Chapter 5

Multiple Sequence Alignment

Overview

1. Introduction
2. Introduction to Biological Databases
3. Pairwise Sequence Alignment
4. Database Similarity Searching
5. Multiple Sequence Alignment
6. Profiles and Hidden Markov Models
7. Protein Motifs and Domain Prediction
8. Gene Prediction
9. Promoter and Regulatory Element Prediction
10. Phylogenetics Basics
11. Phylogenetic Tree Construction Methods and Programs
12. Protein Structure Basics
13. Protein Structure Visualization, Comparison and Classification
14. Protein Secondary Structure Prediction
15. Protein Tertiary Structure Prediction
16. RNA Structure Prediction
17. Genome Mapping, Assembly and Comparison
18. Functional Genomics
19. Proteomics

Multiple sequence alignment (MSA)

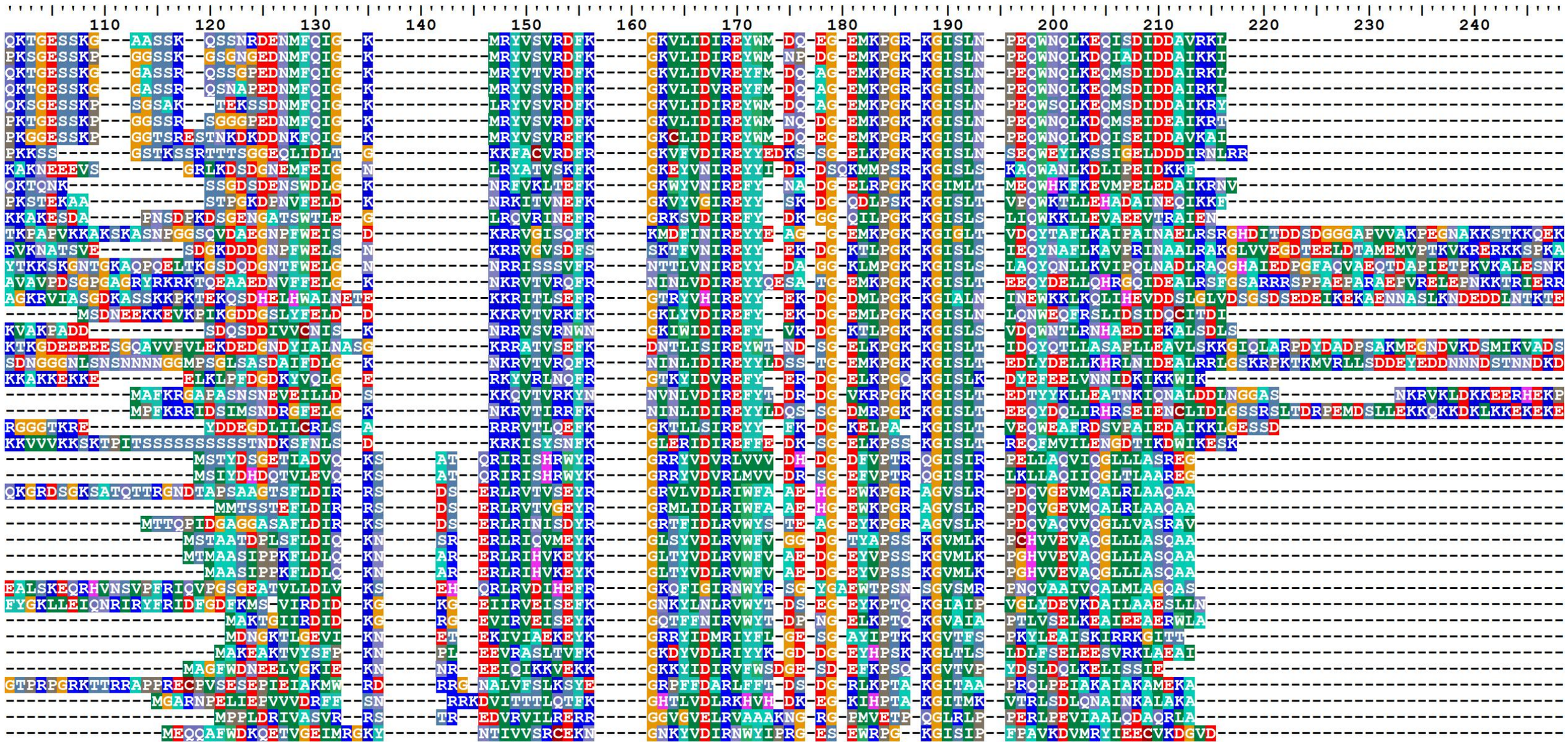
- "Natural" extension of pairwise alignment
- Arrangement of many sequences into a single alignment, in such a way that evolutionary equivalent positions across all sequences are matched
- Reveals more biological information than pairwise alignments can do:

