Bioinformatics I - HS 2023

Lecture 1 (19.09.2023)

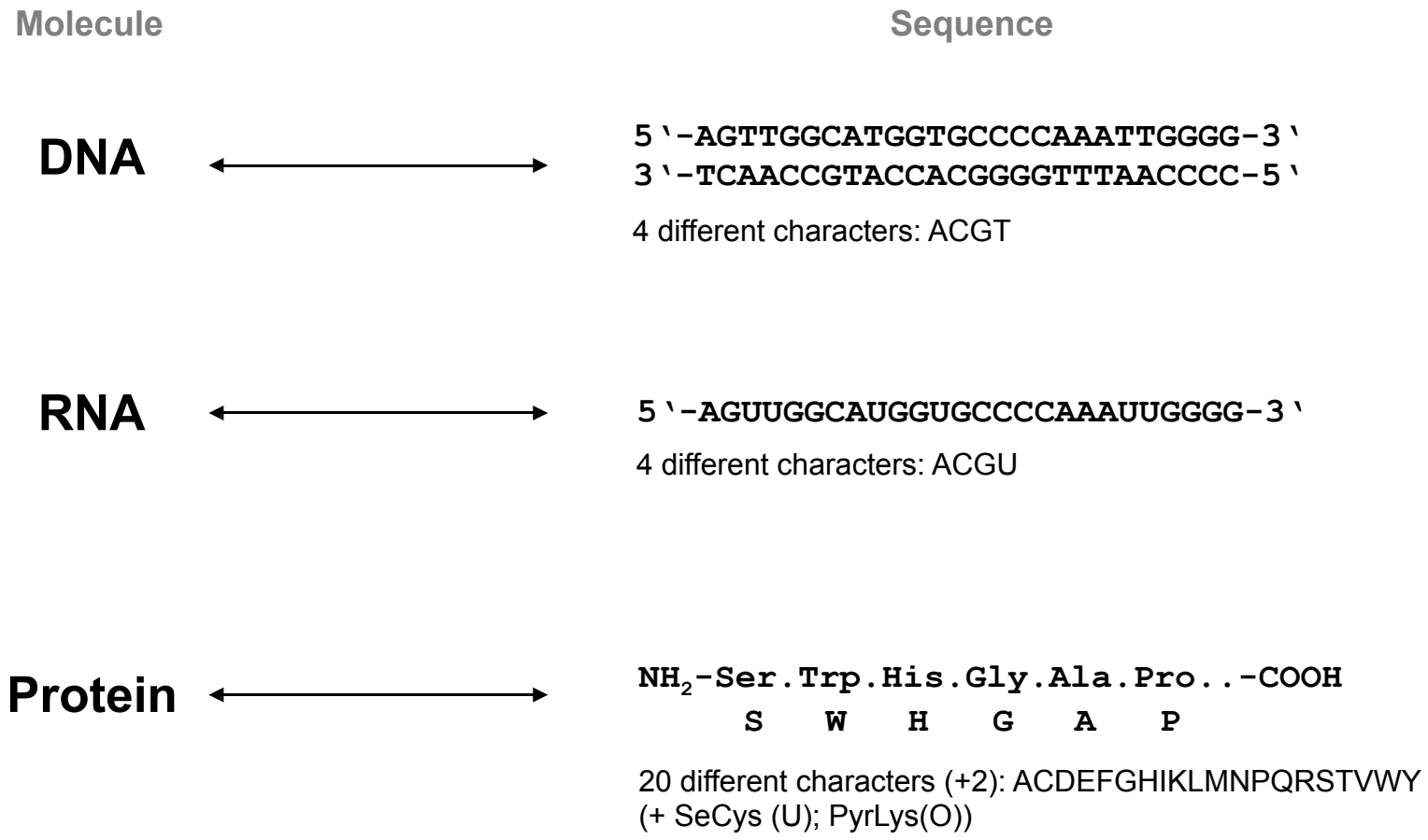
Biological Sequence Informatics

Christian von Mering

mering@mls.uzh.ch

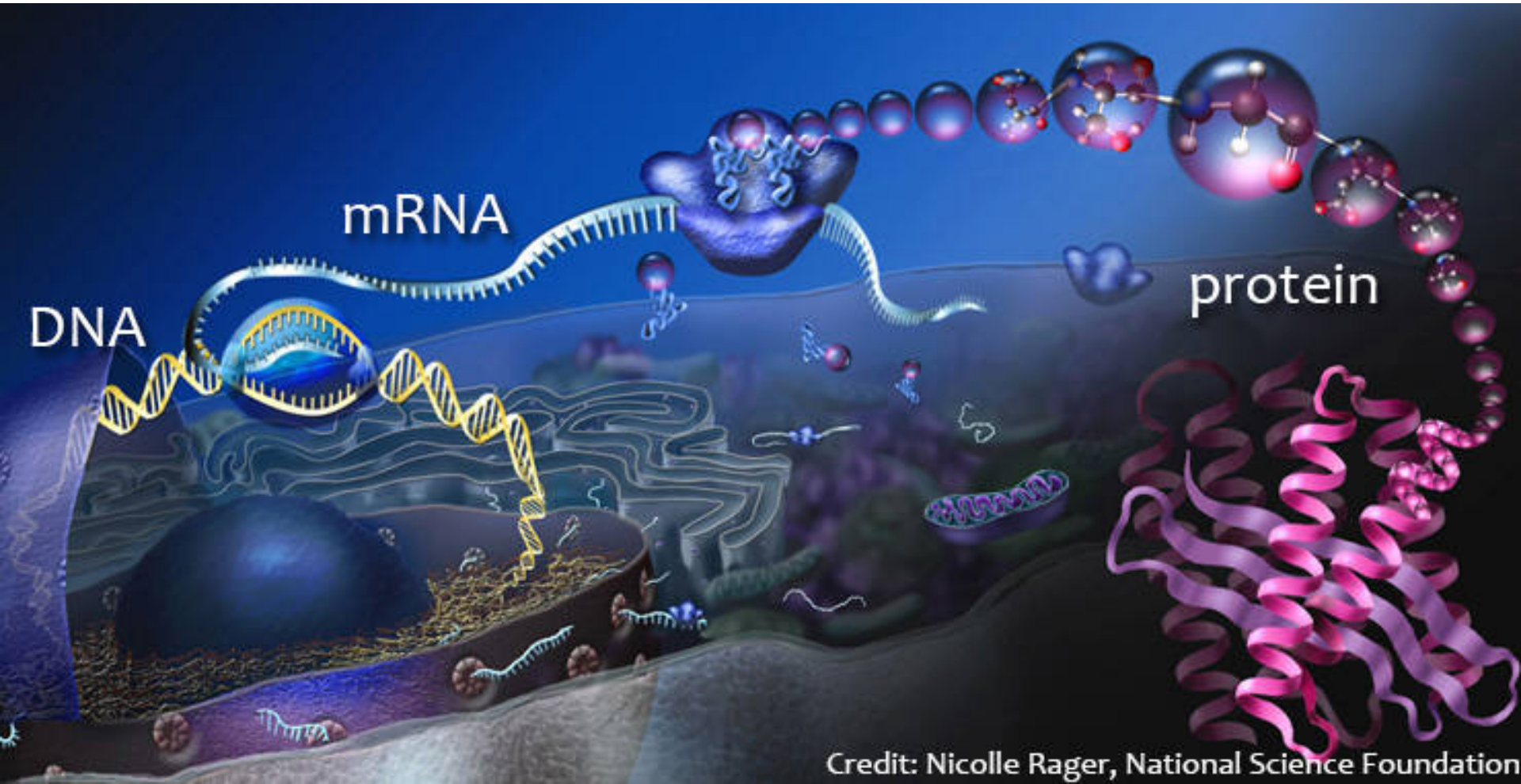
the lecture slides are partially based on an earlier Bioinfo1 lecture by Johannes Fütterer, ETHZ

Sequences in Molecular Biology

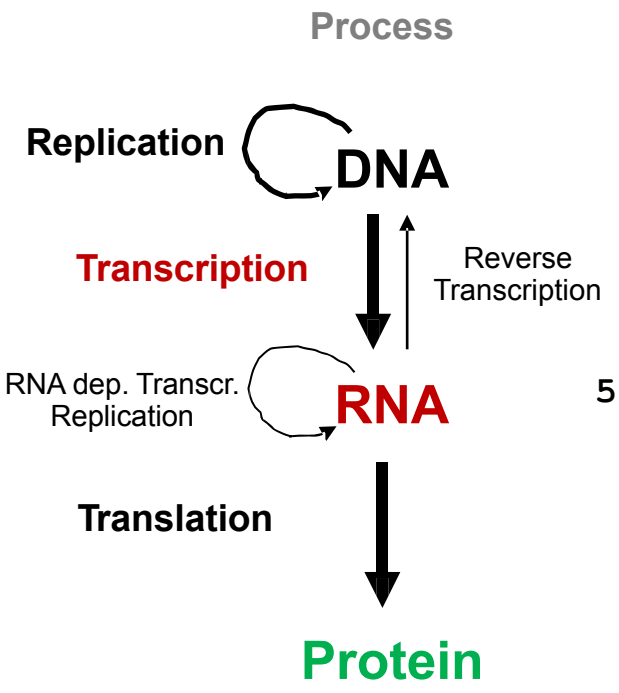


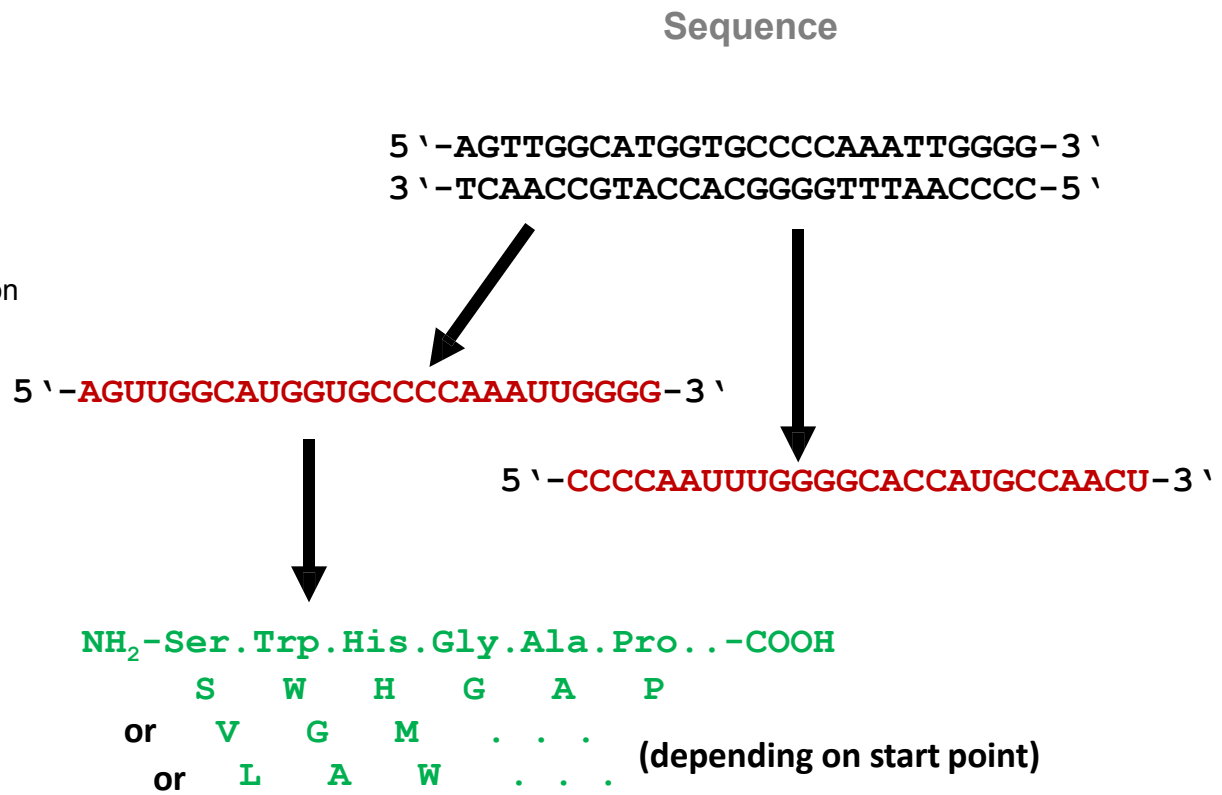
Sequences in Molecular Biology

"Central Dogma": **DNA** ⟶ **RNA** ⟶ **Protein**



Biological processes interconvert sequences



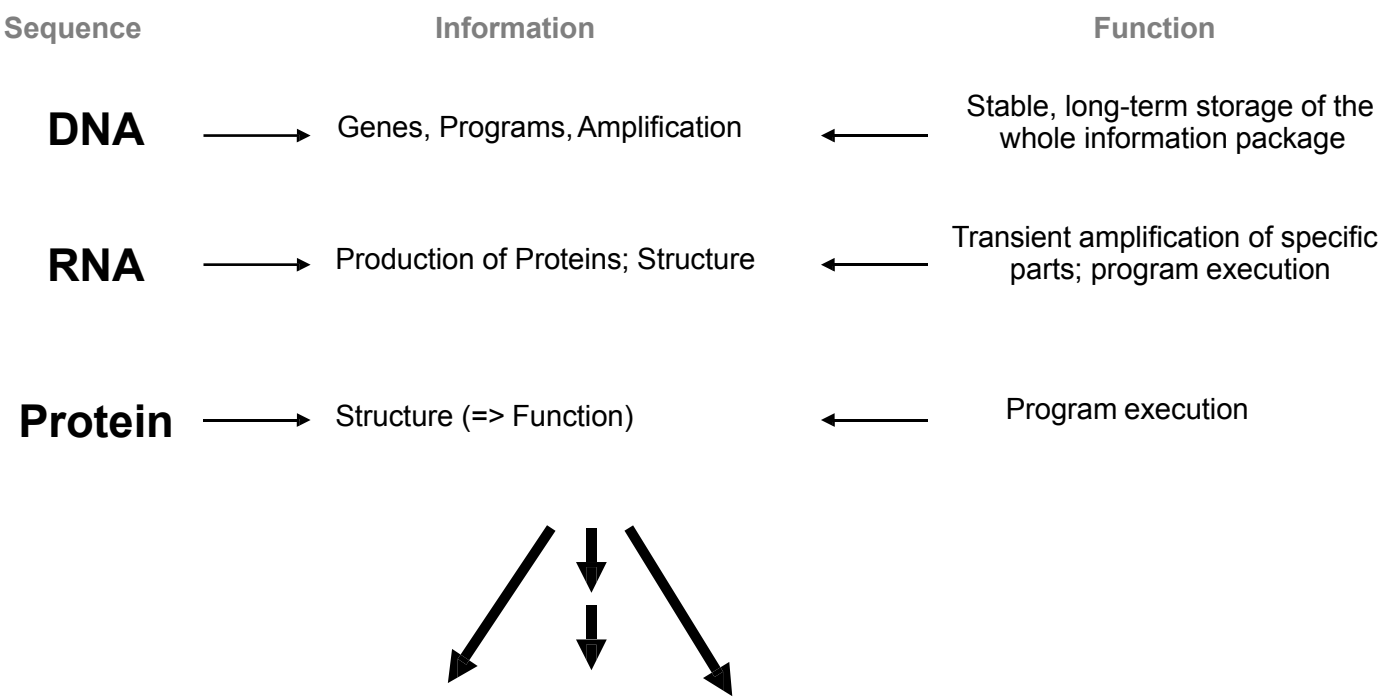


Genetic Code- Table

1st letter	Second Letter: U	C	A	G	3rd letter
U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys	U
	UUC Phe	UCC Ser	UAC Tyr	UGC Cys	C
	UUA Leu	UCA Ser	UAA Stop	UGA Stop	A
	UUG Leu	UCG Ser	UAG Stop	UGG Trp	G
C	CUU Leu	CCU Pro	CAU His	CGU Arg	U
	CUC Leu	CCC Pro	CAC His	CGC Arg	C
	CUA Leu	CCA Pro	CAA Gln	CGA Arg	A
	CUG Leu	CCG Pro	CAG Gln	CGG Arg	G
A	AUU Ile	ACU Thr	AAU Asn	AGU Ser	U
	AUC Ile	ACC Thr	AAC Asn	AGC Ser	C
	AUA Ile	ACA Thr	AAA Lys	AGA Arg	A
	AUG Met	ACG Thr	AAG Lys	AGG Arg	G
G	GUU Val	GCU Ala	GAU Asp	GGU Gly	U
	GUC Val	GCC Ala	GAC Asp	GGC Gly	C
	GUA Val	GCA Ala	GAA Glu	GGA Gly	A
	GUG Val	GCG Ala	GAG Glu	GGG Gly	G

***Note**: Sequences in biology are produced in a defined direction and are usually also written in the same defined direction*
Order of characters is crucial for sequence-function AGTTG ≠ GTTGA or VPQ ≠ QPV

Sequences contain information



Sequences can be interconverted computationally

Genetic Code- Table

1st letter	Second Letter: U		Second Letter: C		Second Letter: A		Second Letter: G		3rd letter
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		UAA	Stop	UGA	Stop	A
	UUG		UCG		UAG	Stop	UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

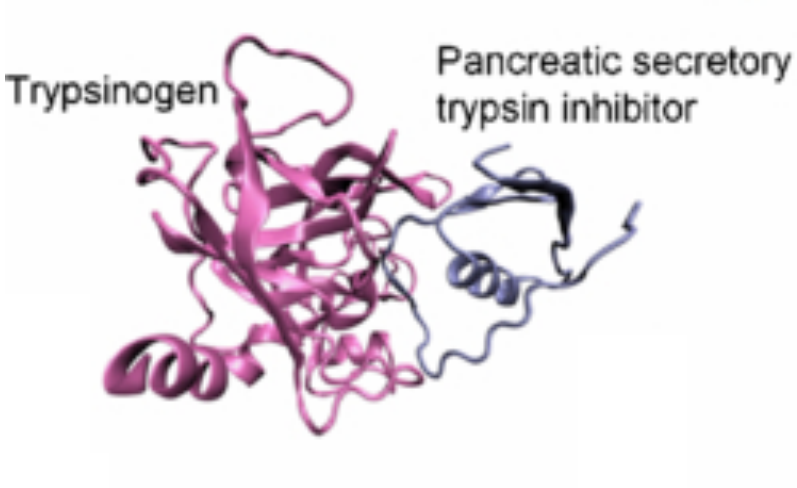
Sequence informatics – what can be learned ?

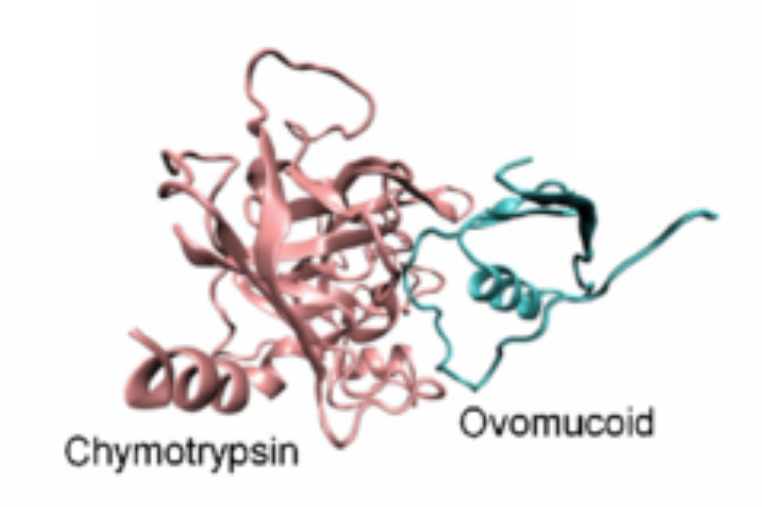
Short DNA pieces: assemble correctly	→	structure of genomes
Sequenced RNA: align with genome	→	location and structure of genes
RNA from special conditions	→	expression control of genes
Sequenced Proteins: align with RNA	→	structure of protein coding regions
proteins from special conditions	→	control of protein production control of function
posttranslational modifications	→	
Comparison of individual sequences	→	deduction of function evolutionary relationships characterization of individual
Comparison of sequence collections	→	ecosystem characterization detection of presence of organism expression profiles

Required: Methods to compare sequence strings quantitatively to determine their similarity

Biological basis for sequence alignment

=> many genes are related by common descent






```
Chymotrypsin   VKKTMVCAGG-DGVISACNGDSGGPLNCQLENGSWEVFGIVSFGSRRGC [...]
               +   M C G  +G   +C GDSGGP+ C   NG  +  G+VS+G   GC
Trypsinogen    ITSNMFCVGFLEGGKDSCQGDSGGPVVC---NGQLQ--GVVSWGD--GC [...]
```

Similarity of biological sequences: some definitions

Similarity: The degree to which two items share certain characters

Homology: descended from a shared common ancestor

Orthology: derived from common ancestor during speciation (often with retained function)

Paralogy: evolved in parallel after gene duplication (often with diverged function)

Note: Sequences are either homologous or not

Analogy: similarity without homology, e.g. due to convergent evolution

Sequence similarity: two sequences contain a number of identical or related characters in corresponding positions

Sequence Similarity

Many possible definitions of “similarity”: length, character content, character distribution,…..

Biological definition: (interrupted) stretches of **identical** or **similar** characters

E.g. search identical sequence segments for assembly of long sequences from short, overlapping fragments

```
AAGCTTACCAAAATTGAAGGGACGTTGACGTAGGGGGACGCTTTAG
                                        GACGCTTTAGTTTAGCCACCGGTATTTAGC
```

Similar characters: physico-chemical characteristics, functional characteristics, evolutionary relation……

Comparison of two (or more) sequences: **Alignment** of identical and similar sequence segments

```
AAGCTTACCAAAATTGAAGGGACGTTGACGTAGGGGGACGCTTTAG
                    AATCTAGCAATTATTGAAGGGACGTTGACGAAGGGGTTCGCTACCG
```

Challenge: Find the best possible alignment **(and do it fast)**

```
AAGCTTACCAAAATTGAAGGGACGTTGACGTAGGGGGACGCTTTAG
AATCTAGCAATTATTGAAGGGACGTTGACGAAGGGGTTCGCTACCG
```

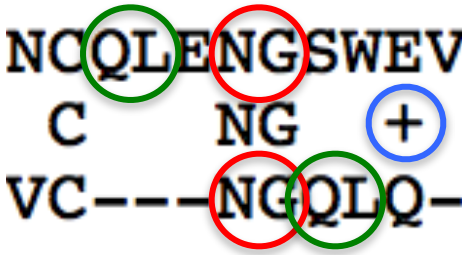
A realistic example

Why is it so difficult !? Isn't this trivial ?

