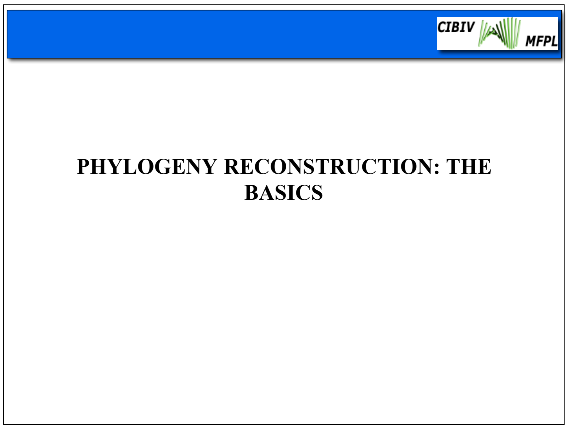
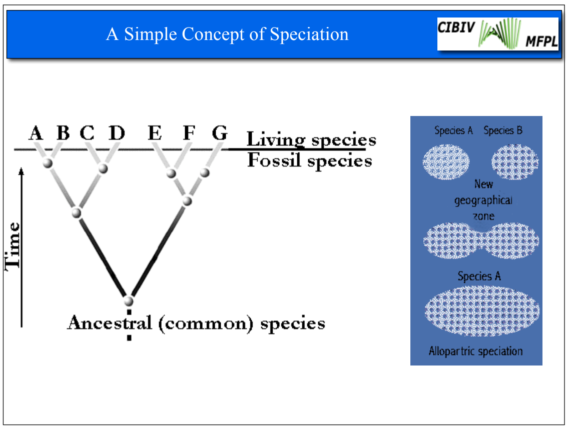
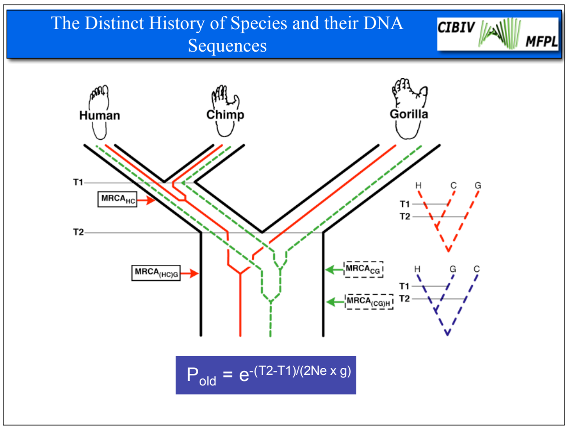


PHYLOGENY RECONSTRUCTION: THE BASICS

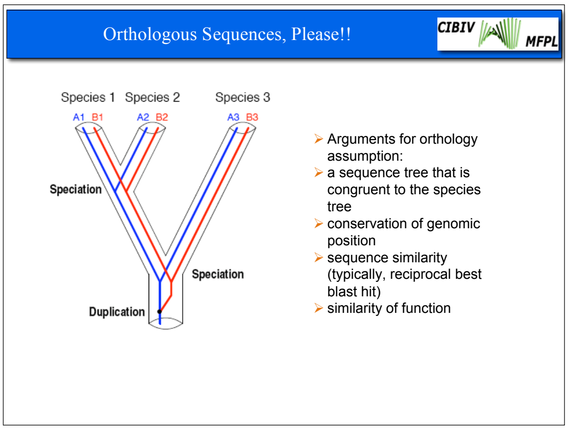


A Simple Concept of Speciation



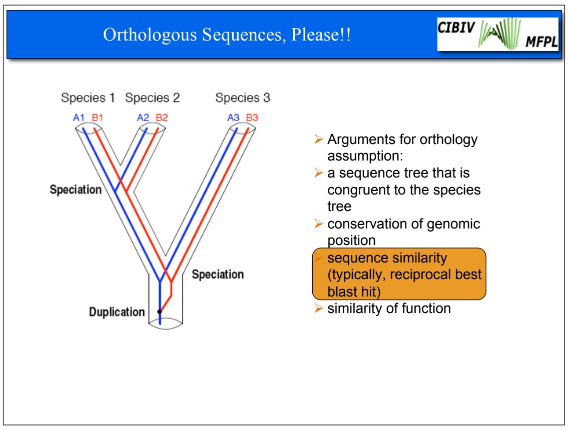
The Distinct History of Species and their DNA Sequences