✓ Conserved sequence patterns and functionally critical amino acid residues stand out more clearly

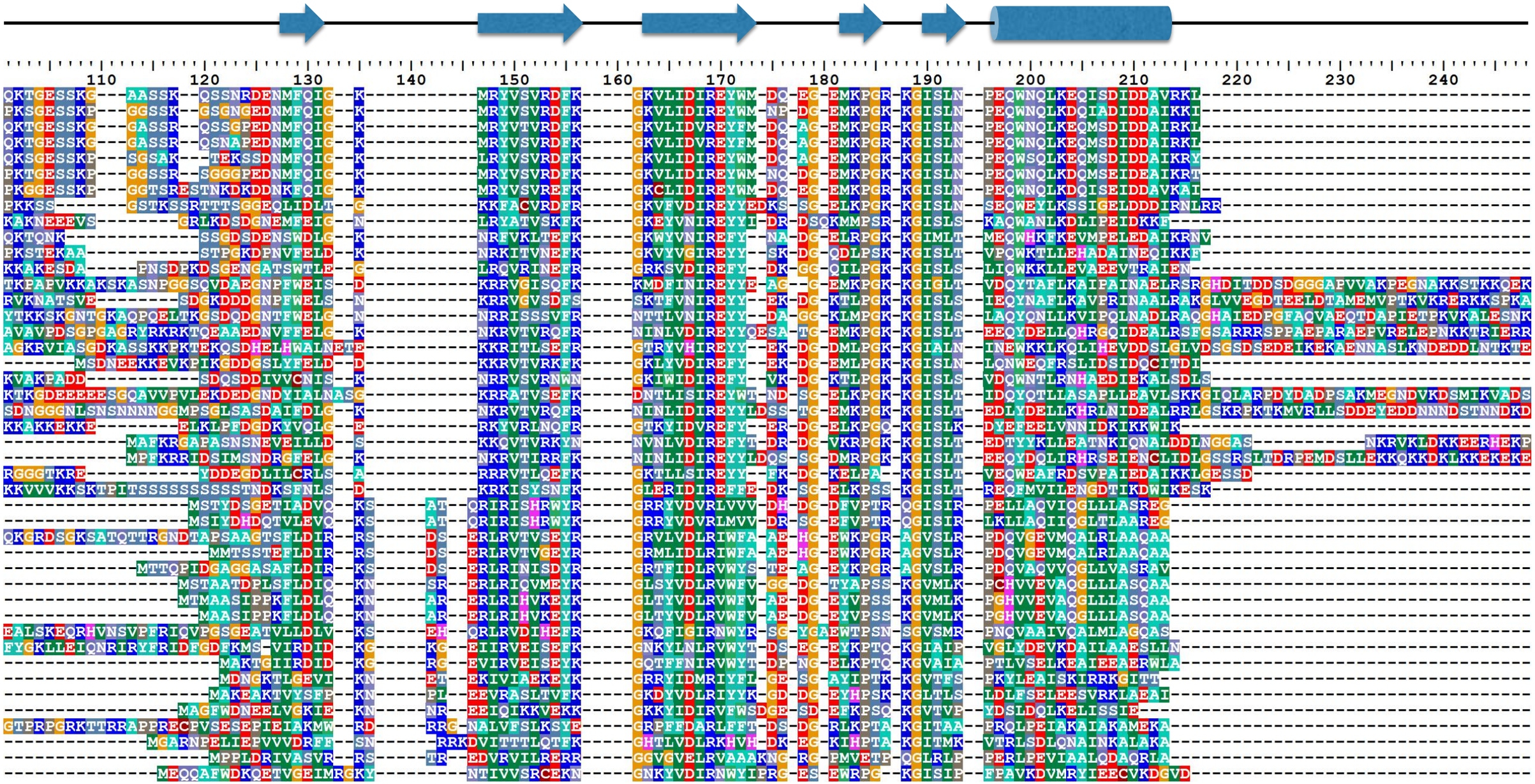
✓ Very helpful in prediction of protein structure