Chymotrypsin VKKTMVCAGG-DGVISACNGDSGGPLNCQLENGSWEVFGIVSFGSRRGC […]

\+ M C G +G +C GDSGGP+ C NG + G+VS+G GC

Trypsinogen ITSNMFCVGFLEGGKDSCQGDSGGPVVC---NGQLQ--GVVSWGD--GC […]



- Why was "NG" aligned and not "QL"?
- What does the "+" mean ?
- How "good" is my alignment?

=> there are <u>many</u> possible ways to align two sequences !!

=> we need a formalized scoring system,
to describe and measure similarities, and to develop statistics …

Scoring systems for calculation of “similarity”

General scoring systems are **context independent** (i.e. values are the same for every occurence of a given pair). They can be written in form of a simple matrix :

e.g. Nucleotide identity matrix (positive score only for identities; no penalty for mismatch)

	A	G	T	C
A	**5**	0	0	0
G	0	**5**	0	0
T	0	0	**5**	0
C	0	0	0	**5**

e.g. Nucleotide substitution matrix (positive scores also for certain subtitutions)

	A	G	T	C
A	**4**	1	0	0
G	1	**4**	0	0
T	0	0	**4**	1
C	0	0	1	**4**

The matrices show scoring values M_{ij} for every conceivable pair i and j.
Matrices are symmetrical

Scoring values for sequence alignments

Quantification according to an **Identity** or **Substitution Matrix**

The matrix assigns a **value to every possible character pair** that could be observed in a comparison

In an alignment of two strings of characters, the scoring values for all occuring characterpairs are combined to produce a score for the respective alignment

Scores are calculated for every possible alignment between two strings, ranked, and compared to random alignments for a statistical evaluation => **Optimal Alignment**

Question: How can we derive realistic scoring values?

Rules for scoring values are based on a specific model for the origin of the expected similarity

The meaning of scoring values - (I) probability

Value in nucleotide identity matrix **= probability** that characters in a pair are **identical**

	A	G	T	C
A	**1**	0	0	0
G	0	**1**	0	0
T	0	0	**1**	0
C	0	0	0	**1**

Alignment score = probability that the aligned sequences are identical =
Product of matrix values of each position of the alignment

Example:

```
AAGCTTACCAAAATTGAAGGGACGTTGACGTAGGGGGACGCTTTAG
AATCTAGCAATTATTGAAGGGACGTTGACGAAGGGGTTCGCTACCG
```

Score of the **global** alignment = 0

Score of **local** alignment (red substring): = 1

Matrix produces a **yes/no** answer; ok for identity, not useful for similarity

The meaning of scoring values - (il) arbitrary score

Value in nucleotide identity matrix **=** **arbitrary score** for a matching pair

	A	**G**	**T**	**C**
A	**1**	0	0	0
G	0	**1**	0	0
T	0	0	**1**	0
C	0	0	0	**1**

Alignment score = sum of matrix values of each position of the alignment

```
AAGCTTACCAAAATTGAAGGGACGTTGACGTAGGGGGACGCTTTAG
AATCTAGCAATTATTGAAGGGACGTTGACGAAGGGGTTCGCTACCG
```

Global aligment = 34 matches of 46 pairs => Score 34
Red substring = 18 matches of 18 pairs => Score 18

Without normalization, the score will grow with the length of the alignment

Evolution as the basis of scoring systems

Biological sequences change by (almost) **random mutation**:

substitutions of characters (e.g. CG → TG; Ala-Thr-Gly → Ala-Ser-Gly)

insertions
deletions
} indels (e.g. CTGG-ACAG ⟷ CTGGAACAG)

Most of the currently observed sequence variation has been **fixed during evolution**

Evolutionary fixed ("accepted") sequence variation is restricted by functionality

=> in functional sequences some mutations are deleterous: less likely to become fixed
=> in „non-functional" sequences mutations are less consequential => more mutations fixed

⇒ Even though every nucleotide and amino acid may technically mutate with almost equal frequency, in real life not all mutations in a coding sequence are observed with equal frequency.
⇒ p_{ij} (probability that i to j change is accepted and can be observed) depends on i and j;

p_{ij} may serve as a quantitative measure of similarity between i and j

Rules for quantitative evaluation of changes in comparisons of biological sequences are based on theoretically or empirically derived "models" of evolution

The PAM concept

Working hypothesis: - **the compared sequences are evolutionary related**
- they differ by N% altered characters (mutations)

PAM = Point Accepted Mutation (or “percent accepted mutation”)

PAM1 reflects an evolutionary distance where 1% of characters have been changed:
=> 99% of character pairs of an aligment should be identical; 1% mismatched

⇒**mutation matrix**

	A	G	T	C
A	**0.99**	0.0033	0.0033	0.0033
G	0.0033	**0.99**	0.0033	0.0033
T	0.0033	0.0033	**0.99**	0.0033
C	0.0033	0.0033	0.0033	**0.99**

Transitions (Pu-Pu/Py-Py changes) = Transversions (Pu-Py changes)

	A	G	T	C
A	**0.99**	**0.006**	0.002	0.002
G	**0.006**	**0.99**	0.002	0.002
T	0.002	0.002	**0.99**	**0.006**
C	0.002	0.002	**0.006**	**0.99**

Transitions = 3 x Transversions

Alignment score = probability that the sequences confirm the working hypothesis, i.e. they are very closely related (1% difference) = product of matrix values for every pair in the alignment

Matrix for greater divergence:
PAM2 = 2% mismatched; PAM4 = 4% mismatched; etc.

The log-odds concept

Conversion of mutation matrix to log-odds matrix: score s_{ij} of a match between nucleotides i and j:

$$s_{ij} = \log (p_i M_{ij} / p_i p_j) = \log (M_{ij} / p_j)$$ (=*log(observed frequency/expected frequency)*)

$P_{i \text{ or } j}$: frequency of nucleotide i or j (= 0.25); M_{ij}: value from the mutation matrix
log base is only a scaling factor; frequently $\log_2$, values for matrix rounded to next integer

	A	G	T	C
A	**2**	**-5**	**-7**	**-7**
G	**-2**	**2**	**-7**	**-7**
T	**-7**	**-7**	**2**	**-2**
C	**-7**	**-7**	**-2**	**2**

log-odds matrix: total alignment score is determined by addition of individual s_{ij} values (instead of multiplication of individual probability values)

Note: This is a convenient example for the PAM/Log-odds concept. Actual values for DNA comparisons are usually derived differently.

a quick aside: Position Specific Scoring Matrices

Scoring Values Depend on Position in a Sequence

(a) Annotated TrpR binding sites

Site ID																			Target Operon
ECK120012644	G	T	A	C	T	A	G	T	T	T	G	A	T	G	G	T	A	T	G	aroL-yaiA-aroM
ECK120012187	G	T	A	C	T	A	G	T	T	T	G	A	T	G	G	T	A	T	G	aroL-yaiA-aroM
ECK120012179	G	A	A	C	T	A	G	T	T	A	A	C	T	A	G	T	A	C	G	trpLEDCBA
ECK120012892	G	A	A	C	T	A	G	T	T	A	A	C	T	A	G	T	A	C	G	trpLEDCBA
ECK120012181	G	A	A	C	T	A	G	T	T	A	A	C	T	A	G	T	A	C	G	trpLEDCBA
ECK120012636	G	T	A	C	T	A	G	A	G	A	A	C	T	A	G	T	G	C	A	aroH
ECK120012183	G	T	A	C	T	A	G	A	G	A	A	C	T	A	G	T	G	C	A	aroH
ECK120012185	G	T	A	C	T	C	G	T	G	T	A	C	T	G	G	T	A	C	A	mtr
ECK120012979	G	T	A	C	T	C	G	T	G	T	A	C	T	G	G	T	A	C	A	mtr
ECK120012894	G	T	A	C	T	C	T	T	T	A	G	C	G	A	G	T	A	C	A	trpR

(b) Position-specific scoring matrix

A	0	3	10	0	0	7	0	2	0	6	7	2	0	6	0	0	8	0	5
T	0	7	0	0	10	0	1	8	6	4	0	0	9	0	0	10	0	2	0
C	0	0	0	10	0	3	0	0	0	0	0	8	0	0	0	0	0	8	0
G	10	0	0	0	0	0	9	0	4	0	3	0	1	4	10	0	2	0	5

(c) Consensus

G	w	A	C	T	m	G	t	k	w	r	C	t	r	G	T	r	C	r

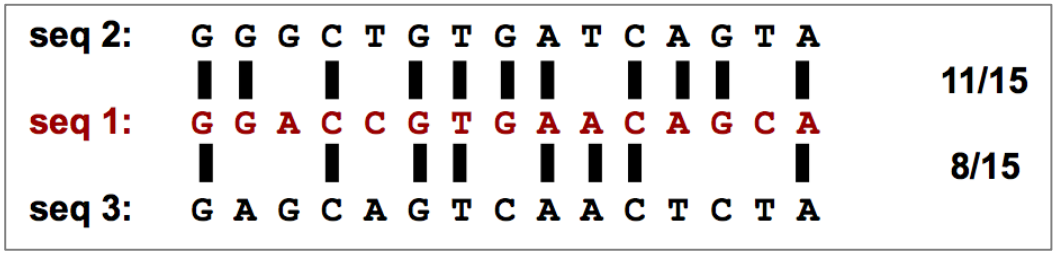
(d) Sequence logo

Useful for finding specific functional sequence motives (TF binding sites, protein domains)

DNA comparison

simple scoring systems treat all matches as equal and all mismatches as equal a negative score is assigned to mismatches

Alignments without gaps (only substitutions allowed):



seq 1 is closer related to seq 2 than to seq 3

But natural sequence variation also involves indels !

=> Gaps in sequence alignments must be possible

DNA comparison with gaps

Alignments with gaps (indels allowed; increases possibilities drastically):

```
seq 1:  G G A C C - G T G A A C A G C - A
          ▌ ▌   ▌   ▌ ▌   ▌ ▌ ▌     ▌   ▌   10/15 (18)
seq 3:  - G A G C A G T C A A C T - C T A
```

But also:

```
seq 1:  G G A C C G T G A A C A G - C A
          ▌ ▌     ▌       ▌ ▌ ▌   ▌ ▌
seq 3:  - G A - - G - - - - C A G T C A A C T C T A
```

values for introduction of gaps are estimated **empirically**:

Affine gap: high value (A) for gap opening + lower value (B) for every gap elongation step

Gap penalty P for a gap of length n: P = A + nB

“Real life” scoring parameters for DNA alignments

Parameters	Smith-Waterman	FASTA	wuBLAST	ncbiBLAST
Match	5	5	5	1
Mismatch	-4	-4	-4	-3
Gap opening	-16	-16	-10	-5
Gap extension	-4	-4	-10	-2

	S-W/F	wuBLAST	ncbiBLAST
AGATCAACGGATTGCTTTCCTGCCGCCATT AGATCTTCGGATT---TTCCTGGGGCCATT	75	69	2

=> scores can be only compared within one scoring system

Information content of biological sequences

Information content of a character in a string depends on the number of possible characters that could be found at that position.

Information content of (sub-)sequences containing only few or a subset of the possible characters is lower
e.g. AATAATTAAAATAAATAA for DNA (longer stretch of only 2 of the 4 possible characters) or

LLDELDDELLDEL for protein (longer stretch of only 3 of the 20 possible characters)

In sequence comparison programs such «Sequences of low complexity" are sometimes filtered and marked as xxxxxxxx or in lower case.

The occurence of such sequences can be biologically relevant, but the sequences are not included in the calculation of similarity values, because character pairing would result in high scores even though it may occur by chance (i.e. any alignment of two such regions will produce a relative high score by chance and not because the aligned positions are homologous).

DNA/DNA vs. protein/protein comparisons

Information content per position in DNA is low

only 4 possibilities => similarity by chance

DNA may contain many positions with no or small functional importance
e.g. non-coding regions, third codon positions such

positions may not be conserved in evolution

=> relatedness may be overlooked

DNA-DNA comparison is only useful for closely related sequences or highly conserved motives

DNA/DNA vs. protein/protein comparisons

Seq 1: CCTGGAGTCCAGCAAAAACGTC

Seq 2: CATGGTGACCACCGAAAAGCTC 15/22

Seq 3: GTTAGAAAGTTCTAAGAATGTG 9/22

Seq 2 seems to be much closer related to Seq 1

But: Sequences have coding potential for peptide sequences:

Seq 1: C.CTG.GAG.TCC.AGC.AAA.AAC.GTC

.leu.glu.ser.ser.lys.asn.val

Seq 2: C.ATG.GTG.ACC.ACC.GAA.AAG.CTC 15/22

.met.val.thr.thr.glu.lys.leu 0/7

Seq 3: G.TTA.GAA.AGT.TCT.AAG.AAT.GTG 9/22

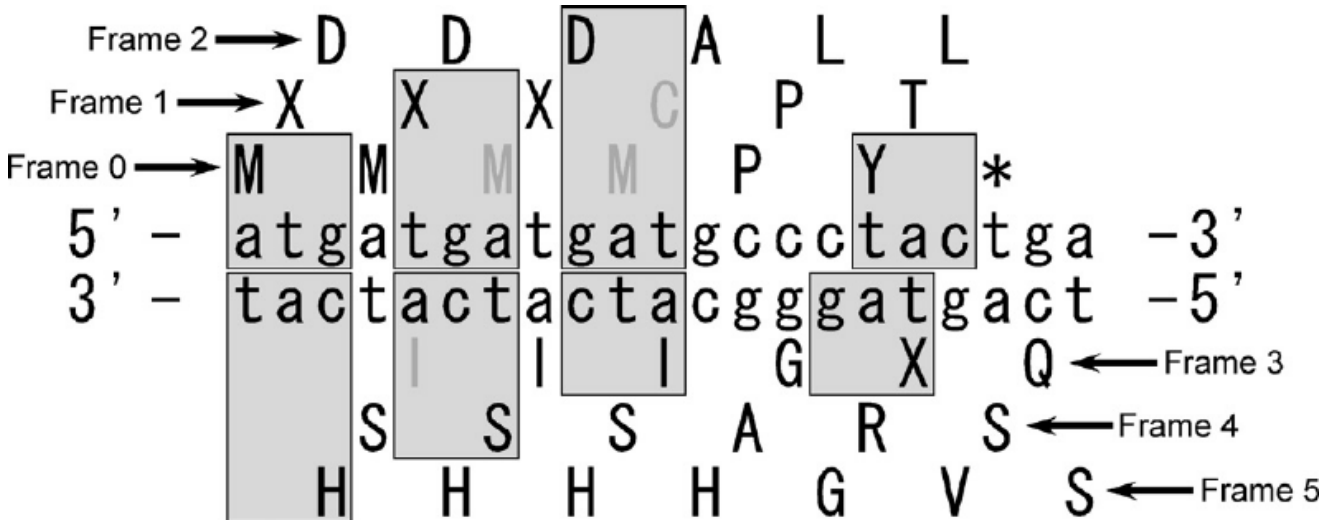
.leu.glu.ser.ser.lys.asn.val 7/7

The peptide encoded by Seq 3 is identical to that encoded by Seq 1 !

=> always search at the level that carries the biological function

Conceptual translation of DNA into protein

six possible reading phases per DNA sequence:



3 nucleotides form a codon => 64 codons possible

20 different amino acids in proteins

1 to 6 different codons per amino acid
=> different DNA molecules can code for same protein

Genetic Code- Table

1st letter	Second Letter: U	Second Letter: C	Second Letter: A	Second Letter: G	3rd letter
U	UUU Phe, UUC Phe, UUA Leu, UUG Leu	UCU Ser, UCC Ser, UCA Ser, UCG Ser	UAU Tyr, UAC Tyr, UAA Stop, UAG Stop	UGU Cys, UGC Cys, UGA Stop, UGG Trp	U C A G
C	CUU Leu, CUC Leu, CUA Leu, CUG Leu	CCU Pro, CCC Pro, CCA Pro, CCG Pro	CAU His, CAC His, CAA Gln, CAG Gln	CGU Arg, CGC Arg, CGA Arg, CGG Arg	U C A G
A	AUU Ile, AUC Ile, AUA Ile, AUG Met	ACU Thr, ACC Thr, ACA Thr, ACG Thr	AAU Asn, AAC Asn, AAA Lys, AAG Lys	AGU Ser, AGC Ser, AGA Arg, AGG Arg	U C A G
G	GUU Val, GUC Val, GUA Val, GUG Val	GCU Ala, GCC Ala, GCA Ala, GCG Ala	GAU Asp, GAC Asp, GAA Glu, GAG Glu	GGU Gly, GGC Gly, GGA Gly, GGG Gly	U C A G

Amino acid substitution matrices

Model: Amino acids are differentially related to each other

1. required steps for mutation ("distance"):

   e.g. Thr codons (ACN) → Ser (UCN) (one step)

   ↓ → Cys (UGY) (at least two steps)

2. similar physico-chemical characteristics:

   e.g. aromatic side chains: Phe, Tyr, Trp (F, Y, W)
   basic side chains: Lys, Arg, His (K, R, H)

   for different purposes, amino acids may be sorted differently

3. actually observed evolutionary conservation

Many different possibilities for reasonable substitution matrices

Popular amino acid substitution matrices

based on comparisons of evolutionary related proteins

PAM concept was developed for amino acid substitutions

Computed by Dayhoff (1978) based on a **model of protein evolution**

Model: Protein evolution through point mutations:

1) **independent from previous substitutions**
2) **independent from the neighbouring amino acid**

reality is **more complex**: e.g. **3-D and functional** constraints

Analysis of **closely related** sequences:

71 protein families
85% identity (allows for manual global alignment, few indels, minimal multiple changes) in total 1572 accepted changes

⇒**Dayhoff matrix form for PAM n :**
(log-odds matrix)

$$D_{ij}^{(n)} = 10\log_{10}\frac{M_{ij}^{(n)}}{f_i}$$

$M_{i,j}$: frequency with which a.a. i mutates to j in a PAM unit

f_i: frequency of a.a. i (a.a.= amino acid)

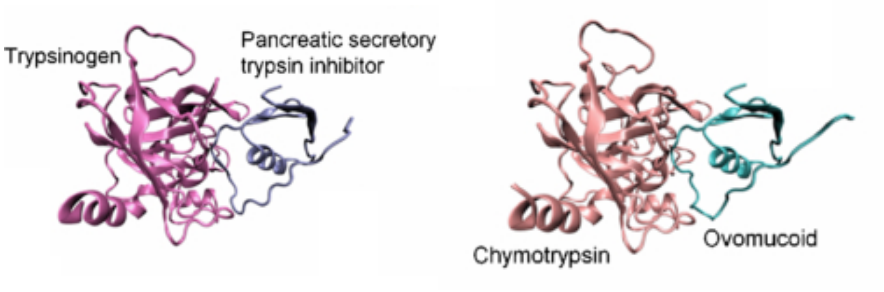
PAM matrices

Alignment:

$$---A_i---$$
$$---A_j---$$

Question: Do A_i and A_j align by chance or because they are evolutionary related ?

(i,j) value in PAM matrix gives the probability ratio of the two possibilities



Important:

in the protein world, spatial alignment of 3D structures can act as “ground truth” of biologically meaningful alignment !

PAM250

```
C  12
S   0   2
T  -2   1   3
P  -3   1   0   6
A  -2   1   1   1   2
G  -3   1   0  -1   1   5
N  -4   1   0  -1   0   0   2
D  -5   0   0  -1   0   1   2   4
E  -5   0   0  -1   0   0   1   3   4
Q  -5  -1  -1   0   0  -1   1   2   2   4
H  -3  -1  -1   0  -1  -2   2   1   1   3   6
R  -4   0  -1   0  -2  -3   0  -1  -1   1   2   6
K  -5   0   0  -1  -1  -2   1   0   0   1   0   3   5
M  -5  -2  -1  -2  -1  -3  -2  -3  -2  -1  -2   0   0   6
I  -2  -1   0  -2  -1  -3  -2  -2  -2  -2  -2  -2  -2   2   5
L  -6  -3  -2  -3  -2  -4  -3  -4  -3  -2  -2  -3  -3   4   2   6
V  -2  -1   0  -1   0  -1  -2  -2  -2  -2  -2  -2  -2   2   4   2   4
F  -4  -3  -3  -5  -4  -5  -4  -6  -5  -5  -2  -4  -5   0   1   2  -1   9
Y   0  -3  -3  -5  -3  -5  -2  -4  -4  -4   0  -4  -4  -2  -1  -1  -2   7  10
W  -8  -2  -5  -6  -6  -7  -4  -7  -7  -5  -3   2  -3  -4  -5  -2  -6   0   0  17
   C   S   T   P   A   G   N   D   E   Q   H   R   K   M   I   L   V   F   Y   W
```

>0: likely mutation
<0: unlikely mutation

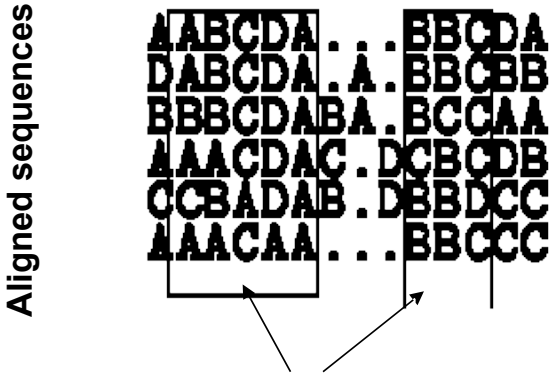
Note: Two sequences differing by 100 PAM units do not differ in all positions:
Effect of multiple substitutions at one site

Residue identity difference (%)	Evolutionary difference (PAM units)
1	1
10	11
20	23
40	56
60	112
80	246

BLOSUM matrices

Blosum: BLOck SUbstitution Matrix

derived from BLOCKS database (Henikoff and Henikoff, 1992)
multiple alignments of distantly related sequences (1764 blocks in 437 protein groups)




```
tlpa_braja AVATAQKIAP LAHGPVAALT MASAPLKLPD LAFEDADGKP KKLSD.....
resa_bacsu SRFNLRTRLY HLQRCICRQR EYIRRSDAPN FVLEDTNGKR IELSD.....
    pestis KIIGLCSLLL LLS..ACKQE KVALGEVAPT LAAYDLQGEA VALEQ.....
helx_rhoca .......... ..QNDPNAMP TALAGKEAPA VRLEPL.... ..GAEAPFTD
cycy_braja .........R LGSGDPSRIP SALIGRPAPQ TALPPLEGLQ ADNVQVPGLD
ccmg_chroma .......... ....DPRKIP SPLVDKPAPE FSLPDLKDPN QT.....LTR
dsbe_ecoli ........RN AEGDDPTNLE SALIGKPVPK FRLESLQNPG QF.....YQA
```

Conserved Blocks

First column in conserved block: AABACA: 6 AA pairs, 4 AB pairs, 4 AC pairs, 1 BC pair
=> Observed probability of an AA pair: 6/15, AB pair 4/15 etc.