$$P_{old} = e^{-(T2-T1)/(2Ne \times g)}$$



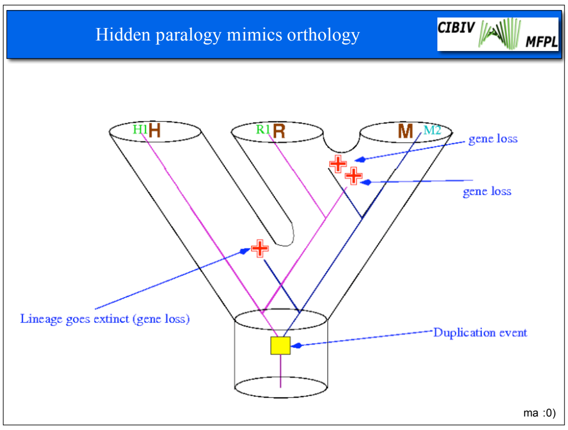
Orthologous Sequences, Please!!

- Arguments for orthology assumption:
- a sequence tree that is congruent to the species tree
- conservation of genomic position
- sequence similarity (typically, reciprocal best blast hit)
- similarity of function



Orthologous Sequences, Please!!

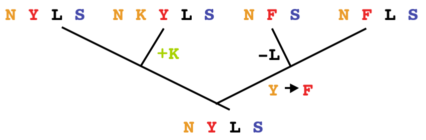
- Arguments for orthology assumption:
- a sequence tree that is congruent to the species tree
- conservation of genomic position
- sequence similarity (typically, reciprocal best blast hit)
- similarity of function



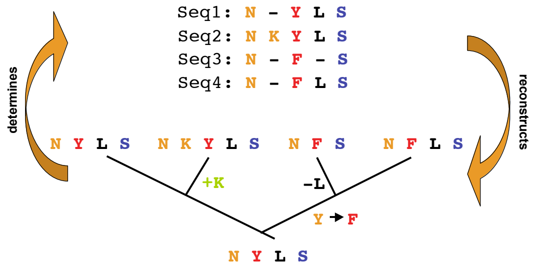
Hidden paralogy mimics orthology

ma :0)

Sequence evolution in a nutshell



Sequence evolution in a nutshell

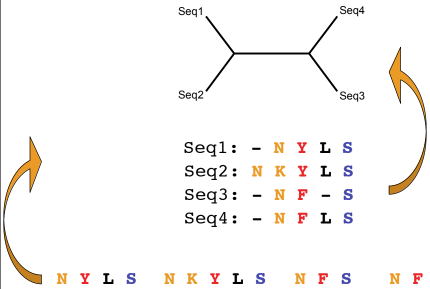


```
Seq1: N - Y L S
Seq2: N K Y L S
Seq3: N - F - S
Seq4: N - F L S
```

The Problem: Finding the homologous positions

N Y L S   N K Y L S   N F S   N F L S

The Problem: Finding the homologous positions



```
Seq1: - N Y L S
Seq2: N K Y L S
Seq3: - N F - S
Seq4: - N F L S
```

The objective function

An mathematical function able to measure the biological quality of an alignment...

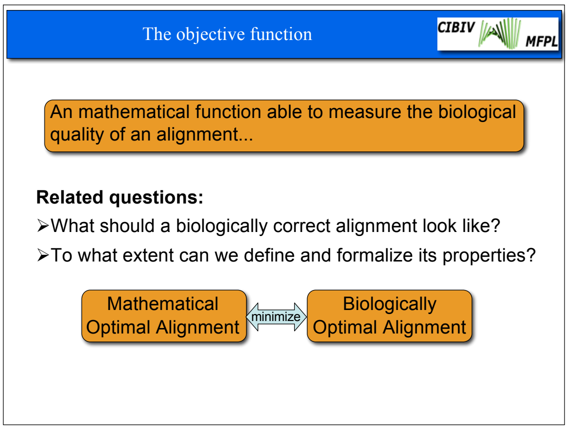
The objective function

An mathematical function able to measure the biological quality of an alignment...