✓ MSA is a prerequisite for phylogenetic analysis

MSA of the ssDNA-binding protein PC4



MSA of the ssDNA-binding protein PC4



ssDNA-binding domain

Scoring a multiple sequence alignment

Scoring function:

- Based on the concept of *sum of pairs* (SP), *i.e.* sum of the scores of all possible pairs of sequences
- Most multiple sequence alignment algorithms try to achieve the <u>maximum SP score</u>

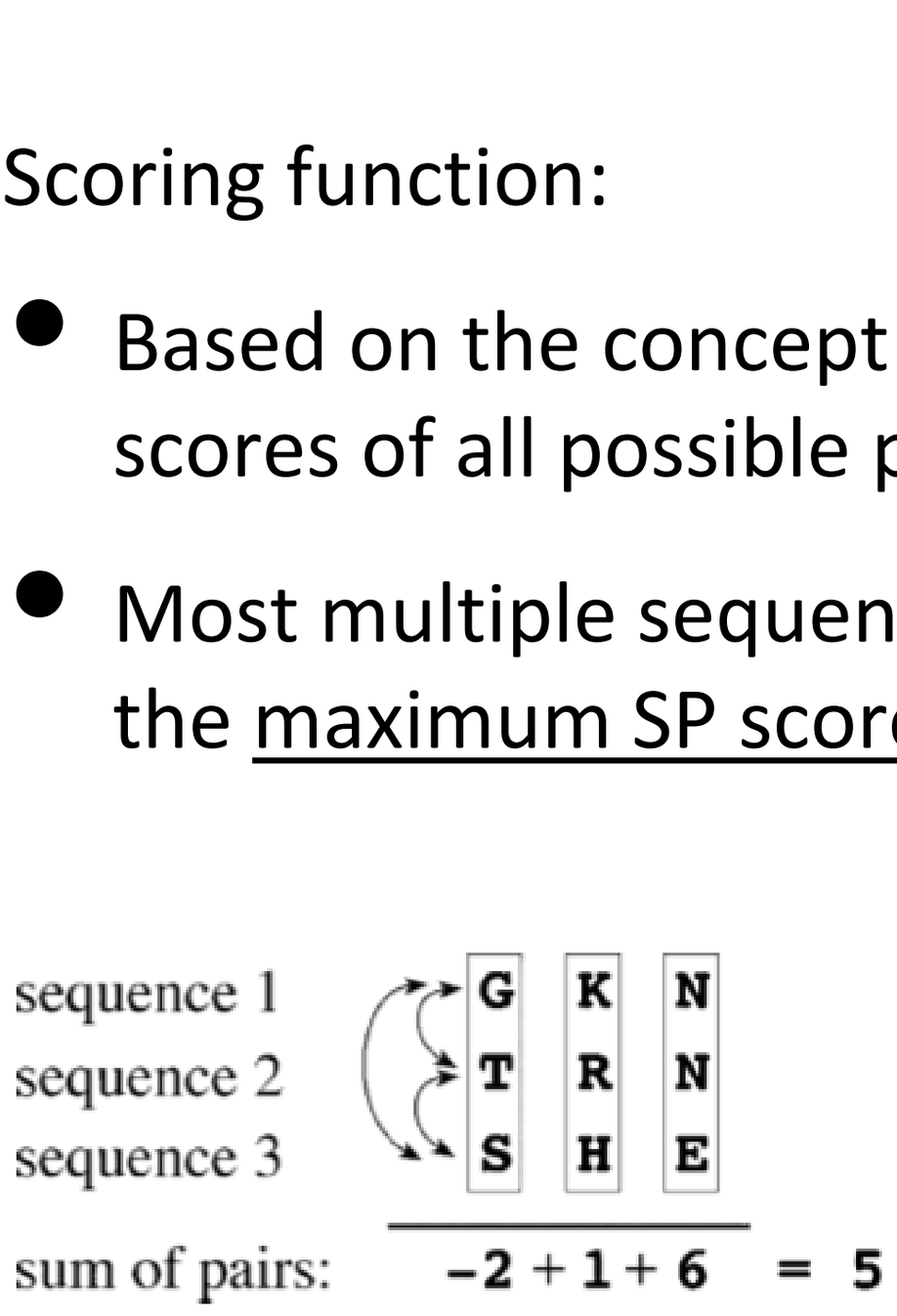


Figure 5.1: Given a multiple alignment of three sequences, the sum of scores is calculated as the sum of the similarity scores of every pair of sequences at each position. The scoring is based on the BLOSUM62 matrix (see Chapter 3). The total score for the alignment is 5, which means that the alignment is $2^5 = 32$ times more likely to occur among homologous sequences than by random chance.

Exhaustive *vs* heuristic approaches

Dynamic programming can **in principle** be adapted to align any number of sequences, but:

- Computing time and memory required increase exponentially as the number of sequences increases
- Dynamic programming becomes impractical when dealing with more than 10 sequences
- Heuristic approaches are almost always preferrable!

Exhaustive algorithms: dynamic programming

- Dynamic programming requires *N*-dimensional search matrix for *N* sequences, *e.g.* a 3-dimensional matrix for 3 sequences
- Back-tracking is applied through the *N*-dimensional matrix to find the highest score path that represents the optimal alignment

Semi-exhaustive alignment: DCA (Divide-and-Conquer Alignment)

- http://bibiserv.techfak.uni-bielefeld.de/dca
- Each of the sequences is broken up into two smaller sections; further divisions are carried out if the sections are not short enough
- Dynamic programming is applied to align each set of "subsequences"
- Breaking points are determined based on regional similarities of the sequences (using heuristic methods)