These values can be converted to a log-odds matrix

$$s_{i,j} = \log_2 \frac{q_{ij}}{e_{ij}}$$

q_{ij}: normalized observed pair frequency between i and j
e_{ij}: normalized expected frequency of pair ij (reflects the frequency of occurence

Various BLOSUM matrices

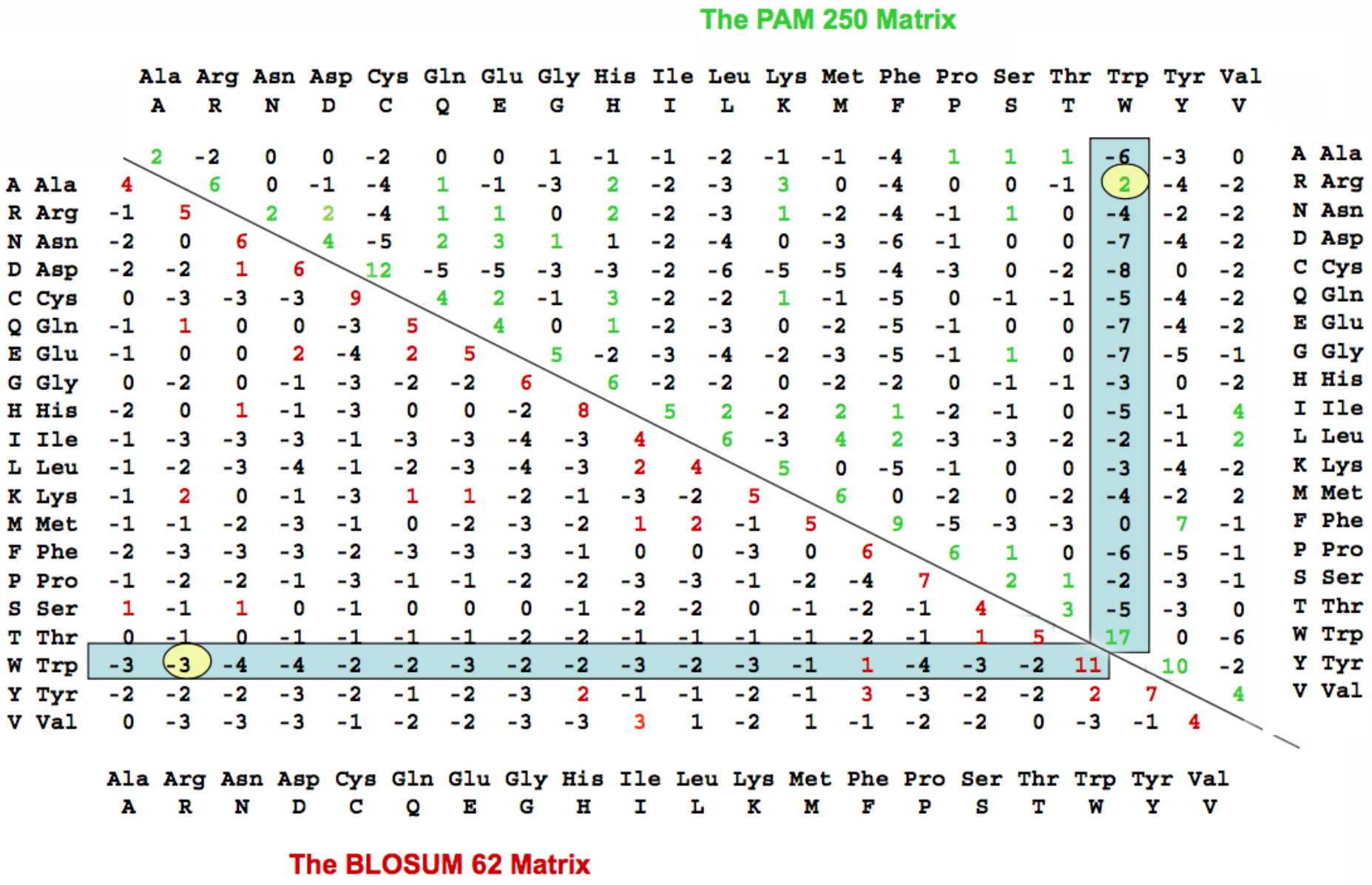
BLOSUM 62: All sequences in the BLOCKS with >62% similarity were collapsed in one.
Removes bias due to contribution of too many closely related sequences in the sequence collection.

A	Ala	4																		
R	Arg	-1	5																	
N	Asn	-2	0	6																
D	Asp	-2	-2	1	6															
C	Cys	0	-3	-3	-3	9														
Q	Gln	-1	1	0	0	-3	5													
E	Glu	-1	0	0	2	-4	2	5												
G	Gly	0	-2	0	-1	-3	-2	-2	6											
H	His	-2	0	1	-1	-3	0	0	-2	8										
I	Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4									
L	Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4								
K	Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5							
M	Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5						
F	Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6					
P	Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7				
S	Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4			
T	Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5		
W	Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	
Y	Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

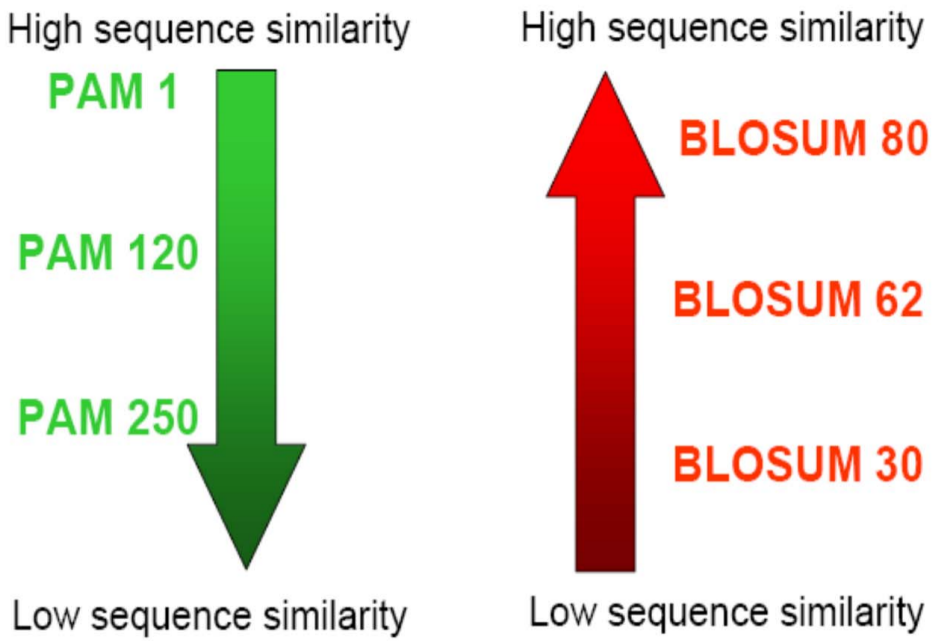
Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

The BLOSUM 62 Matrix

PAM vs. BLOSUM matrices



PAM vs. BLOSUM matrices



All matrices work
Comparison of closely related sequences: low number PAM or high number BLOSUM matrices better
Comparison of less related sequences: high number PAM or low number BLOSUM matrices
Routine: BLOSUM 62; if in doubt, try several different matrices

Protein substitution matrix - summary

For commonly used programs:
Matrix derived from frequency of **actually occuring amino acid pairs** in corresponding positions of **conserved protein** regions; values indicate **probability** that a given amino acid pair reflects evolutionary relationship and does not occur by chance. Different matrices reflect different evolutionary relationships.

Cys-Cys (C-C) pair: high value because C is rare and match by accident therefore unlikely

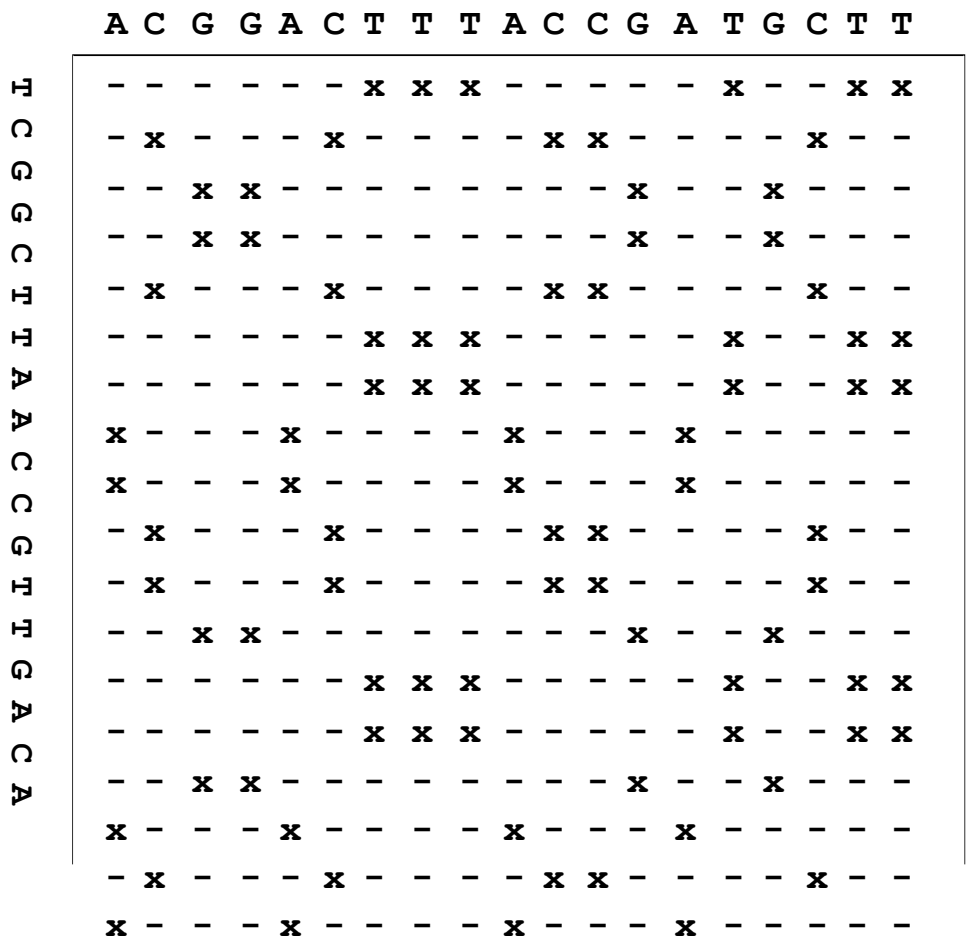
Leu-Gln (L-Q) pair: negative value; not a likely substitution; Amino acids are functionally different

Tyr-Phe (Y-F) pair: high value; frequently observed substitution Amino acids are functionally similar

A	Ala	4																		
R	Arg	-1	5																	
N	Asn	-2	0	6																
D	Asp	-2	-2	1	6															
C	Cys	0	-3	-3	-3	9														
Q	Gln	-1	1	0	0	-3	5													
E	Glu	-1	0	0	2	-4	2	5												
G	Gly	0	-2	0	-1	-3	-2	-2	6											
H	His	-2	0	1	-1	-3	0	0	-2	8										
I	Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4									
L	Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4								
K	Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5							
M	Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5						
F	Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6					
P	Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7				
S	Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4			
T	Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5		
W	Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	
Y	Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7
V	Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

Sequence alignment: algorithms

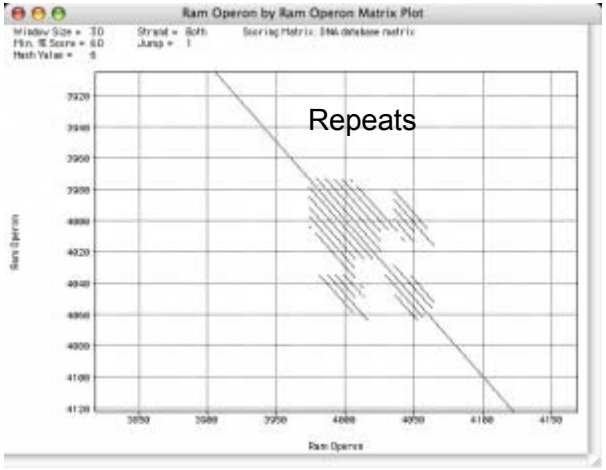


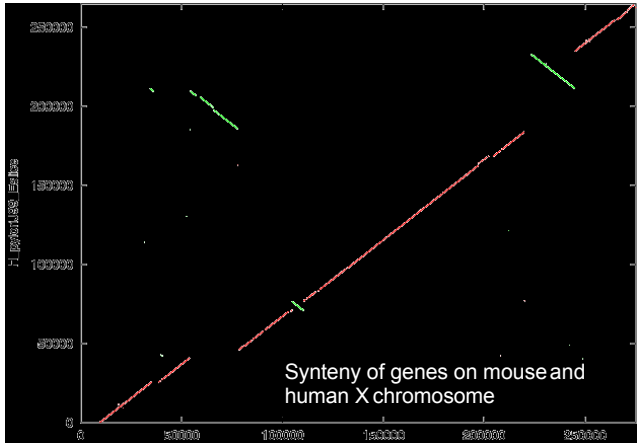
Alignment matrix:
Positions with locally positive alignment score are marked x

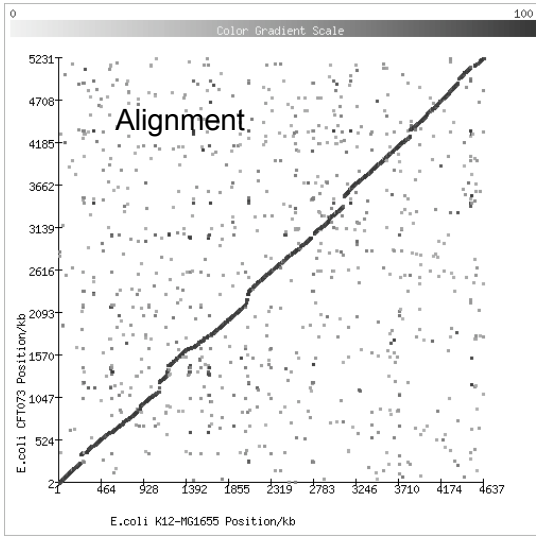
"Positions":

characters
words
genes
……

Dot Plots







DotPlot

=>Rough idea, where the best alignment could be (marked diagonal regions)
Detection of repeats within a sequence

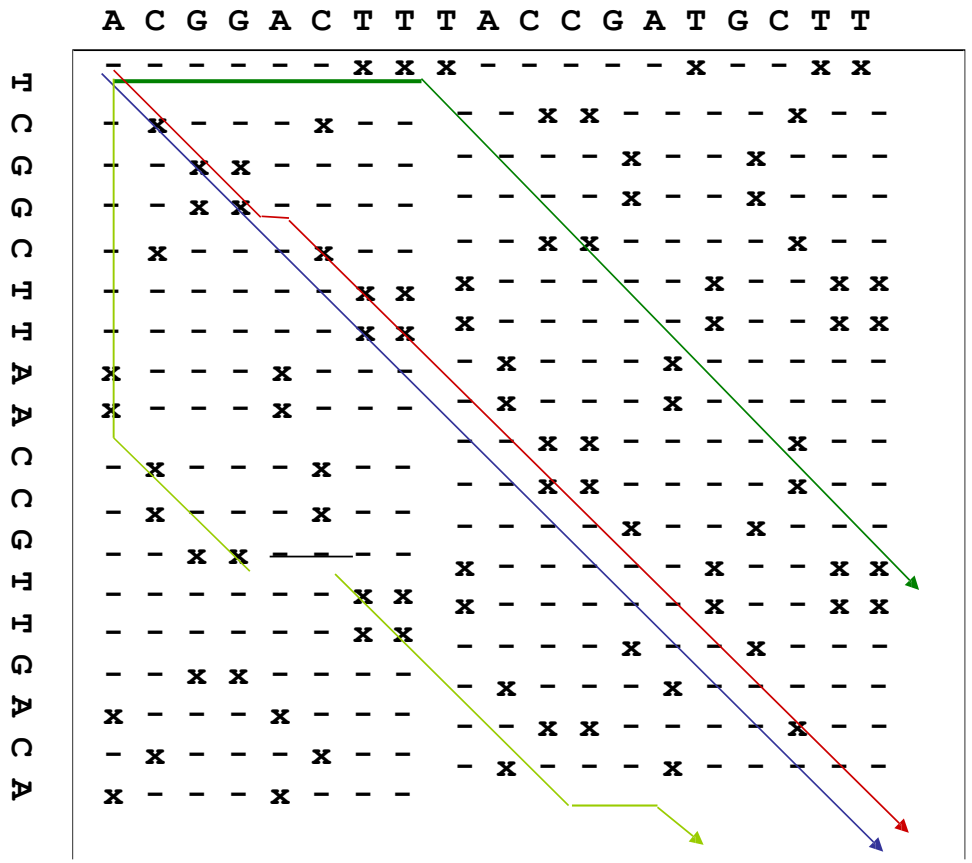
Identities can be characters, strings, genes…….

Performed by programs like “Dotter” http://sonnhammer.sbc.su.se/Dotter.html

Problem:
Visual, no calculation of the best alignment

Searching for the optimal alignment

Exploration of all paths through the alignment-matrix to identify the maximum value for the scoring function => optimal alignment guaranteed (global or local)

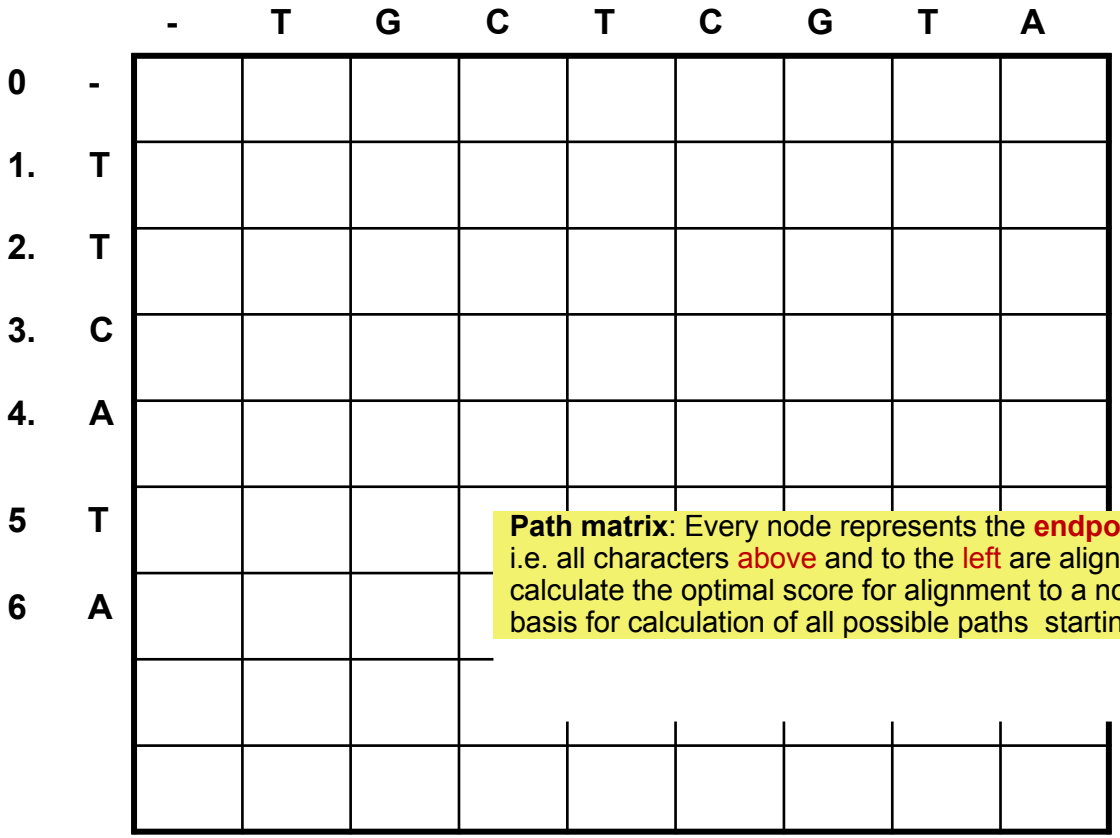


Many possible paths

dynamic programming:
an algorithmic technique in which an optimization problem is solved by caching subproblem solutions instead of recalculating them

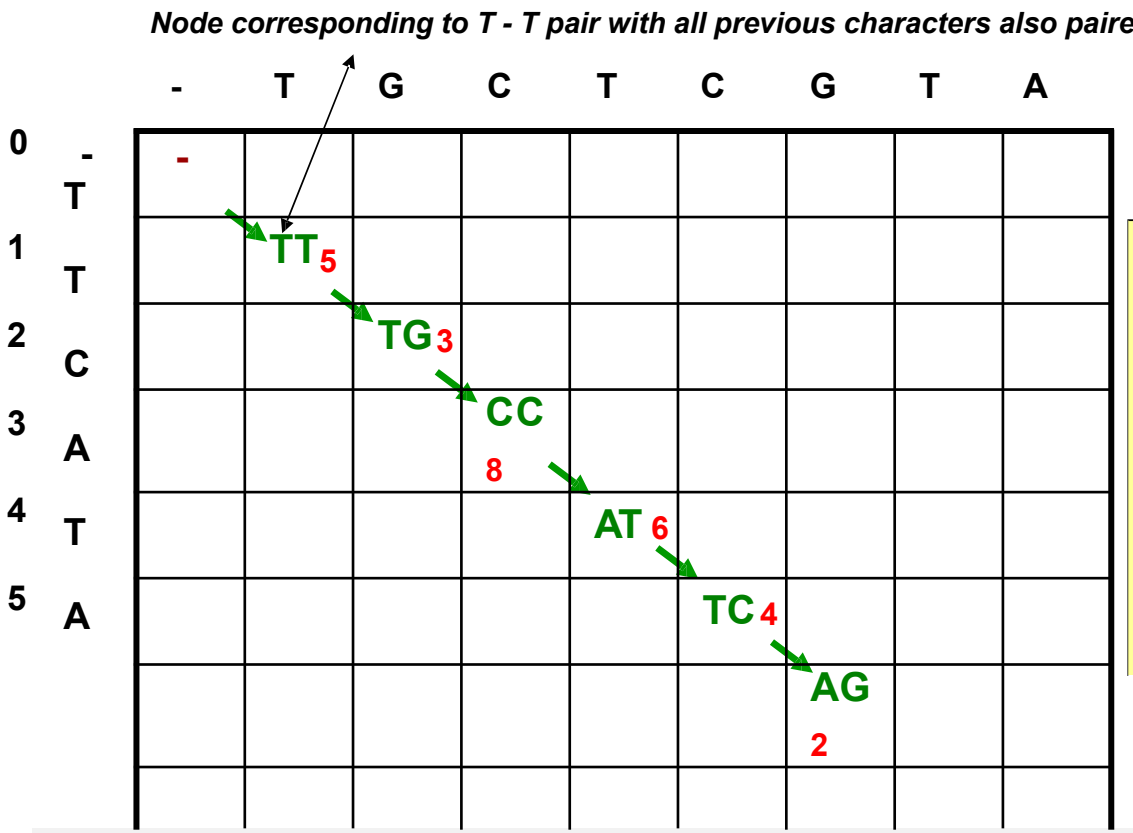
Smith-Waterman algorithm (Needleman-Wunsch)

Path Matrix



Path matrix: Every node represents the **endpoint** of an alignment, i.e. all characters above and to the left are aligned. Algorithms calculate the optimal score for alignment to a node and use this as basis for calculation of all possible paths starting from there.