Related questions:

➢What should a biologically correct alignment look like?

➢To what extent can we define and formalize its properties?



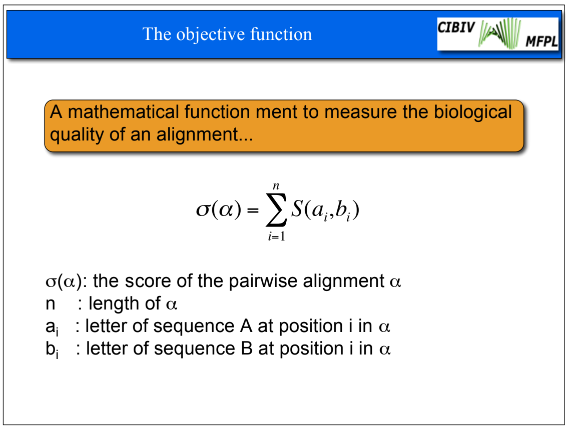
The objective function

An mathematical function able to measure the biological quality of an alignment...

Related questions:

- ➢What should a biologically correct alignment look like?
- ➢To what extent can we define and formalize its properties?

Mathematical Optimal Alignment ⟷ minimize ⟷ Biologically Optimal Alignment

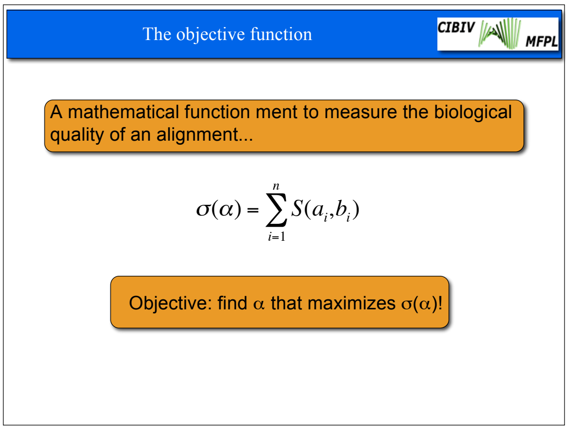


The objective function

A mathematical function ment to measure the biological quality of an alignment...

$$\sigma(\alpha) = \sum_{i=1}^{n} S(a_i, b_i)$$

$\sigma(\alpha)$: the score of the pairwise alignment α
n : length of α
a_i : letter of sequence A at position i in α
b_i : letter of sequence B at position i in α

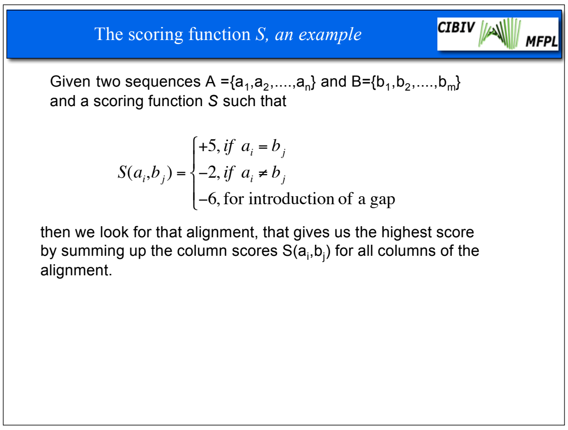


The objective function

A mathematical function ment to measure the biological quality of an alignment...

$$\sigma(\alpha) = \sum_{i=1}^{n} S(a_i, b_i)$$

Objective: find α that maximizes $\sigma(\alpha)$!