- Still computationally intensive, only a very limited number of sequences can be handled

Progressive alignment

General principles of this *heuristic* approach:

- Stepwise assembly of a multiple alignment by repeatedly aligning pairs of sequences
- Pairs can be aligned using *e.g.* the Needleman-Wunsch global alignment method
- Further sequences are added by aligning them to one of the previously aligned sequences, or to a consensus sequence

Consensus sequence

	Amino acid position				
.....	*n*	*n*+1	*n*+2	n+3	.....
	A	W	Q	R	
	A	W	N	K	
	G	Y	Q	R	
	A	W	Q	R	
	A	F	-	R	
	S	W	-	K	
	A	F	Q	R	
Consensus sequence:					
	A	**W**	**Q**	**R**	

Which sequences should be aligned first?

- Highly similar sequences can be aligned very accurately; aligning more divergent sequences is less obvious!
- Therefore, the best strategy would be to align pairs of similar sequences first, before adding the more divergent ones to the (increasingly informative!) consensus sequence

Progressive alignment using a guide tree

This is the typical approach used by almost all MSA programs, *e.g.* **CLUSTAL**:

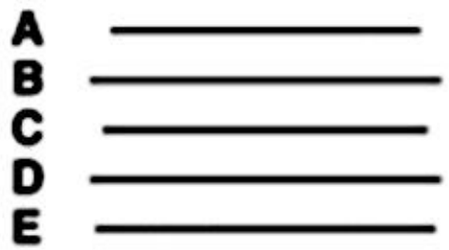
- Initially: pairwise alignments are produced for each possible pair, *e.g.* using Needleman-Wunsch
- Pairwise scores (either percent identity or similarity scores) are converted to an evolutionary distance matrix
- Based on the distance matrix, a simple phylogenetic tree (a so-called *guide tree*) is generated using *e.g.* the neighbour-joining method (chapter 11)

Progressive alignment using a guide tree

- The two most closely related sequences are converted to a consensus sequence with gap positions fixed
- Subsequently, this consensus is treated as a single sequence
- Next most related sequence pair according to the tree is aligned, a consensus sequence created, and so on
- Process is repeated until all the sequences are aligned