Filling the Path Matrix (arbitrary choice)



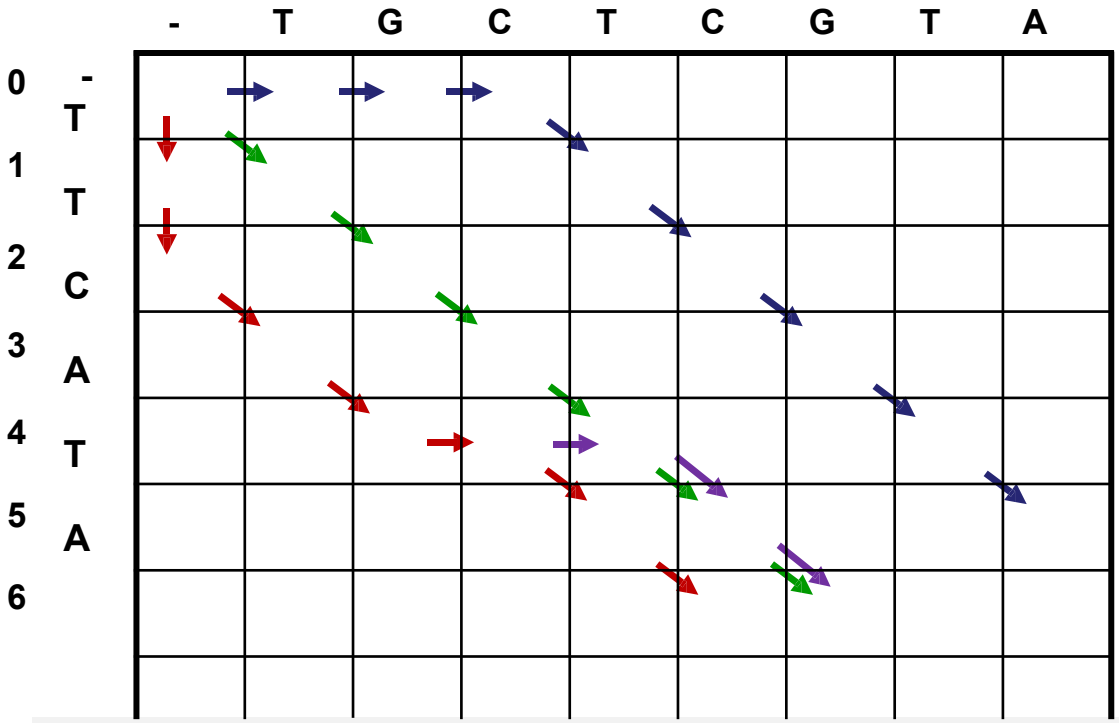
Score function uses weights for matches, mismatches and gaps:
e.g.
Match = +5
Mismatch = -2
Gap = -6

-TGCTCGTA
-TTCATA

Score = 2

Score = Sum of alignment scores up to this position

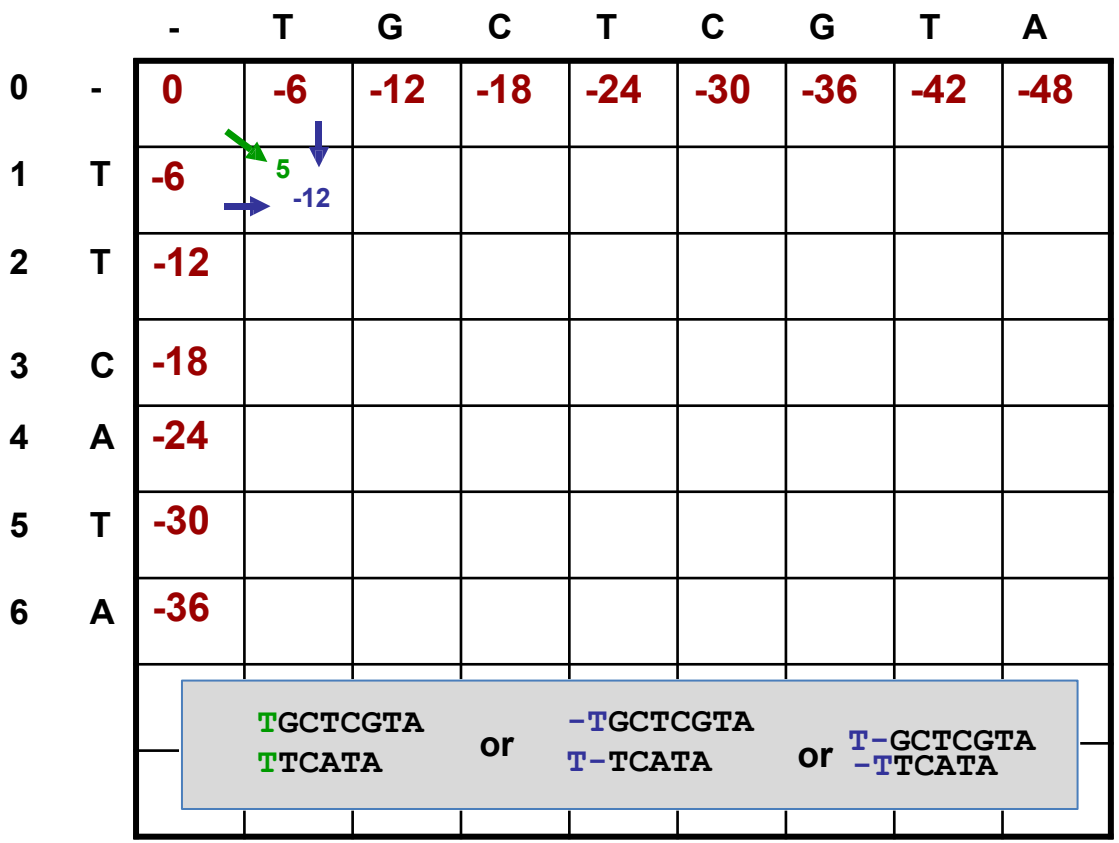
Filling the Path Matrix (by many ways)



Match = +5
Mismatch = -2
Gap = -6

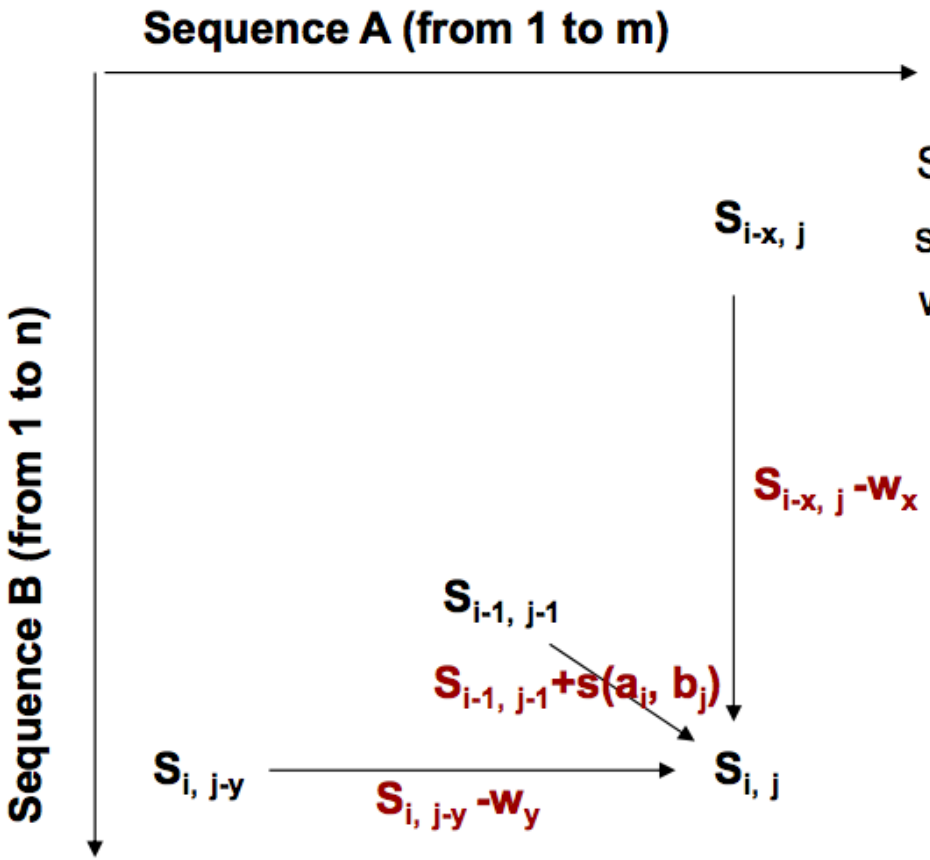
```
-TGCTCGTA      -TGCTCGTA      ---TGCTCGTA    ---TGCTCGTA
----TTCATA     -TTCATA        -TTCA-TA         -TTCA--TA
```

Filling the Path Matrix (optimal)



Match = +5
Mismatch = -2
Gap = -6

Formalism



$S_{i,\ j}$ = score at pos. i and j in sequ. A and B

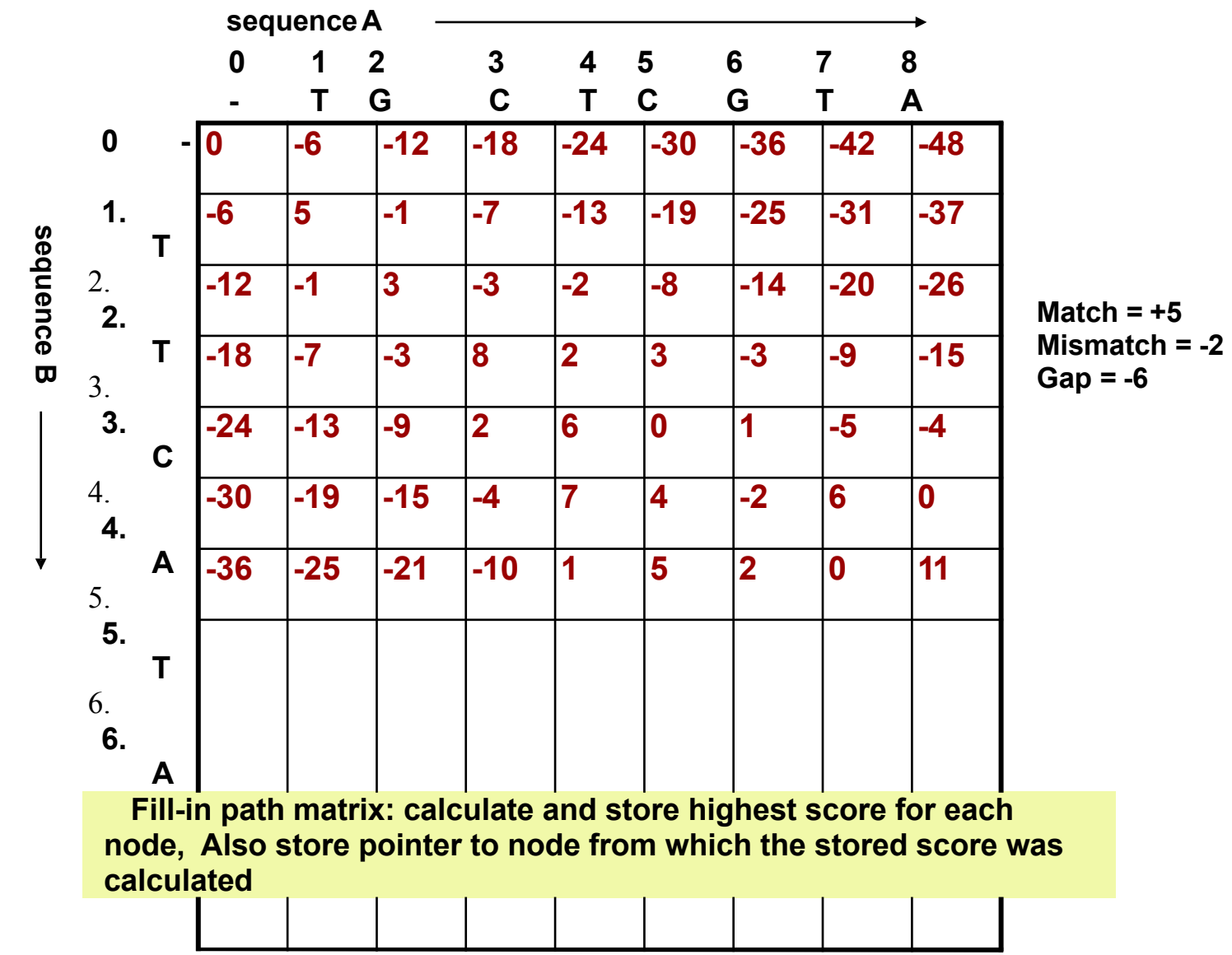
$s(a_i, b_j)$ = score for aligning characters i and j

w_x = penalty for a gap of length x

Calculate for each node: $S_{i,\ j} = \max \{S_{i-1,\ j-1} + s(a_i, b_j), \max_{y \geq 1} (S_{i,\ j-y} - w_y), \max_{x \geq 1} (S_{i-x,\ j} - w_x)\}$

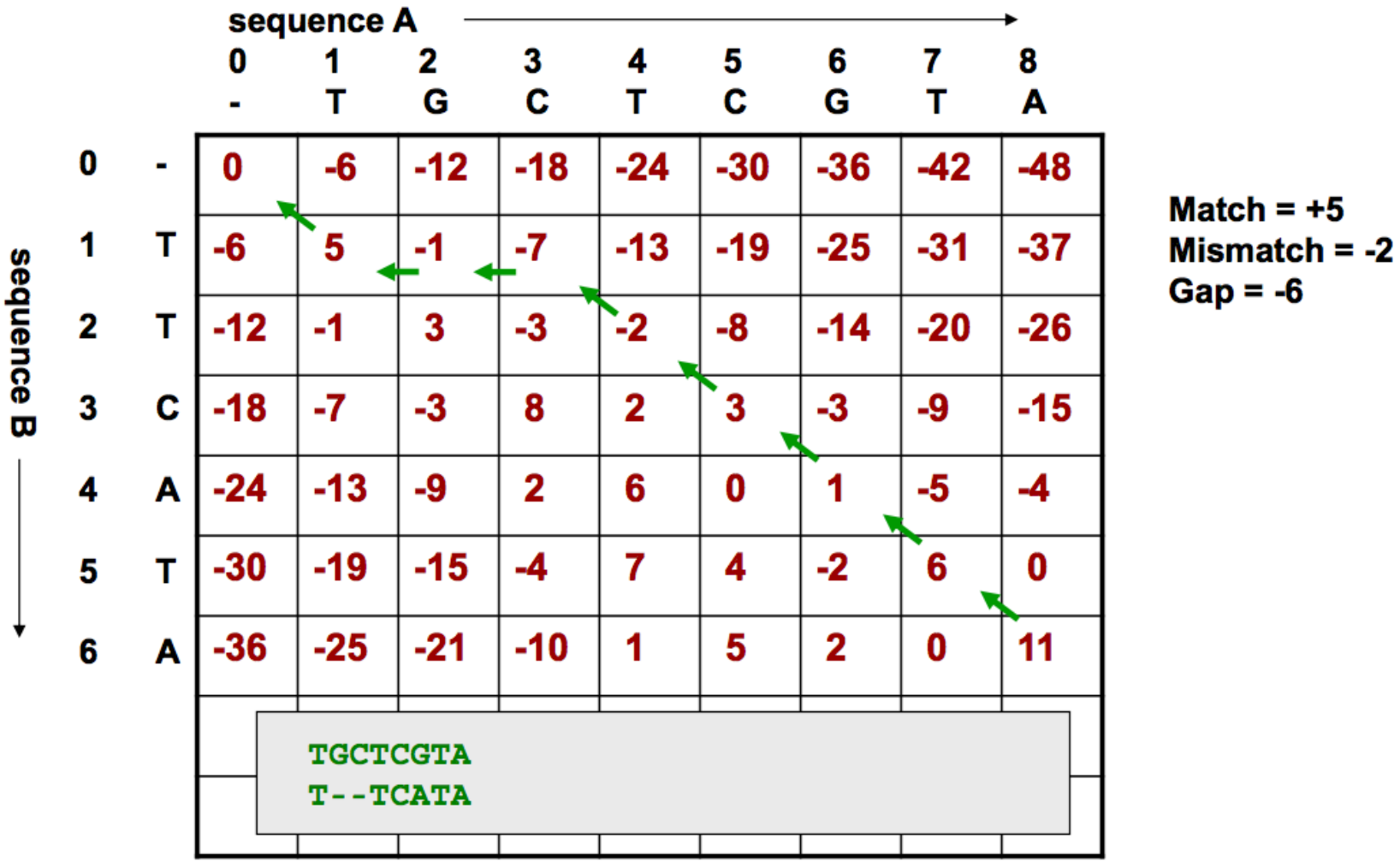
Store the maximal value and the pointer to the node which was used to calculate the value => trace-back matrix

Filling the entire Path Matrix



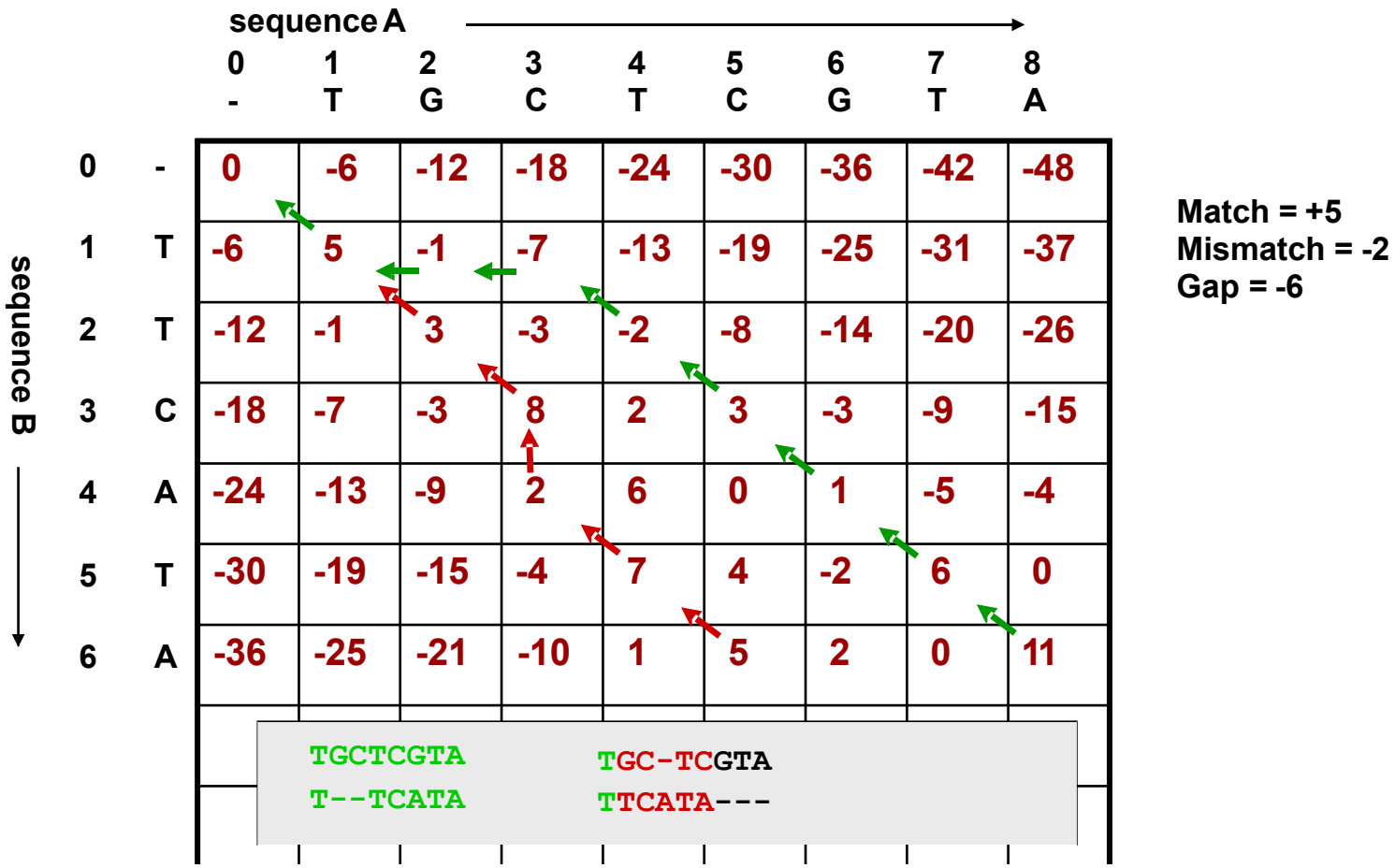
Fill-in path matrix: calculate and store highest score for each node, Also store pointer to node from which the stored score was calculated

Trace-back step: optimal alignment



Trace-back path matrix: reconstruct optimal path leading to the highest score

Trace-back step: suboptimal alignment



Trace-back path matrix: the optimal path may not produce the highest score at every step

Optimal alignment

Algorithms: **Needleman-Wunsch** (global) / **Smith-Waterman** (local)

Highest possible alignment score guaranteed

But

calculation-time and storage **intensive**:

n x m to n x m^2 calculation steps required (n < m); n x m for storage

Too slow for database searches

Solution: restrict search space by pre-selection of "promising" regions

Faster sequence alignment: heuristics

BLAST: Basic Local Alignment Search Tool (Altschul et al., 1990; 1997)

FASTA: FAST-All (Lipman and Pearson, 1985; Pearson and Lipman, 1988)

Definition "Heuristic": An algorithm that usually, but not always, works or that gives nearly the right answer.