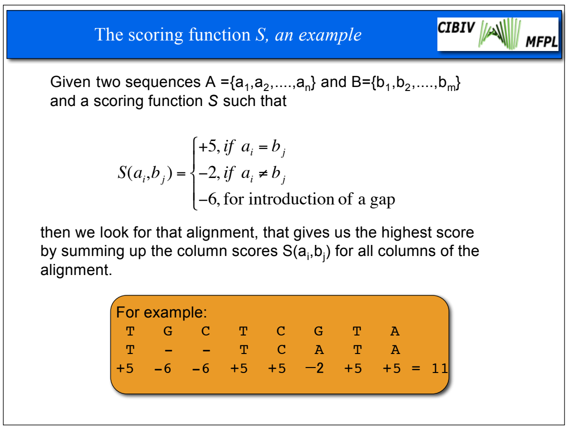


The scoring function *S, an example*

Given two sequences A ={a_1,a_2,....,a_n} and B={b_1,b_2,....,b_m} and a scoring function *S* such that

$$S(a_i,b_j) = \begin{cases} +5, \textit{if } \ a_i = b_j \\ -2, \textit{if } \ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

then we look for that alignment, that gives us the highest score by summing up the column scores S(a_i,b_j) for all columns of the alignment.



The scoring function *S, an example*

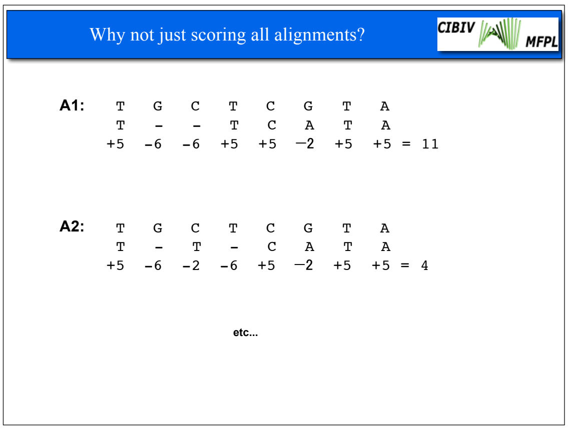
Given two sequences A ={a_1,a_2,....,a_n} and B={b_1,b_2,....,b_m} and a scoring function *S* such that

$$S(a_i,b_j) = \begin{cases} +5, \textit{if } \ a_i = b_j \\ -2, \textit{if } \ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

then we look for that alignment, that gives us the highest score by summing up the column scores S(a_i,b_j) for all columns of the alignment.

For example:

```
 T   G   C   T   C   G   T   A
 T   -   -   T   C   A   T   A
+5  -6  -6  +5  +5  -2  +5  +5 = 11
```



Why not just scoring all alignments?

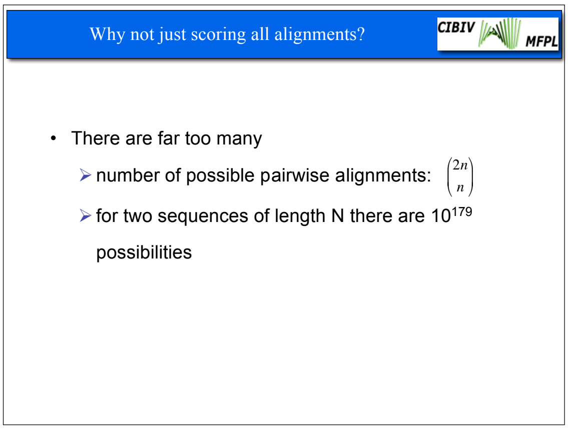
A1:

```
 T   G   C   T   C   G   T   A
 T   -   -   T   C   A   T   A
+5  -6  -6  +5  +5  -2  +5  +5 = 11
```

A2:

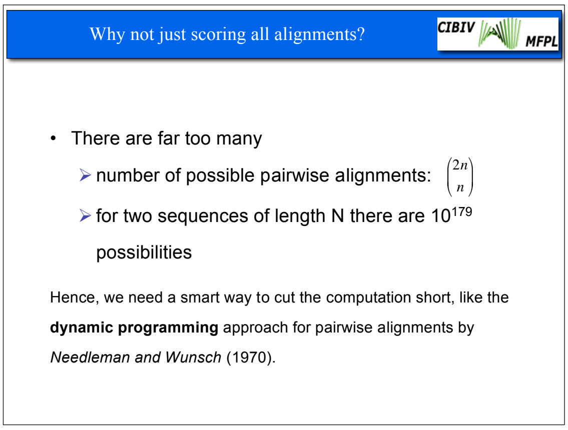
```
 T   G   C   T   C   G   T   A
 T   -   T   -   C   A   T   A
+5  -6  -2  -6  +5  -2  +5  +5 = 4
```

etc...