Progressive alignment using a guide tree

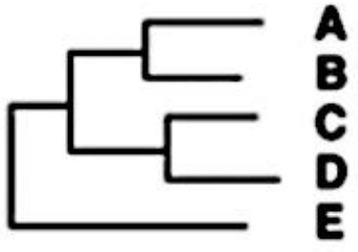
1. making a guide tree:



all individual pairwise alignment and construction of distance matrix

	A	B	C	D	E
A	–				
B	11	–			
C	20	30	–		
D	27	36	9	–	
E	30	33	20	27	–

calculating a guide tree; C & D the closest pair; A & B the next closest pair



2. aligning the sequences:

aligning C/D and A/B separately using dynamic programming

C/D and A/B alignments reduced to consensus sequences which are aligned to each other

creating a new consensus for C/D/A/B which aligns with E

completing alignment

Adjustments to the alignment parameters

Parameters for each consecutive alignment step may be chosen according to how similar two sequences are, which can be estimated from the guide tree

- Choice of substitution matrix (BLOSUM*N*, PAM*N*)
- Gap penalties may be adjusted (*e.g.* larger penalties in conserved regions)

Dealing with overrepresented sequence families

Some sequence families may be overrepresented within the dataset (*e.g.* near-identical sequences from a group of closely related organisms)

Such sequences will completely dominate consensus sequences and make alignment of distantly related proteins less reliable

Usually, some kind of weighting scheme is applied:

- Redundant / very closely related sequences are "down-weighted" in determining the consensus
- Weight factor for each sequence is determined by its branch length on the guide tree

Problems in early implementations of progressive alignment

- Early programs were not suitable for sequences of different lengths, as they relied on global alignment methods (*e.g.* Needleman-Wunsch)
- Final alignment result was often strongly influenced by the order of sequence addition and the quality of guide trees, which would not be called into question anymore during the multiple alignment stage ("greedy algorithm")
- *E.g.* gaps introduced in early steps of alignment would be fixed and never corrected ("once an error, always an error")
- Therefore, final alignment sometimes far from optimal (especially for highly divergent sequences)

Countless improvements have been proposed…

- T-Coffee
- DbClustal
- Poa
- PRALINE
- PRRN
- DALIGN2
- Match-Box
- Etc…

Block-based alignment methods

- Global alignment methods often fail to recognise conserved domains and motifs in highly divergent sequences of varying lengths
- Block-based local alignment strategies first identify conserved blocks shared by all the sequences (*e.g.* using hashing methods similar to FASTA) and then produce an MSA for these

Iterative approaches

- Repeated modification of an existing (suboptimal) solution to find the optimal solution
- An iterative method starts from a low-quality alignment and gradually improves it until no further gains in the alignment score can be achieved
- Of course, as with all heuristic methods: no guarantee whatsoever that the best possible alignment has really been found!