Principle:

Sequences with significant similarity contain short strings (words) with identity

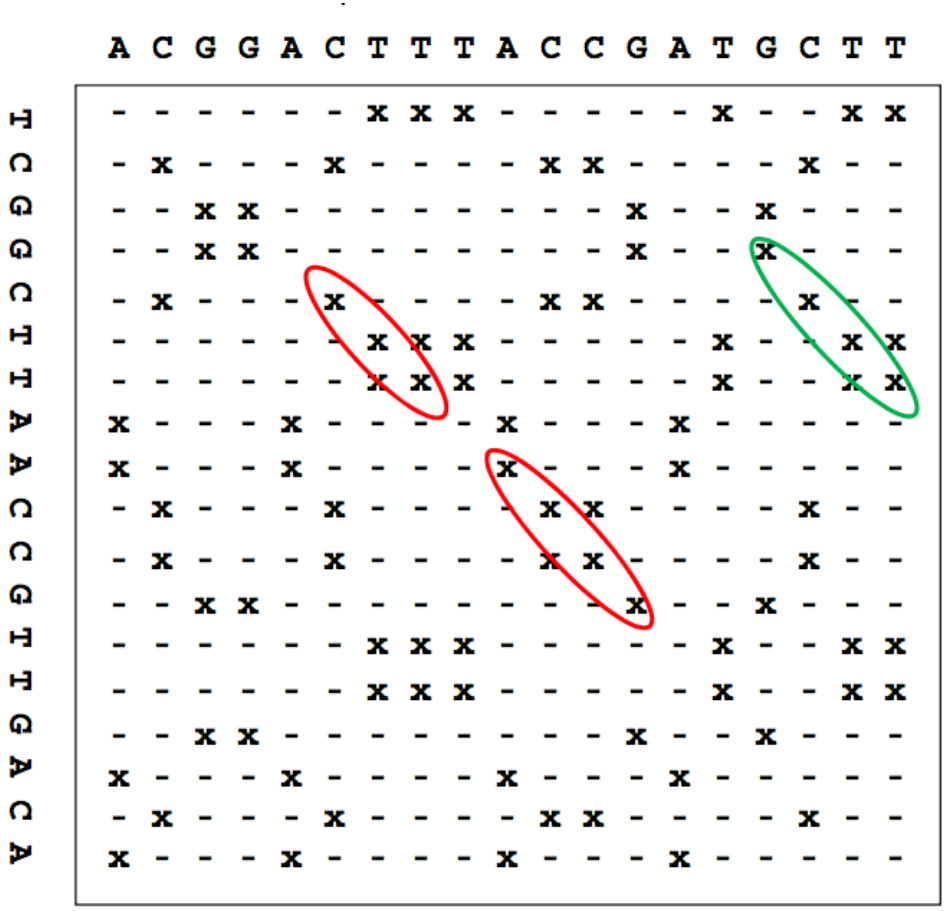
1. Divide query in all possible words (1 to 4 for amino acids; 6 to 14 for DNA) word lengths: "k-tuples"
2. Determine positions of matching words in each database sequence
   => hot-spots, hits
3. Attempt to extend hit-alignments in both directions without introduction of gaps
   => high scoring segment pairs (hsp)

4. Extend alignments with introduction of gaps

Speed and sensitivity depend on search-parameters and on the choice of primary hits that will be processed further.

FASTA (Pearson and Lipmap, 1988)

Step 1: define “promising” diagonals: search for ungapped regions sharing more than one exact k-tuple

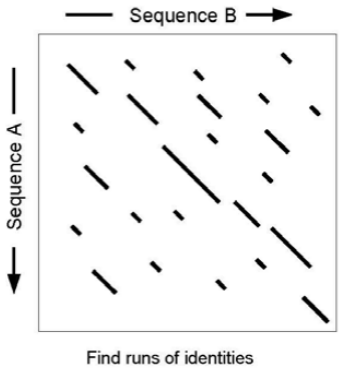


Hash table

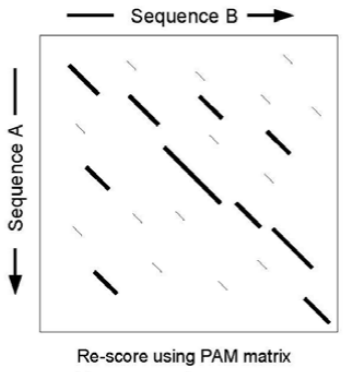
Word (only some shown)	Pos vertic. Seq	Pos horiz. Seq	Pos Hor - ver
CGG	2	2	0
CTT	5	6, 17	1, 12
TTA	6	8	2
GCT	4	16	12
ACC	9	10	1
CCG	10	11	1
GAC	15	4	-11

Database entry, Precalculated

FASTA (Pearson and Lipmap, 1988)



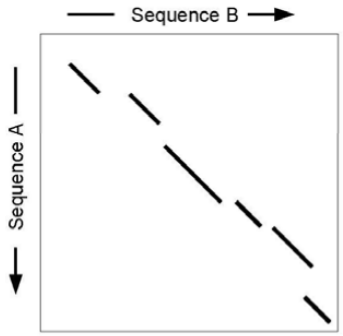
Find runs of identities



Re-score using PAM matrix
Keep top scoring segments.

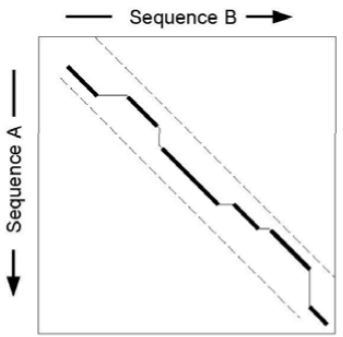
speed is largely determined by ktup size for finding the initial identities: the longer the faster, but with reduced sensitivity

(c)



Apply "joining threshold"
to eliminate segments that
are unlikely to be part of the alignment
that includes highest scoring segment.

(d)



Use dynamic programming
to optimise the alignment in a
narrow band that encompasses
the top scoring segments.

http://en.wikipedia.org/wiki/FASTA

BLAST: Basic Local Alignment Search Tool

Altschul et al., 1990; 1997

Similar to FASTA, with some alterations

1. Define matching (not only identical) words with scores above a given threshold. Word size e.g. 11 for DNA or 3 for proteins. = hits

2. Search two hits within a predefined distance (e.g. <40 amino acids) on a diagonal and combine them in a high scoring segment pair (HSP)

3. Initiate gapped extension (dynamic programming) only on the best HSP

BLAST words (neighborhood words)

Sequences are split in words of defined length (k-tuples, e.g. 3)

"Neighborhood "words that match these above a fixed threshold are calculated with a substitution matrix

Sequence: **AGSDDFTSSCILVYAGLIWDETNMYYHCATTLIELDKRRK….**

3-tuple	Match (Score)	1. Pos. Subst.	2. Pos. Subst.	3. Pos. Subst.	Mult. Subst.
AGS	AGS (14)	SGS (11)	none	AG(*A,N,T*) (10+)	none
GSD	GSD (16)	none	G(*A,R,N,D,S,Q,K,M,P,T*)D (12+)	GS(*E,N*) (10+)	none
SDD					
DFT					
.					
.					
.					
YHC	YHC (24)	*X*HC (17+)	Y*X*C (15+)	YH*X* (16+)	(*F,W*)NC (10+)

Scores from the Blosum62 substitution matrix
word length 3 and threshold score 11 are defaults in WWW BLAST searches,
word length can be altered to 2 but no changes of threshold score are possible

Word tables are precalculated for database entries and are used in the initialization step: computationally advantageous

BLAST sequence alignment

matching words:

the sum of substitution values (derived from a scoring matrix) for a word pair must exceed a predefined threshold (is often fixed in web-based applications)

successful hit-extension:

extension occurs as long as the new score does not drop more than a defined threshold below the so far obtained highest score

output:

longest alignment that cannot be improved by further elongation
theory of substitution score for ungapped alignments is well established no

theory for introduction of gaps => empirical gap penalties

choice of gap penalties and substitution matrix influences output

Significance of BLAST hits

Bit score: $S' = (\lambda S - \ln K)/\ln 2$

S: raw score

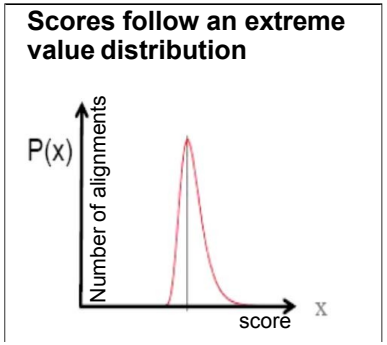
λ: log base of scoring matrix

K: scale of search space size

Expectation value: $E = mn2^{-S'}$

m: length of query

n: length of database sequence



Expectation value: Frequency of an accidental alignment with the respective score in a given search procedure (=comparison of obtained score with scores of all other alignments obtained in the search); the smaller the better

Some other programs use

Z score: $Z = \frac{\text{(score – average score of N permutations)}}{\text{standard deviation of randomized score distribution}}$

Z score: compares actual score to score of N (e.g. 100) randomized sequences with the same character frequencies; $Z \geq 3$ often regarded as significant
(note: this is not the z score of FastA!)

Implementations and methods

Alignment methods

BLAST	http://www.ncbi.nlm.nih.gov/; ftp://ncbi.nlm.nih.gov/blast
wuBLAST	http://www.ebi.ac.uk/ http://www.ebi.ac.uk/Tools/sss/
FASTA	fasta/nucleotide.html http://www.ch.embnet.org/software/
LALIGN	LALIGN_form.html http://www.sanger.ac.uk/resources/
DOTTER	software/seqtools/ http://www.ebi.ac.uk/Tools/msa/ http://
Multiple seqAlign	multalin.toulouse.inra.fr/multalin/
MultAlin	

Motifs and patterns

BLOCKS	http://blocks.fhcrc.org http://www.sanger.ac.uk/resources/
Pfam	databases/pfam.html http://expasy.org
PROSITE (+many links)	

Presentation Methods

ALSCRIPT	http:// http://www.csb.yale.edu/userguides/seq/alscript/

Conversion utilities

http://www.ebi.ac.uk/Tools/sfc/

Phylogenetic resources

(huge collection of links): http://evolution.genetics.washington.edu/phylip/software.html

Search for Life Science Web services

http://www.biocatalogue.org/

http://en.wikipedia.org/wiki/Sequence_alignment_software

Websites for sequence searches

EBI: European Bioinformatics Institute (EMBL)
http://www.ebi.ac.uk/
FASTA3, wuBlast2

NCBI: National Center for Biotechnology Information (NIH)
http://www.ncbi.nlm.nih.gov/
Blast in all flavours

Search space in GenBank Aug. 2015: 199'823'644'287 bases in 187'066'846 sequences

~~**+ 1'163'275'601'001 bases in 302'955'543 WGS records**~~

Local sequence searches:
GCG package (UNIX based; FASTA and many other options) Downloadable

stand-alone versions of new ncbiBLASTs and wuBLAST2

Comparison of search performance

Protein family	Smith-W.	oriBLAST	BLAST	PSI-BLAST
Serine Protease	275	273	275	286
Ras	255	249	252	375
Globin	28	26	28	623
Cytochrome P450	211	197	211	224
run time	36	1.0	0.34	0.87

Altschul et al., Nucleic Acids Res. 25, 3389-3402 (1997)

Use of ambiguous words, pattern profiles

General substitution matrices are build on collections of „all proteins“

In functional sequence elements or protein motives the variation of some (or all) sequence positions may be restricted by functional constraints.

This can be modelled more specifically by

substitution matrices build only on a specific motif

Hidden Markov Models (HMMs) for a specific model

Regular expressions

e.g. ***PSI BLAST*** *(Position specific iterative BLAST) uses a detected alignment to calculate a new PSSM (position specific scoring matrix) and performs a new BLAST with this PSSM etc. More distant relationships may be detected.*

Precalculated PSSMs for known protein domains form the basis of „Conserved domain“ CD search, automatically performed whenever protein sequences are submitted to BLAST

BLAST programs

programs	Database	Query	Comments
blastp	protein	protein	finds also distant relationships
blastn	nucleotide	nucleotide	default for close relationships
blastx	protein	translated nucleotide	useful for analysis of new DNA and EST sequences
tblastn	translated nucleotide	protein	unannotated coding regions in database sequences
tblastx	translated nucleotide	translated nucleotide	EST analysis

Special BLAST programs

programs	Comments
BLAST1.4	first BLAST version
QBLAST =BLAST2.0	current NCBI default; “2 hit search strategy” for increased speed; performs “gapped BLAST”
PSI-BLAST	Position-Specific Iterative BLAST: generates a PSSM from multiple alignments of a protein query to a database and uses the PSSM repeatedly to search for more distant hits PSSM = position-specific scoring matrix *(option in blastp)*
PHI-BLAST	Pattern Hit Initiated BLAST: seeks for alignments that preserve a specific protein motif *(option in blastp)*
RPS-BLAST	Reverse Position-Specific BLAST: compares a protein query to predefined PSSMs for known conserved protein domains. Invoked by activating CD search in the BLAST window
Align 2 sequences	pairwise alignment of two defined sequences (*now also incorporated as option in different BLAST programs)*
Taxonomy BLAST	lists BLAST hits according to taxonomy

Special BLAST programs

programs	Comments
MegaBLAST	optimized for aligning longer sequences that differ only slightly uses longer words and a different algorithm (“greedy algorithm”) faster, works with longer sequences than BLAST for nucleic acids *(option in blastn)*
discontiguous MegaBLAST	uses a different type of words for initiation of alignments: words can be discontiguous, e.g. 11 or 12 matches in a template region of 16, 18 or 21 nucleotides. Options for 1 or 2 initial hits implemented different possibilities to analyze coding and non-coding regions (differentiated by the importance of the third codon position) for nucleic acids *(option in blastn)*
BLAST to find short, almost perfect matches	now many BLAST procedures recognize query length and adapt search parameters automatically (option can be deactivated)

BLAST programs can be used with many different databases or subsections of databases

Output

Contains information of type and quality of matches

	Score (bits)	E-value
gi|1172846|sp|Q09028|RB48_HUMAN CHROMATIN ASSEMBLY FACTOR...	131	4e-31

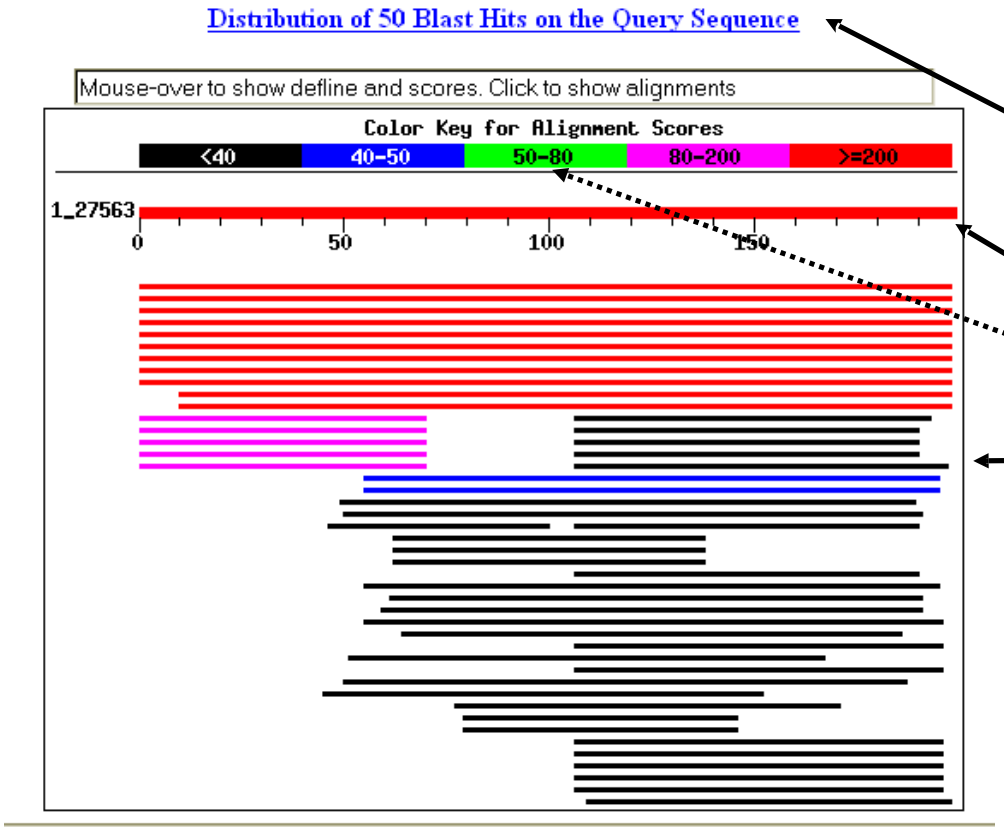
Score: raw score is calculated according to the number and weight of matches and gaps
depends on substitution matrix used

bits score is normalized with parameters reflecting the search strategy

E-value: statistic significance of hit; describes roughly how often a hit with a given score can be expected to occur randomly with the applied search strategy

with short queries no low E-values can be expected!!

Output



BLAST output

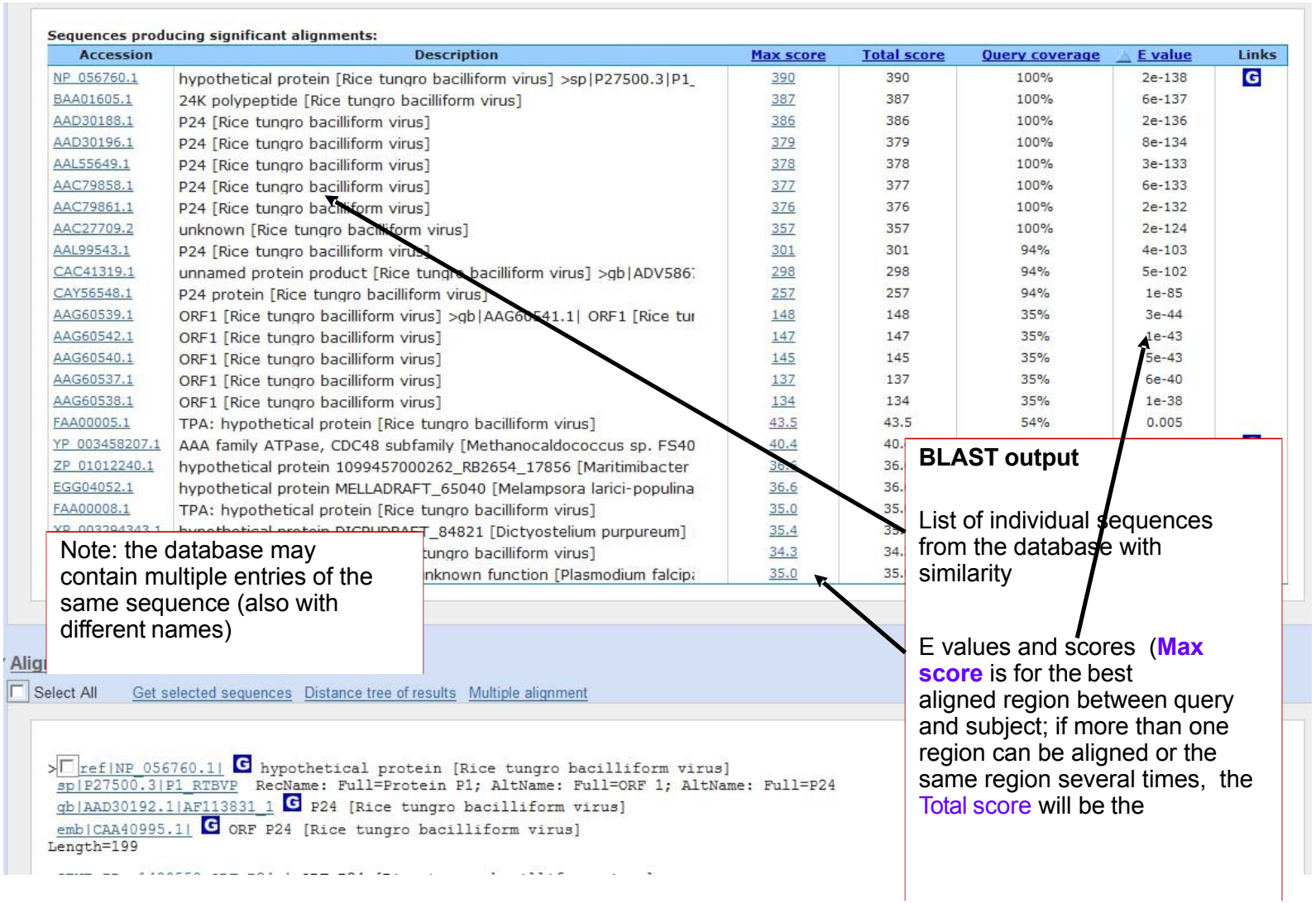
Number of detected hits (= similar sequences in the database)

Query sequence (=Input)

Regions where subject sequences (from the database) are similar (color code for alignment scores)

Page continues

Output



Sequences producing significant alignments:

Accession	Description	Max score	Total score	Query coverage	E value	Links
NP_056760.1	hypothetical protein [Rice tungro bacilliform virus] >sp|P27500.3|P1_	390	390	100%	2e-138	
BAA01605.1	24K polypeptide [Rice tungro bacilliform virus]	387	387	100%	6e-137	
AAD30188.1	P24 [Rice tungro bacilliform virus]	386	386	100%	2e-136	
AAD30196.1	P24 [Rice tungro bacilliform virus]	379	379	100%	8e-134	
AAL55649.1	P24 [Rice tungro bacilliform virus]	378	378	100%	3e-133	
AAC79858.1	P24 [Rice tungro bacilliform virus]	377	377	100%	6e-133	
AAC79861.1	P24 [Rice tungro bacilliform virus]	376	376	100%	2e-132	
AAC27709.2	unknown [Rice tungro bacilliform virus]	357	357	100%	2e-124	
AAL99543.1	P24 [Rice tungro bacilliform virus]	301	301	94%	4e-103	
CAC41319.1	unnamed protein product [Rice tungro bacilliform virus] >gb|ADV586	298	298	94%	5e-102	
CAY56548.1	P24 protein [Rice tungro bacilliform virus]	257	257	94%	1e-85	
AAG60539.1	ORF1 [Rice tungro bacilliform virus] >gb|AAG60 41.1| ORF1 [Rice tu	148	148	35%	3e-44	
AAG60542.1	ORF1 [Rice tungro bacilliform virus]	147	147	35%	1e-43	
AAG60540.1	ORF1 [Rice tungro bacilliform virus]	145	145	35%	5e-43	
AAG60537.1	ORF1 [Rice tungro bacilliform virus]	137	137	35%	6e-40	
AAG60538.1	ORF1 [Rice tungro bacilliform virus]	134	134	35%	1e-38	
FAA00005.1	TPA: hypothetical protein [Rice tungro bacilliform virus]	43.5	43.5	54%	0.005	
YP_003458207.1	AAA family ATPase, CDC48 subfamily [Methanocaldococcus sp. FS40	40.4	40.			
ZP_01012240.1	hypothetical protein 1099457000262_RB2654_17856 [Maritimibacter	38.	36.			
EGG04052.1	hypothetical protein MELLADRAFT_65040 [Melampsora larici-populina	36.6	36.			
FAA00008.1	TPA: hypothetical protein [Rice tungro bacilliform virus]	35.0	35.			
XP_003284343.1	hypothetical protein DICPUDRAFT_84821 [Dictyostelium purpureum]	35.4	35.			
	tungro bacilliform virus]	34.3	34.			
	nknown function [Plasmodium falcip	35.0	35.			

Select All Get selected sequences Distance tree of results Multiple alignment

>ref|NP_056760.1| hypothetical protein [Rice tungro bacilliform virus]
sp|P27500.3|P1_RTBVP RecName: Full=Protein P1; AltName: Full=ORF 1; AltName: Full=P24
qb|AAD30192.1|AF113831_1 P24 [Rice tungro bacilliform virus]
emb|CAA40995.1| ORF P24 [Rice tungro bacilliform virus]
Length=199

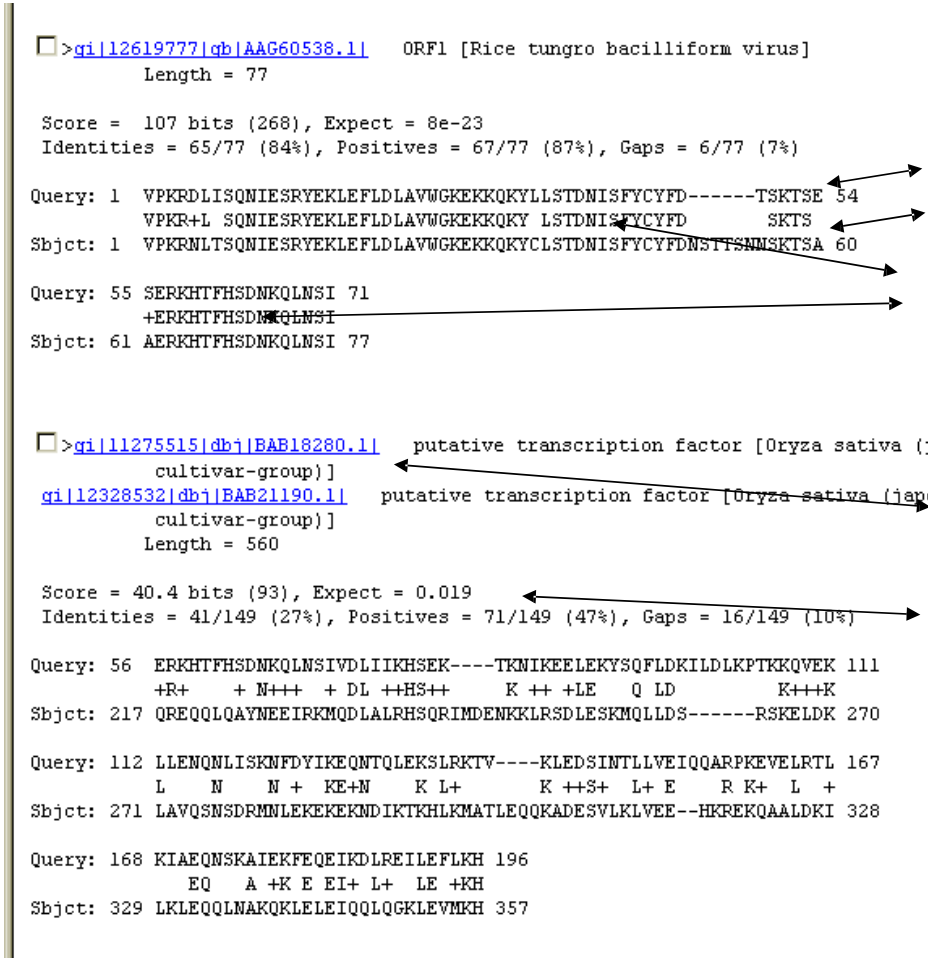
Note: the database may contain multiple entries of the same sequence (also with different names)

BLAST output

List of individual sequences from the database with similarity

E values and scores (**Max score** is for the best aligned region between query and subject; if more than one region can be aligned or the same region several times, the Total score will be the

Output



Actual sequence alignment:

Protein sequence in single letter code of query sequence (with gaps) and of detected similar sequence

Identities between the two sequences or similarities with a positive alignment score (+)

Next hit (represents in this case two identical database entries) Information about the quality

Scores and E values like here are

probably not indicative for a significant homology (unless they come from short, very good alignment regions)

Output interpretation

Data usage

not all protein sequences will yield helpful results not

all "significant" matches will be found

not all found matches must be "significant"

relevant matches:

Prediction of protein function
Definition of functionally significant sequence motives in proteins or nucleic acids

Determination of phylogenetic relationships (multiple alignments, tree construction)

Structure prediction

Acquisition of complete sequences from partial sequence information (EST, peptide, etc.) Guide

to intelligent mutagenesis for functional analyses

Assembly of complete genomes from partial sequences

References

substitution matrices

Gonnet, G.H., Cohen, M.A., and Benner, S.A. (1992) Exhaustive matching of the entire protein sequence database. Science **256**, 1443–1145.

Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences U.S.A. **89**, 10915–10919.

Schwartz, R.M. and Dayhoff, M.O. (1979) Matrices for detecting distant relationships, in Atlas of Protein Sequences and Structure (Dayhoff, M.O. ed.) **5**, National Biomedical Research Foundation, Washington, D.C., U.S.A. pp. 353–358.

alignment programs

Altschul, S.F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic Local Alignment Tool. Journal of Molecular Biology **215**, 403–410.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research **25**, 3389–3402.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48**, 443–453.

Pearson, W.B. (1998) Empirical statistical estimates for sequence similarity searches. Journal of Molecular Biology **276**, 71–84.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence analysis. Proceedings of the National Academy of Sciences U.S.A. **85**, 2444–2448.

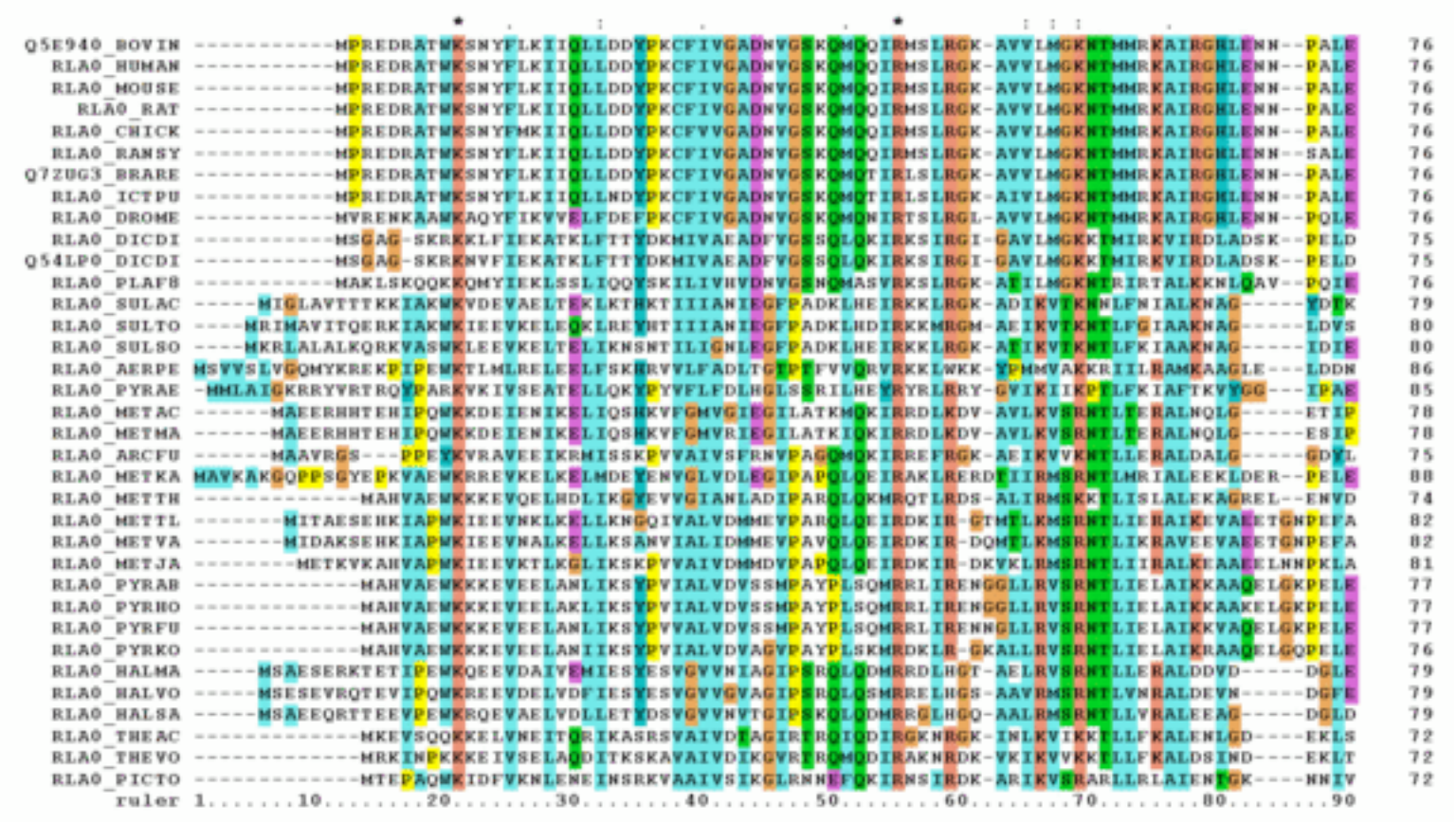
Smith, T.F. and Waterman, M.S. (1981) Comparison of bio-sequences. Advances in Applied Mathematics **2**, 482–489.

Wilbur, W.J. and Lipman, D.J. (1983) Rapid Similarity Searches of Nucleic Acid and Protein Data Banks. Proceedings of the National Academy of Sciences U.S.A. **80**, 726–730.

http://en.wikipedia.org/wiki/Sequence_alignment_software

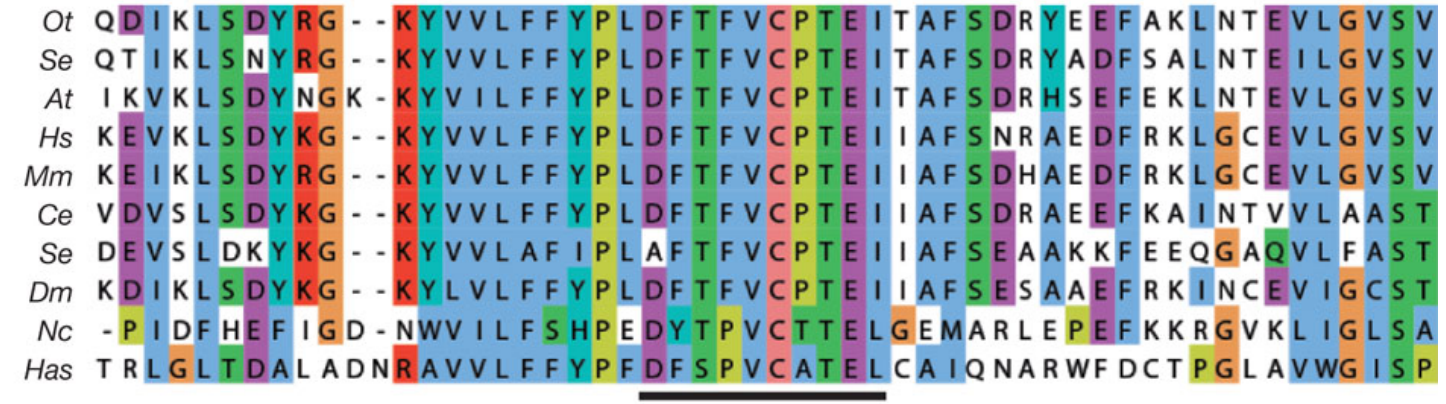
[Bonus Material: Multiple Sequence Alignment]

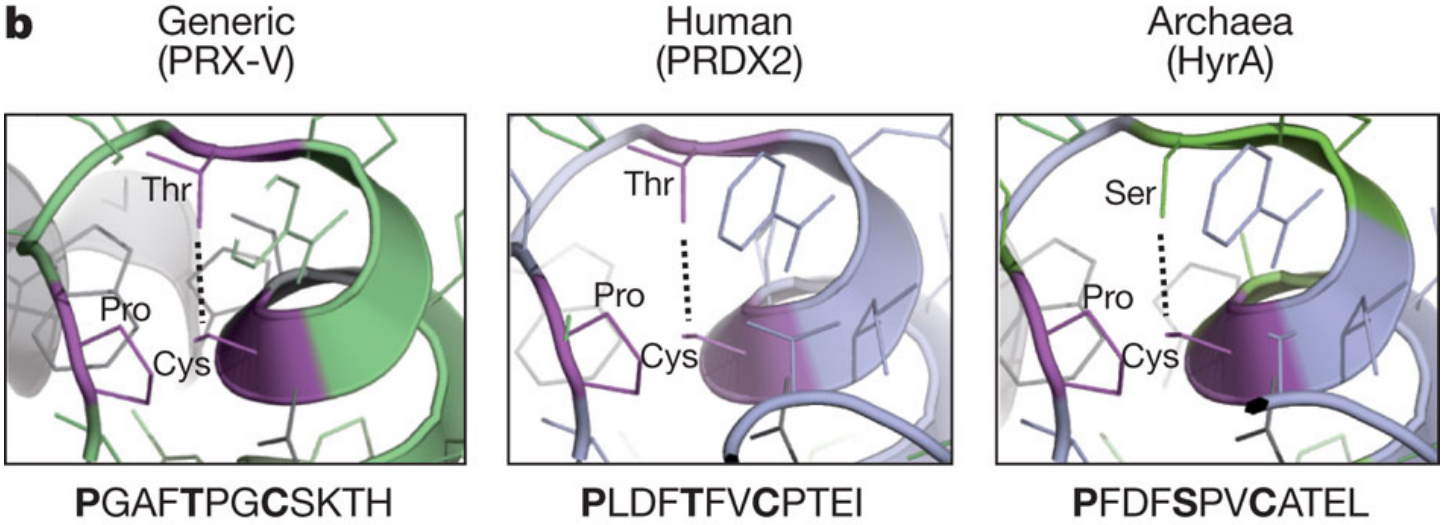
Multiple Sequence Alignment



Motivations for sequence alignment

1) to identify and check the state of “active sites”

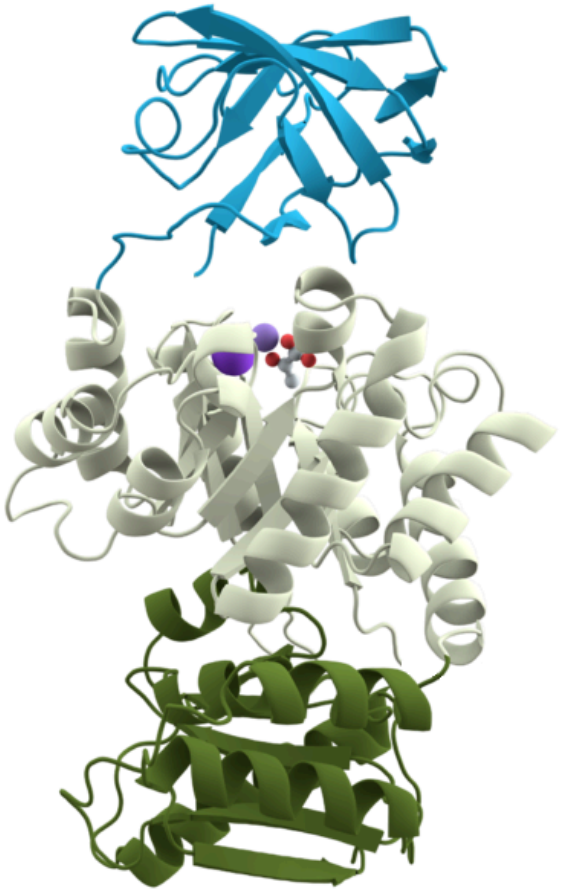




From: Peroxiredoxins are conserved markers of circadian rhythms. Nature 485, 459–464 (24 May 2012)

Motivations for sequence alignment

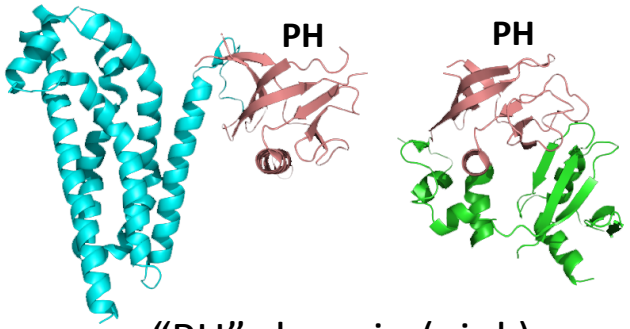
2) to identify and characterize “protein domains”



Pyruvate Kinase

definition:

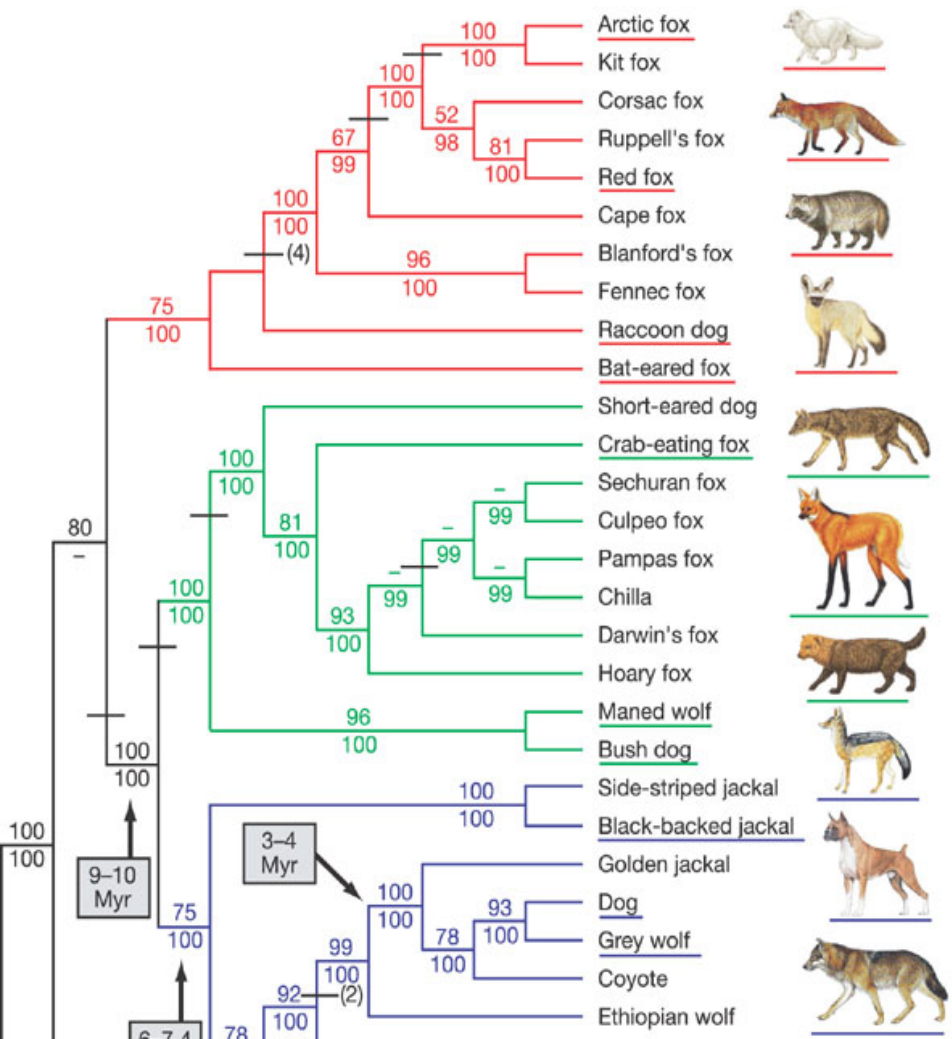
“parts of proteins that can evolve, function, and exist <u>independently</u> of the rest”

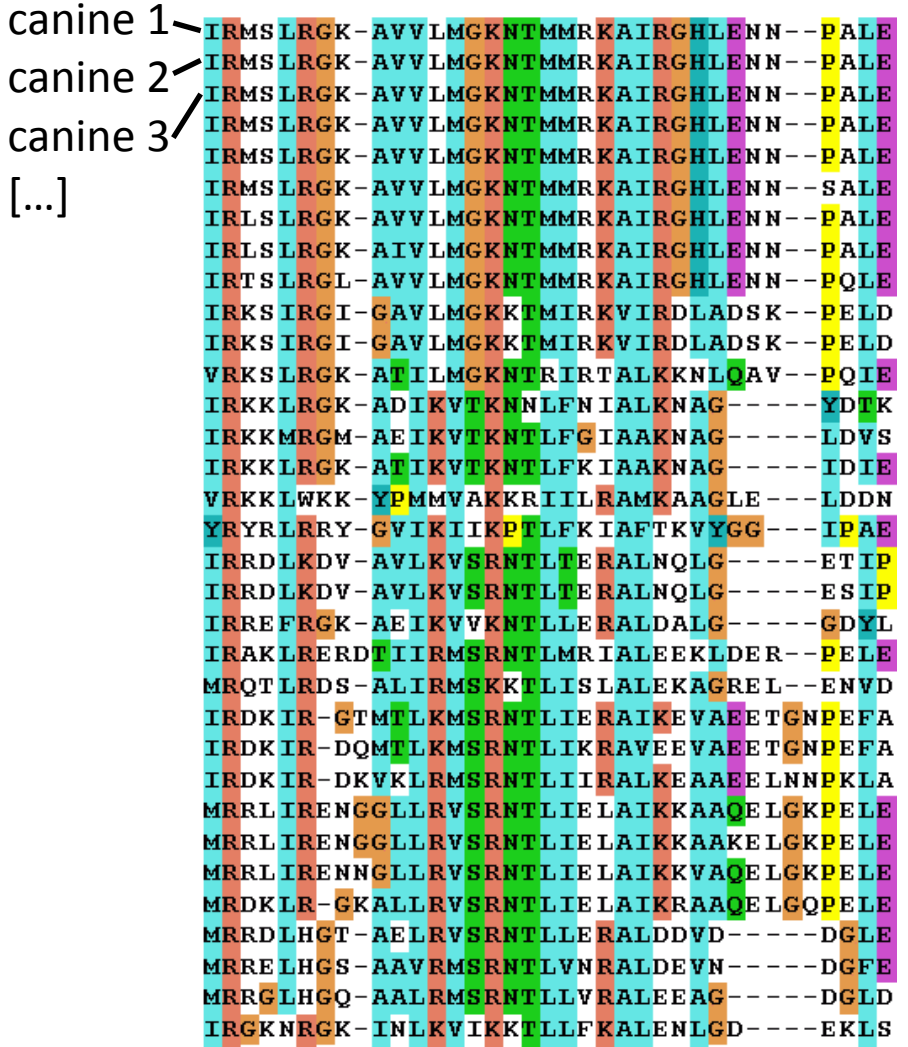


“PH”-domain (pink);
occurring in two different proteins

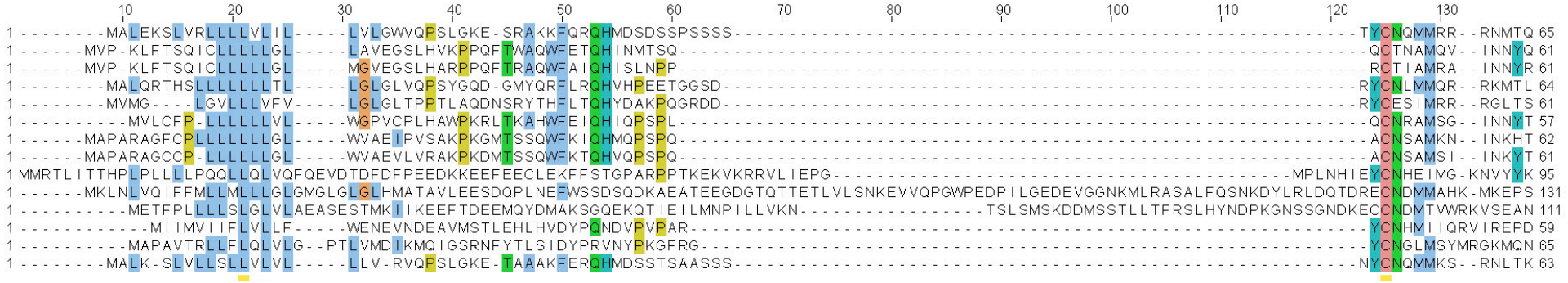
Motivations for sequence alignment

3) to make phylogenetic inferrences (“trees”)





Multiple Alignment



Combinatorial Explosion: very many possible solutions

Complexity: $O(\text{alignment\_length}^{\text{number\_seqs}})$

=> an NP-complete problem !!