Iterative alignment in PRRN

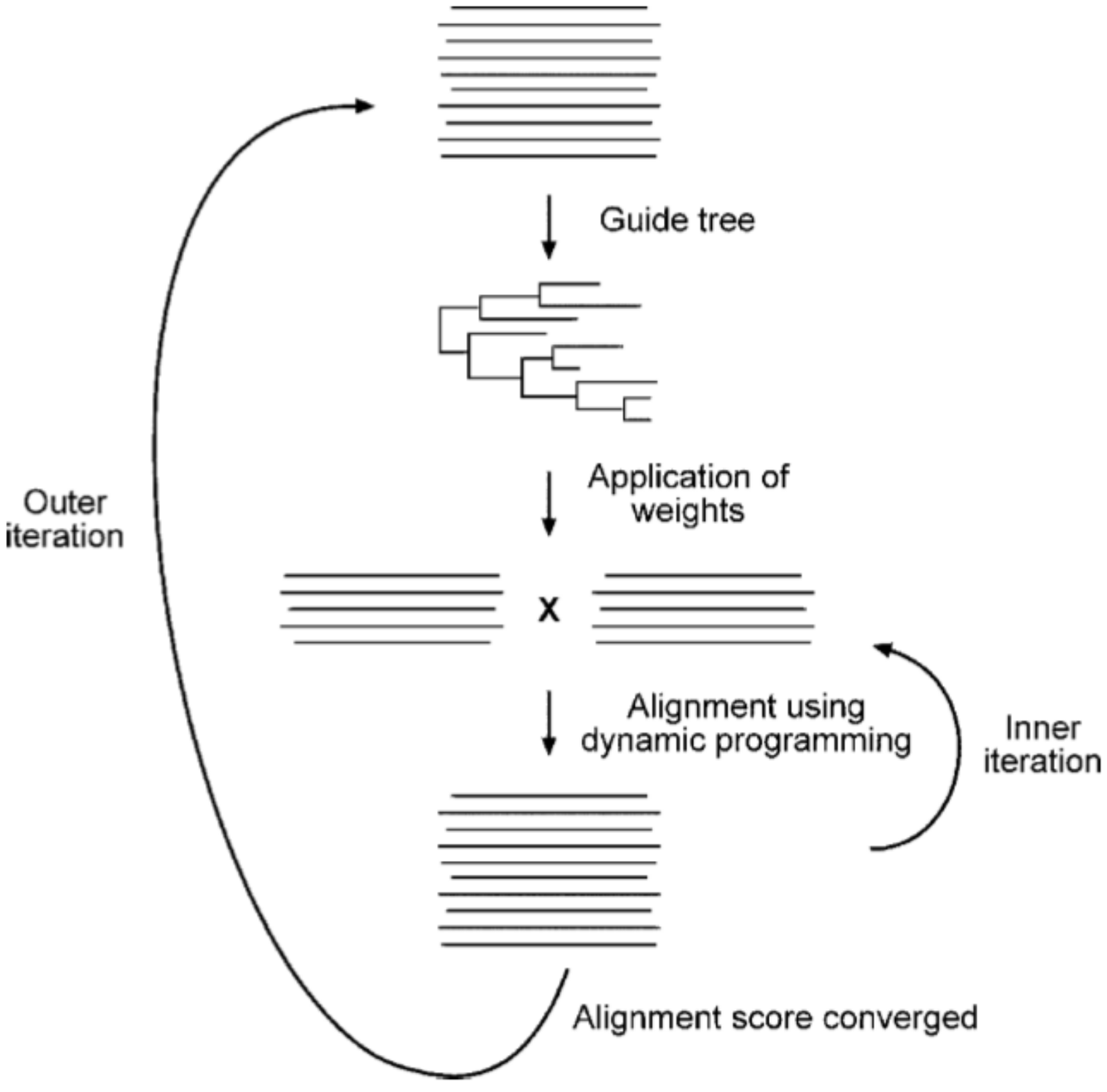


Figure 5.4: Schematic of iterative alignment procedure for PRRN, which involves two sets of iterations.

https://www.genome.jp/tools-bin/prrn

"Outer" iteration:

- Derivation of an UPGMA tree (*cf.* Chapter 11)
- Optimization of alignment with weights

"Inner" iteration:

- Alignment of two randomly chosen half-sets of sequences (both treated as one sequence) by dynamic programming

Using profiles and Hidden Markov Models (HMMs)

- Rather than a single consensus sequence, either a *profile* or a *Hidden Markov Model (HMM)* is constructed
- This provides information about the likelihood of finding certain residues at a given position in the sequence
- Sequences to be added are aligned to the profile or HMM, rather than to a single sequence

	Amino acid position				
.....	*n*	*n*+1	*n*+2	n+3	.....
	Ala 0.5	Trp 0.8	Gln 0.4	Arg 0.8	
	Gly 0.3	Tyr 0.1	Asn 0.3	Lys 0.2	
	Thr 0.1	Phe 0.1	Glu 0.1		
	Ser 0.1		Asp 0.1		
			Ser 0.1		

So many approaches, which one(s) to choose?

CLUSTAL-Ω (the latest iteration of CLUSTAL, arguably the most widely used MSA software):

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

Other MSA servers currently accessible at the EBI web site (https://www.ebi.ac.uk/jdispatcher/msa):

Kalign, MAFFT, MUSCLE, T-Coffee, WebPRANK

Protein *vs* DNA sequences

Protein-coding DNA sequences:

- Alignment at protein level is more sensitive than at the DNA level
- Sequence alignment at the DNA level can often result in frameshift errors as alignment gaps are introduced irrespective of codon boundaries resulting in biologically unrealistic alignments
- Although alignment at DNA level is necessary in some situations (PCR primer design, DNA-based molecular phylogenetic trees) protein-based MSAs are generally preferred