Multiple Alignment

PROBCONS

MUSCLE

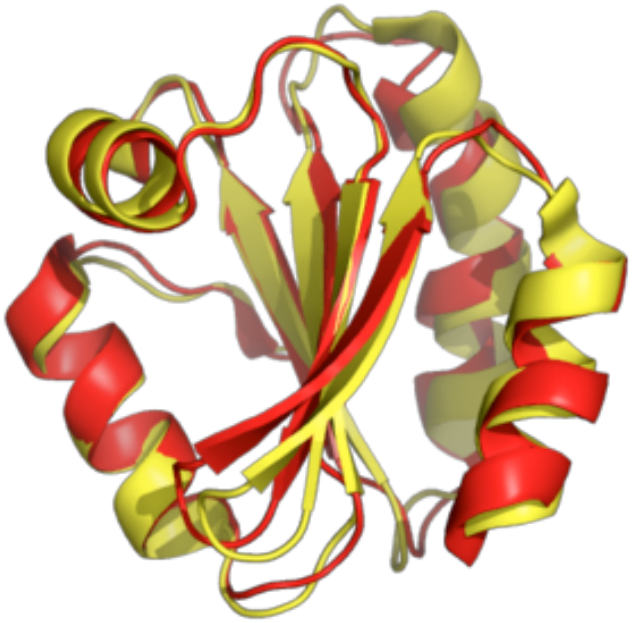
SATé

MAFFT

PRANK

DIALIGN-TX

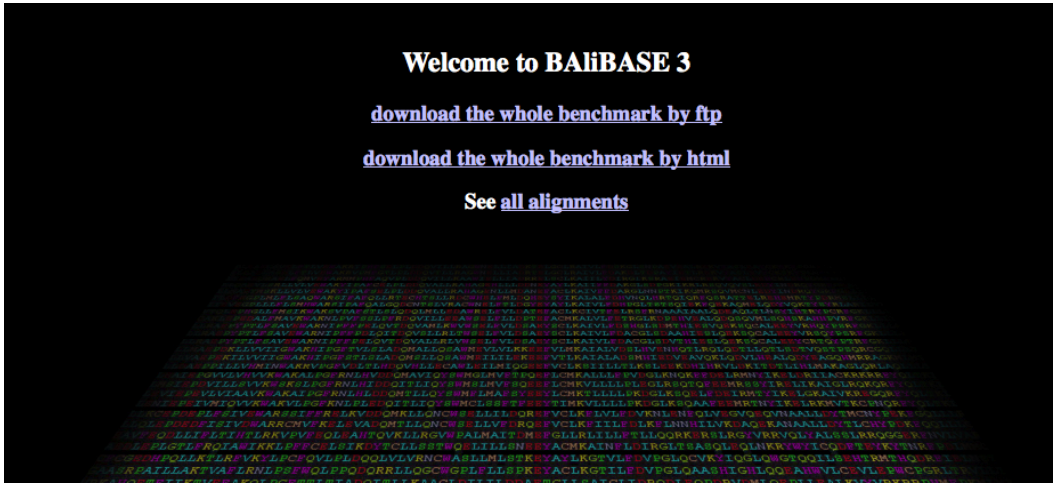
Quality of MSA: Benchmarking



Structural Aligments offer the best benchmarks !

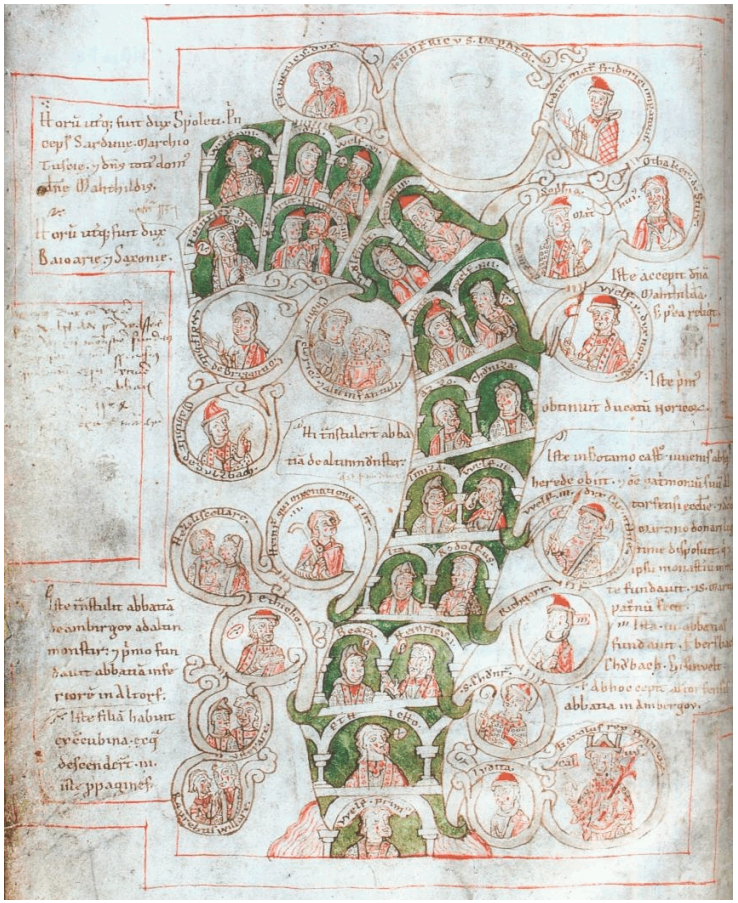
"BAliBASE":
Benchmark Alignment Database

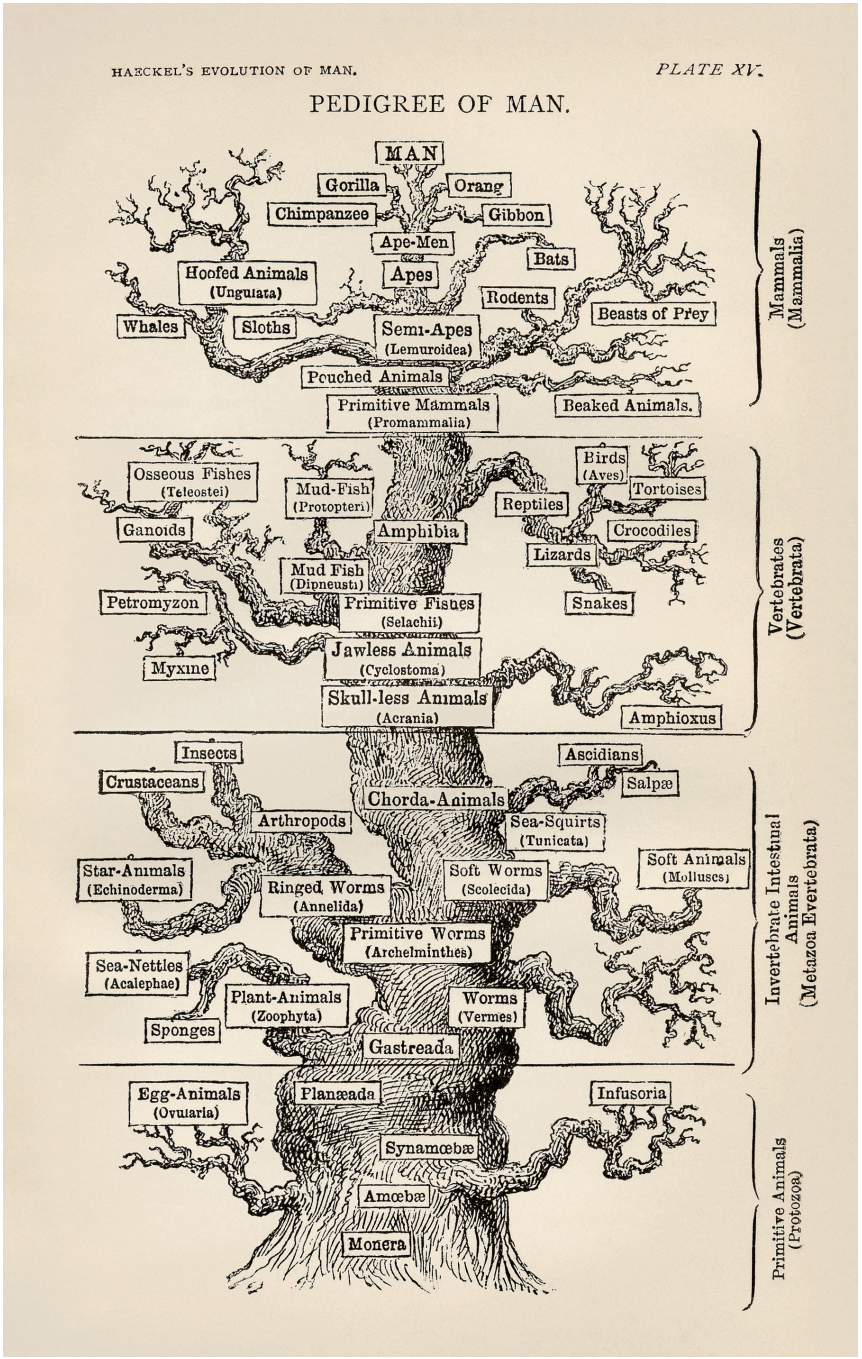
Hand-made multiple sequence alignments
Based on selected structural alignments



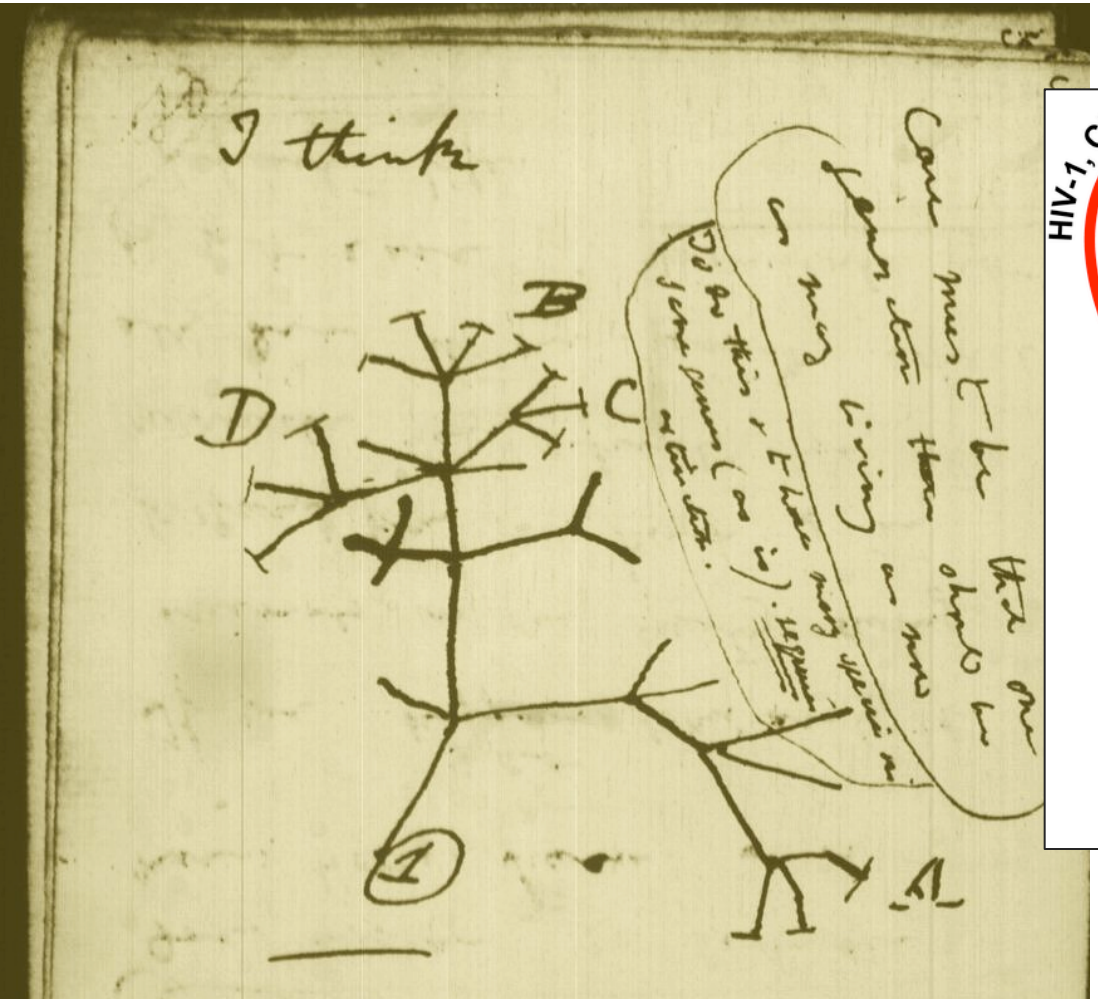
Phylogeny Reconstruction

- a quick overview -

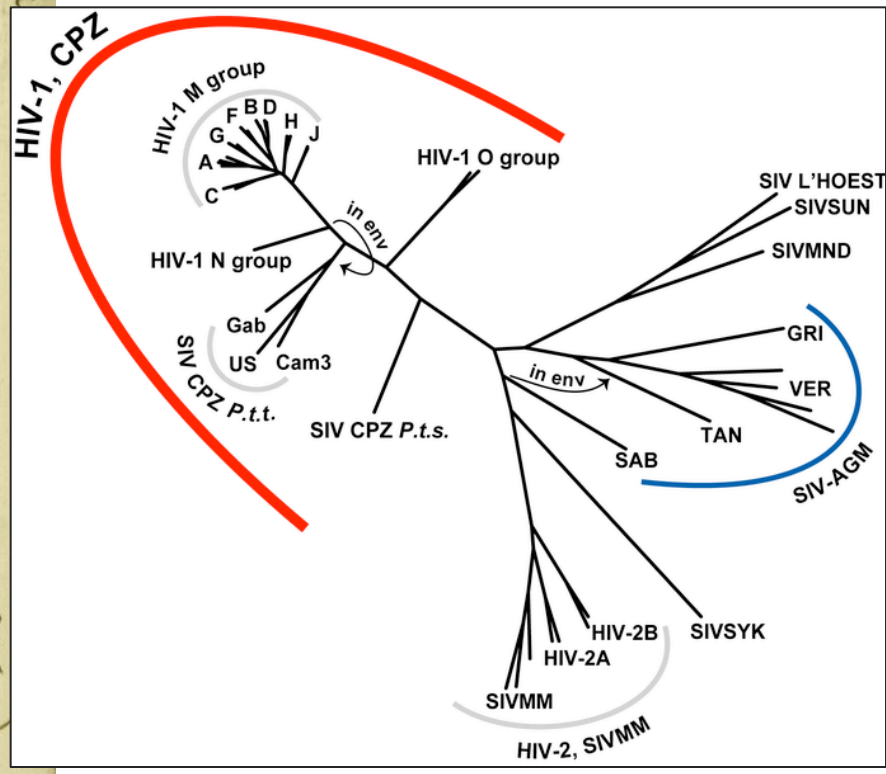




Some iconic phylogenetic trees

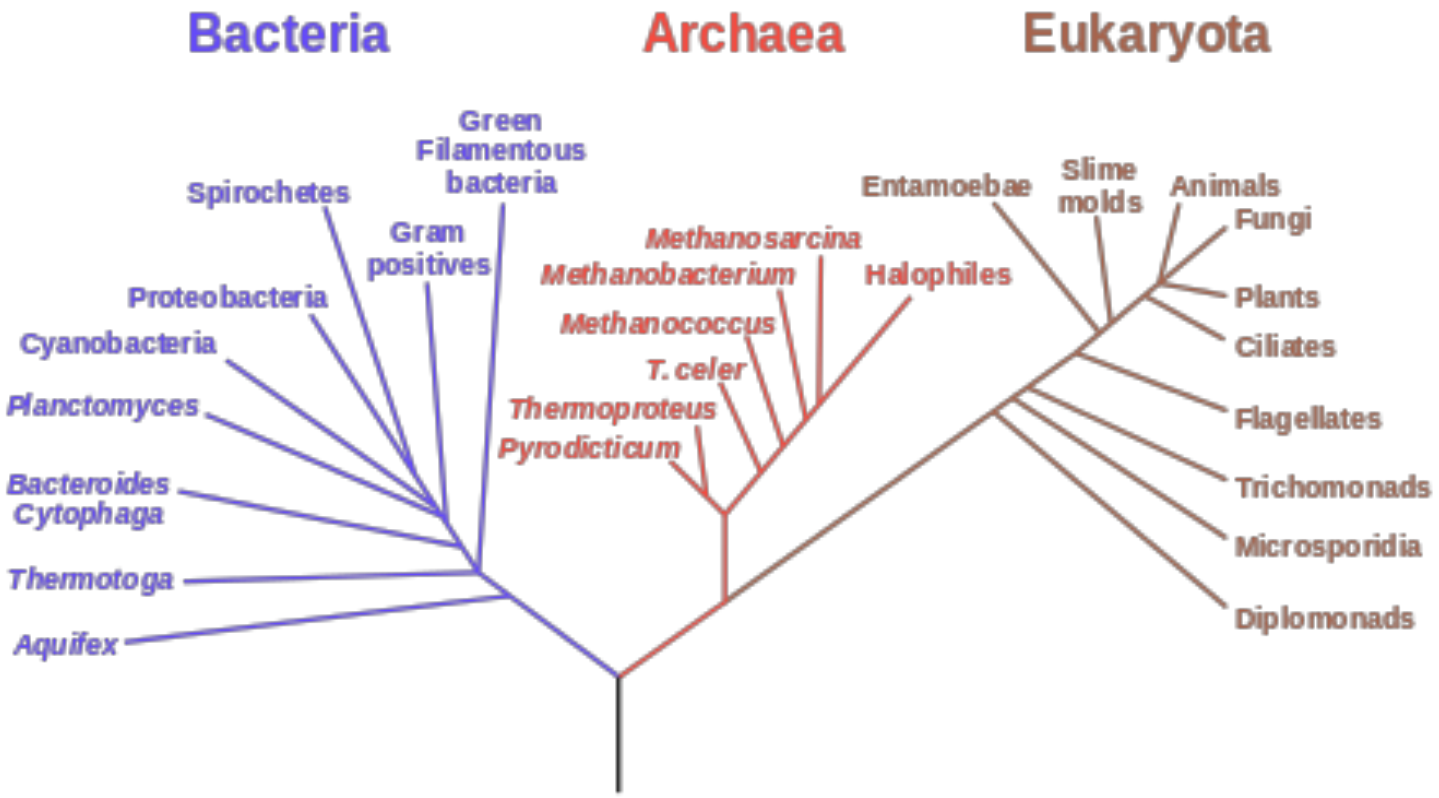


Charles Darwin, *personal notebook*



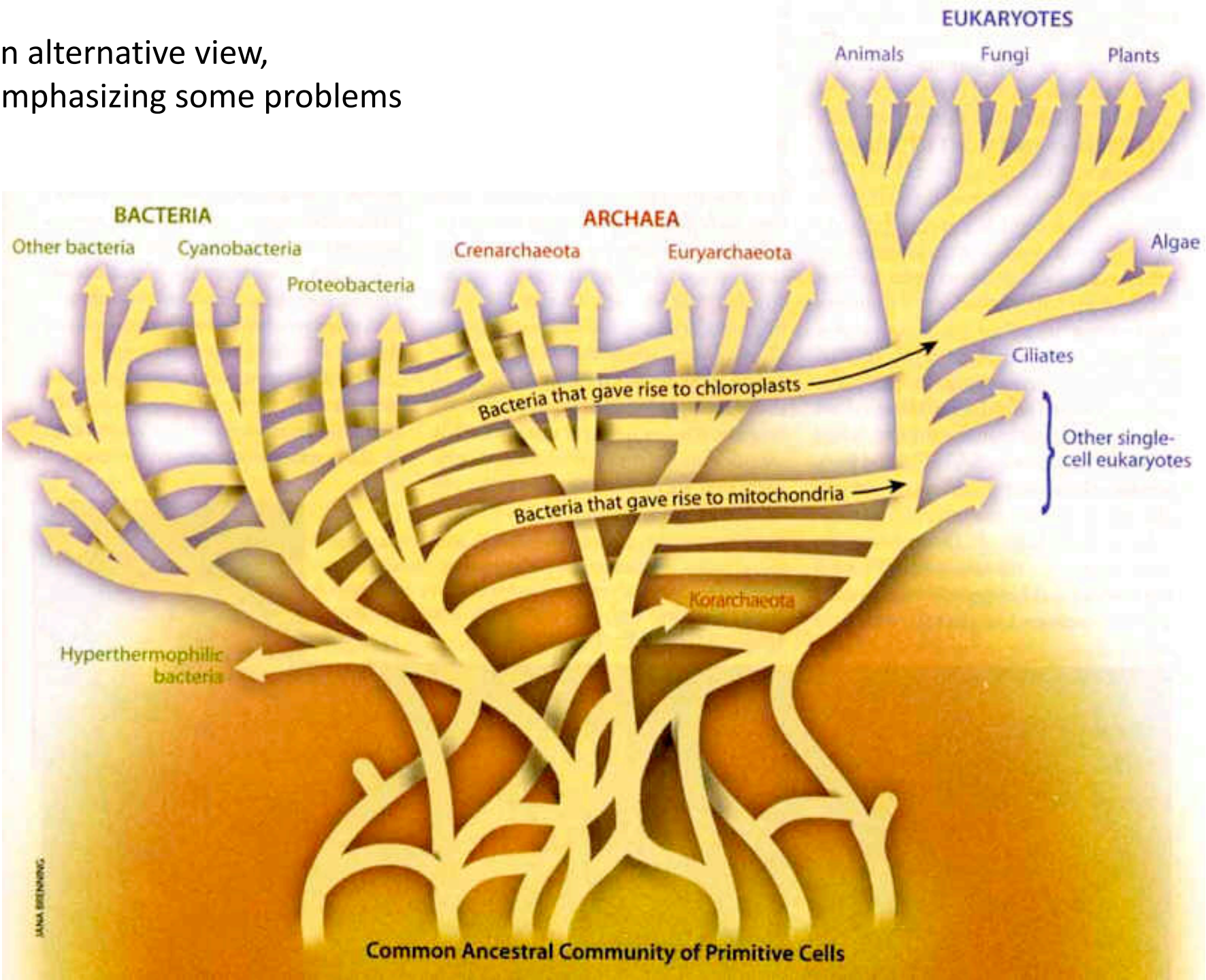
HIV (AIDS),
and closely related
animal viruses.

Some iconic phylogenetic trees



current, conventional version of the ‘tree of life’

an alternative view,
emphasizing some problems



Generating phylogenetic trees

- from gene/protein sequences -

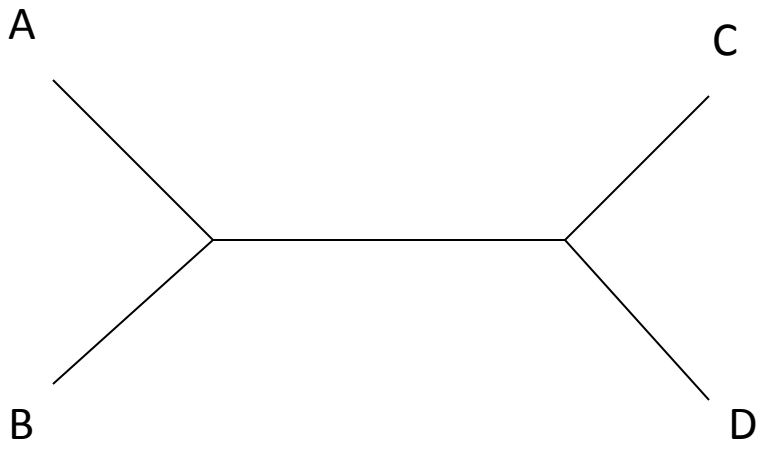
- Phenetic: trees are constructed based on observed characteristics directly, not on evolutionary history

} Distance methods

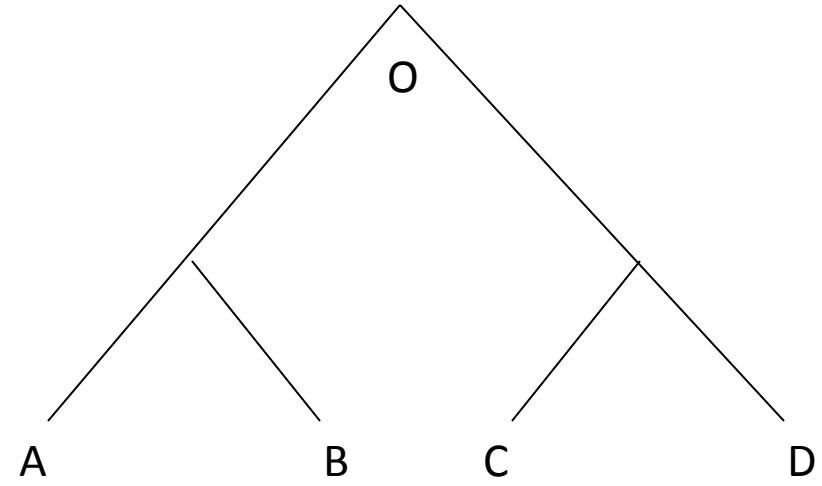
- Cladistic: trees are constructed based on fitting observed characteristics to some model of evolutionary history

} Parsimony and Maximum Likelihood methods

Unrooted tree



Rooted Tree



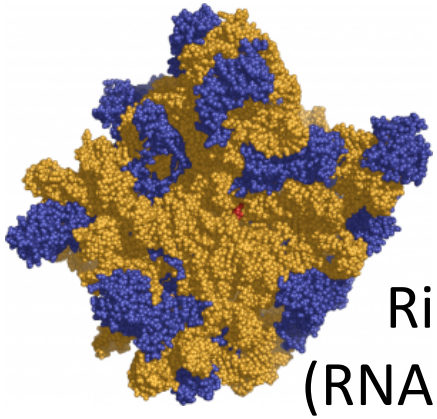
Numer of topologies for m taxa

M	Rooted tree	UnRooted Tree
	$(2m-3)! / 2^{m-2}(m-2)!$	$(2m-5)! / 2^{m-3}(m-3)!$
2.	1	1
3	3	1
4	15	3
5	105	15
6	945	105
7	10395	945
8	135135	10395
9	2027025	135135
10	34459425	2027025

Which genes to use ?

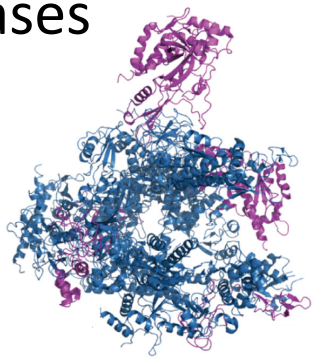
suitable marker genes …

- **… should occur in every organism**
- **… should rarely undergo horizontal transfer**
- **… should be evolving 'slowly'**
- **… should only occur in one copy per genome**
- **… should function in a process that sees no change**



Ribosomes
(RNA or proteins)

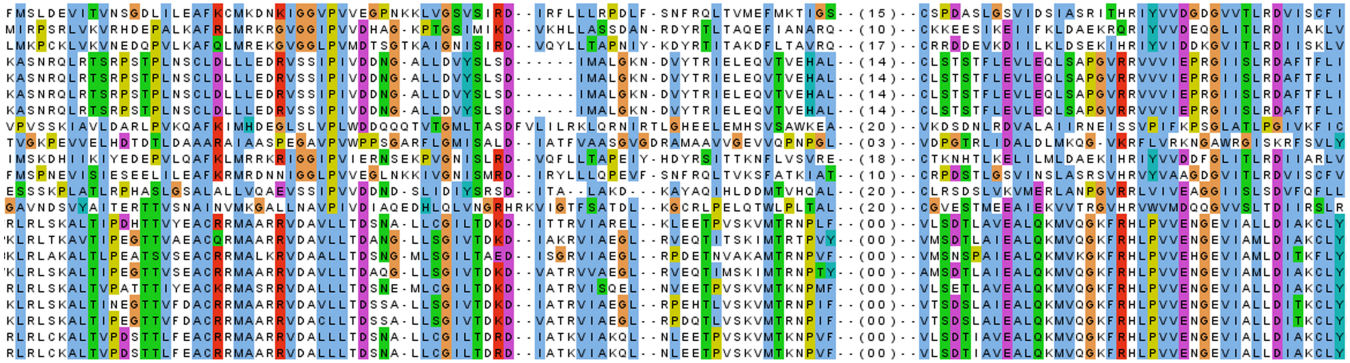
Polymerases



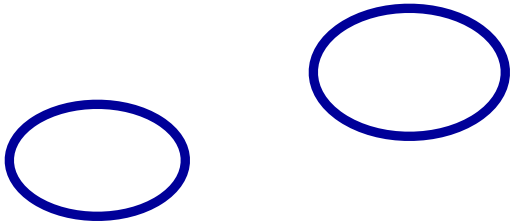
But, for recent events:
fast-evolving genes

Example for a phenetic technique: Neighbor Joining

1) Alignment

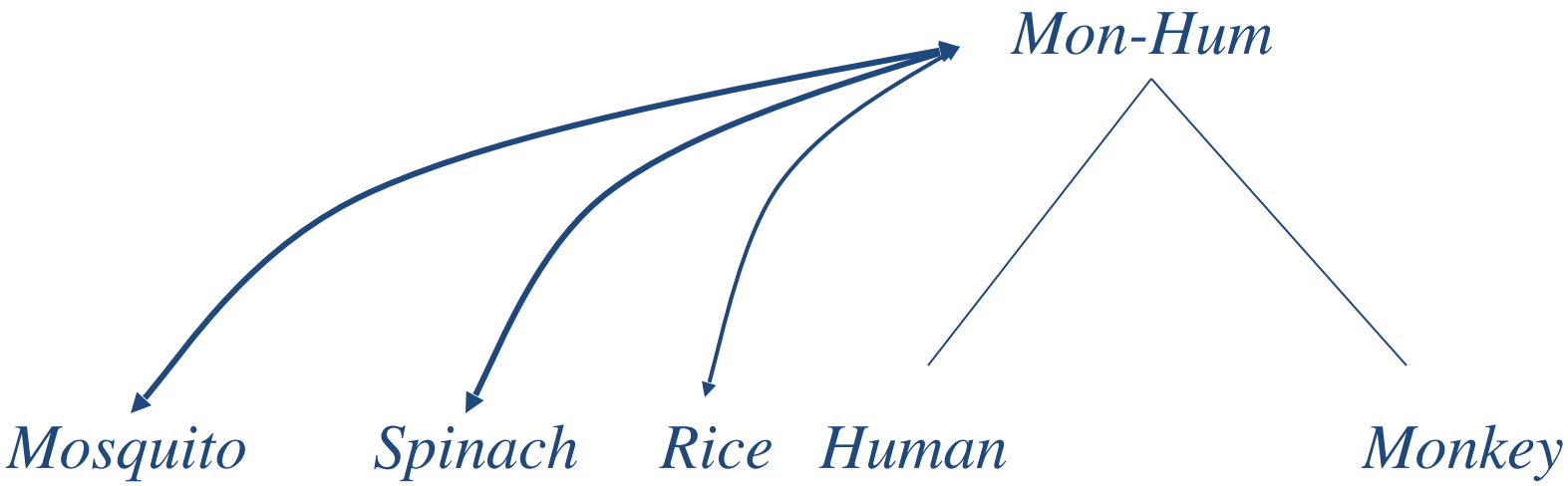


2) Distance matrix



First Step

PAM distance 3.3 (Human - Monkey) is the minimum. So we'll join Human and Monkey to MonHum and we'll calculate the new distances.



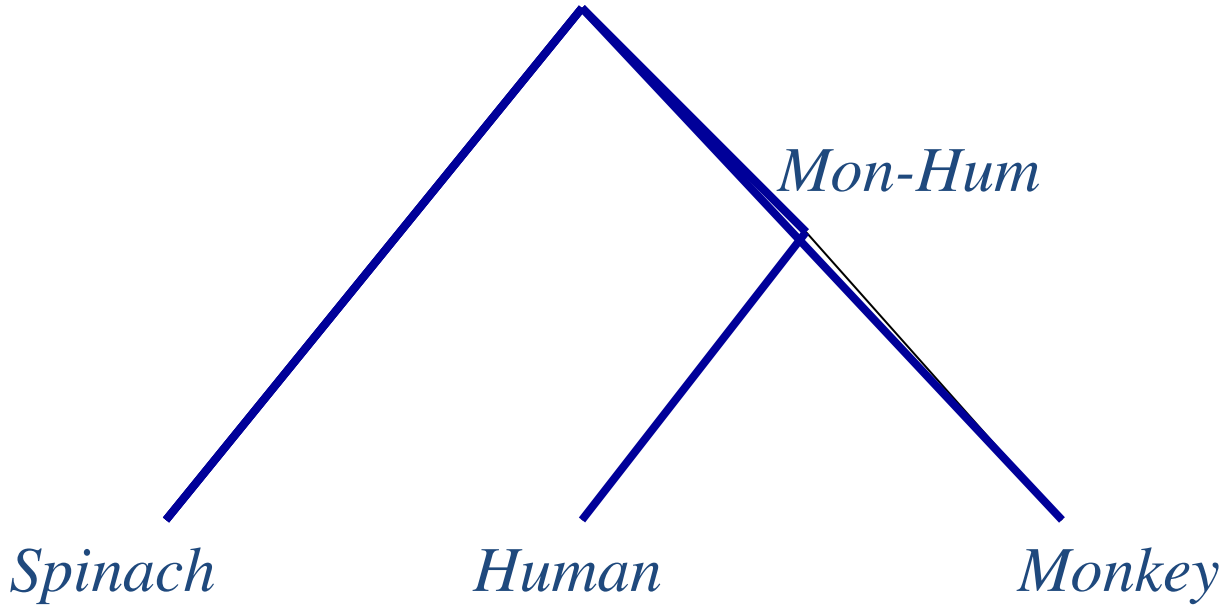
Calculation of new distances

After we have joined two species in a subtree we have to compute the distances from every other node to the new subtree. We do this with a simple average of distances:

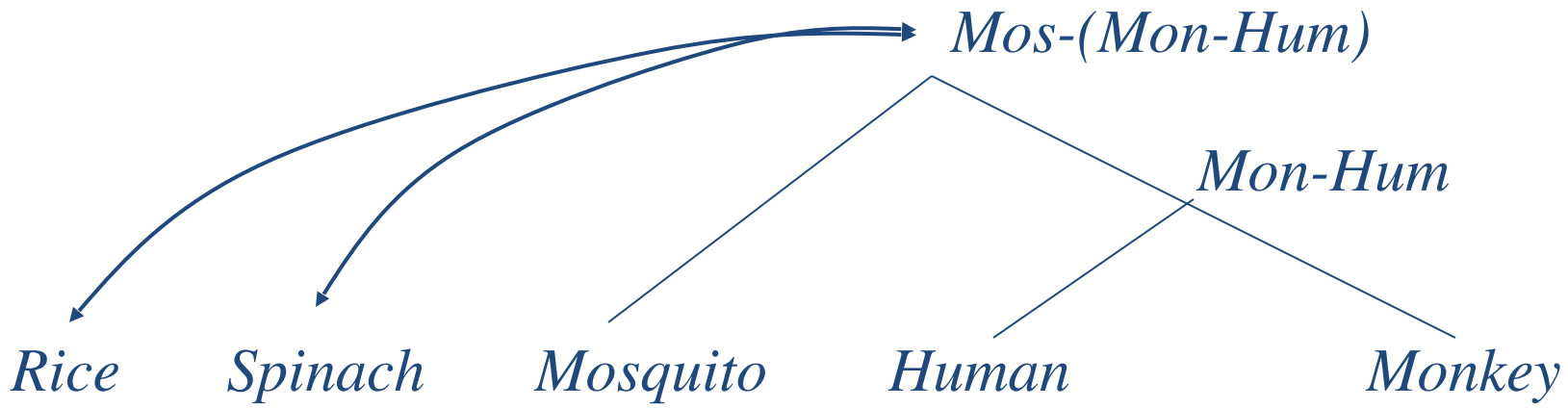
Dist[Spinach, MonHum]

= (Dist[Spinach, Monkey] + Dist[Spinach, Human])/2

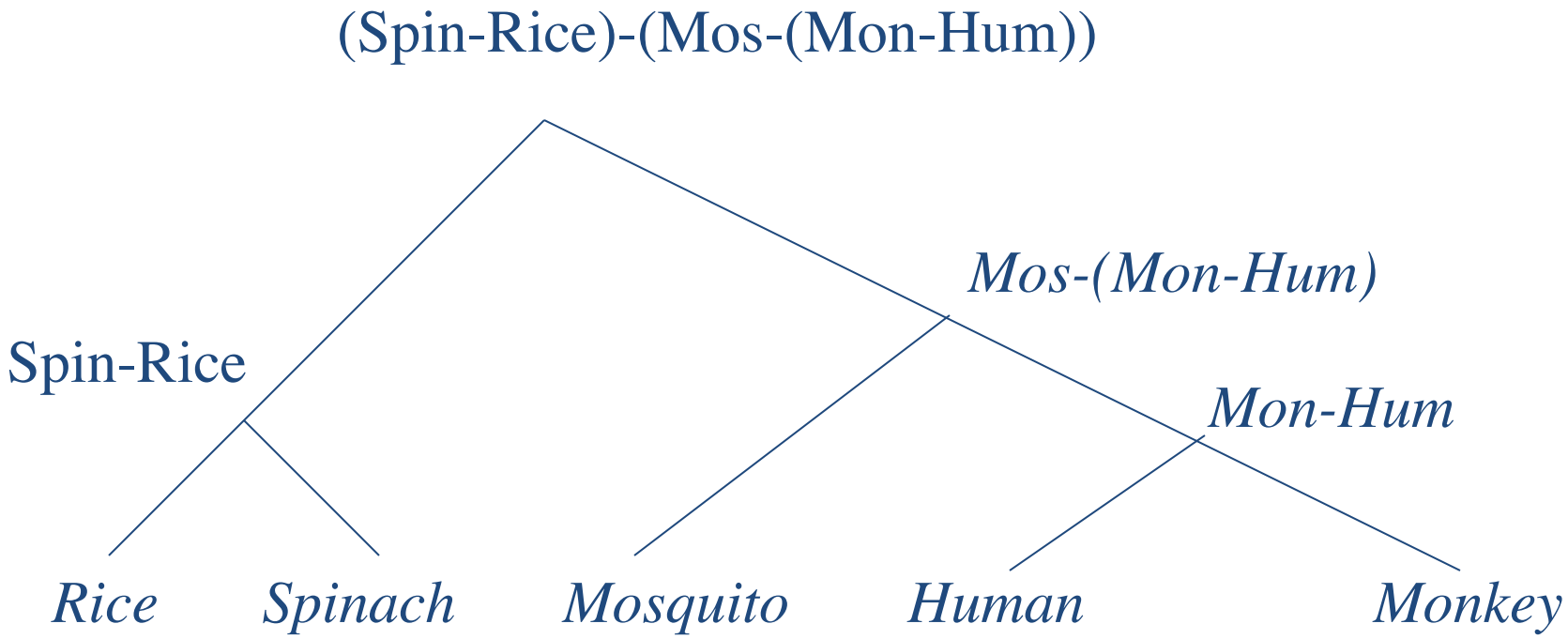
= (90.8 + 86.3)/2 = 88.55



Next cycle



Last joining



Example for a cladistic technique: Maximum Likelihood

- The likelihood is the probability of the data given the model
- The probability of observing the data under the assumed model will change depending on the parameter values of the model.
- The aim of maximum likelihood is to choose the value of the parameter that maximizes the probability of finding the data.

What is an evolutionary model in this context ?

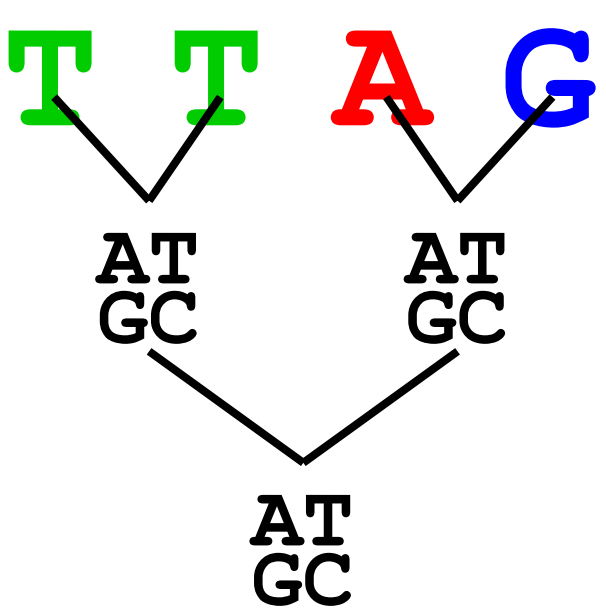
"an empirical matrix describing the relative rates of amino acid replacements"

Dayhoff matrix (Dayhoff et al., 1978)
JTT matrix (Jones et al., 1992)
mtREV matrix (Adachi and Hasegawa, 1996)
WAG matrix (Whelan and Goldman, 2001).

Typically, the model has additional free 'parameters':

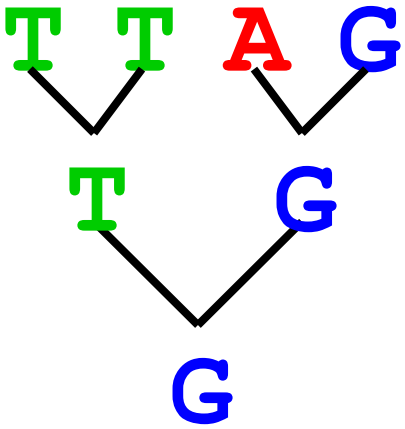
-The rate of evolution can vary across parts of the tree
-The rate of evolution can vary from site to site in the protein

How is maximum likelihood computed ?



1) Image all ancestral possibilities and evolutionary paths.

2) Compute the likelihood of each path



L(path) = L(root) x Π L(branches)

=P(G→T)P(G→G) P(G→A)P(G→G) […]

3) multiply all likelihoods over all possible paths

4) throughout, do not forget to optimize all free parameters

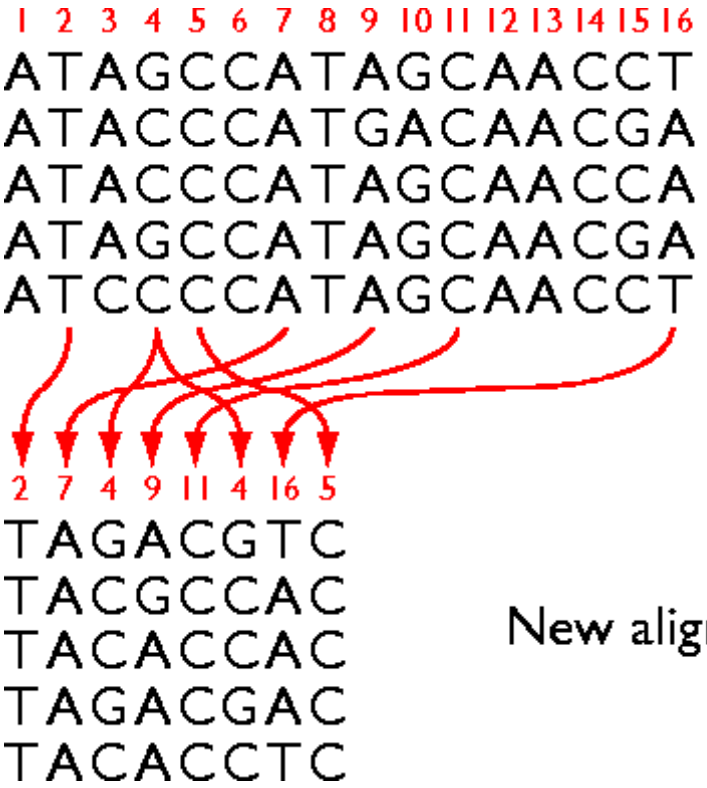
5) Repeat for each tree topology, identify the one with best Likelihood

How do we verify a tree?

Difficult ! Very few trees are actually known with certainty

a) Simulation

b) Bootstrapping



The real multiple alignment

New alignment