Protein *vs* DNA/RNA sequences

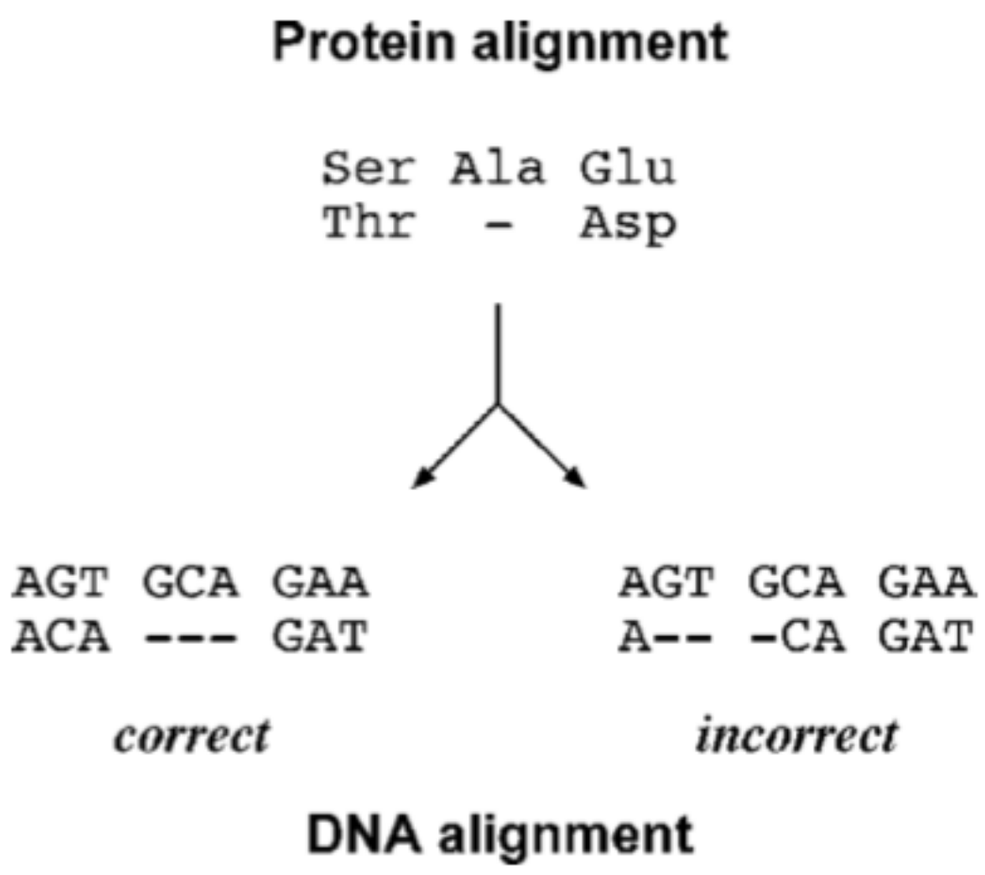


Figure 5.5: Comparison of alignment at the protein level and DNA level. The DNA alignment on the left is the correct one and consistent with amino acid sequence alignment, whereas the DNA alignment on the right, albeit more optimal in matching similar residues, is incorrect because it disregards the codon boundaries.

Protein *vs* DNA/RNA sequences

Protein-Coding DNA Sequences:

- DNA can be translated into an amino acid sequence before carrying out alignment
- After alignment of the protein sequences, the alignment can be converted back to a DNA alignment
- RevTrans https://services.healthtech.dtu.dk/services/RevTrans-2.0

Editing an alignment by hand

- Automated alignment (no matter by which heuristic method) sometimes contains misaligned regions
- Corrections resulting from experimental evidence or mere experience may be needed
- Word processors can be used to edit the alignment, but dedicated software exists

Editing an alignment by hand

E.g. BioEdit:

- https://bioedit.software.informer.com
- Free alignment viewer/editor for Windows
- Can also do BLAST searches, plasmid drawing and restriction mapping

Format conversion

Many different file formats exist to store alignment results. Sometimes conversions are needed if alignments are to be read by other software (*e.g.* for phylogenetic analysis).

E.g. Seqret at EBI:

- https://www.ebi.ac.uk/jdispatcher/sfc/emboss_seqret
- Web-based program that is able to do conversions to/from any known sequence and alignment format