Why not just scoring all alignments?

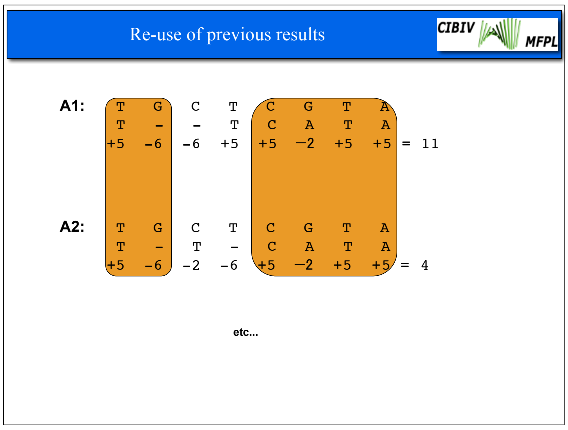
- There are far too many
  - number of possible pairwise alignments: $\binom{2n}{n}$
  - for two sequences of length N there are 10^{179} possibilities



Why not just scoring all alignments?

- There are far too many
  - number of possible pairwise alignments: $\binom{2n}{n}$
  - for two sequences of length N there are 10^{179} possibilities

Hence, we need a smart way to cut the computation short, like the **dynamic programming** approach for pairwise alignments by *Needleman and Wunsch* (1970).

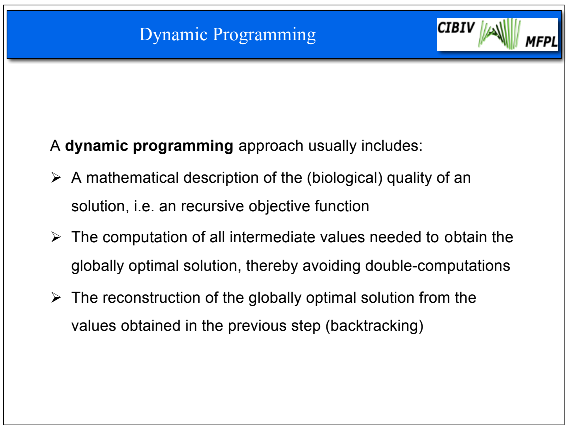


Re-use of previous results

```
A1:  T   G    C   T    C   G   T   A
     T   -    -   T    C   A   T   A
    +5  -6   -6  +5   +5  -2  +5  +5  = 11


A2:  T   G    C   T    C   G   T   A
     T   -    T   -    C   A   T   A
    +5  -6   -2  -6   +5  -2  +5  +5  = 4
```

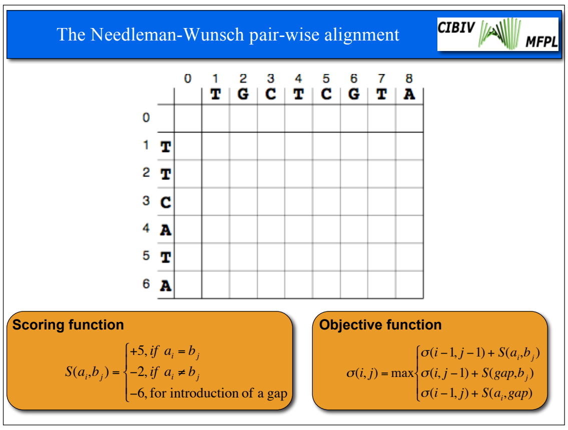
etc...



Dynamic Programming

A **dynamic programming** approach usually includes:

- A mathematical description of the (biological) quality of an solution, i.e. an recursive objective function
- The computation of all intermediate values needed to obtain the globally optimal solution, thereby avoiding double-computations
- The reconstruction of the globally optimal solution from the values obtained in the previous step (backtracking)



The Needleman-Wunsch pair-wise alignment

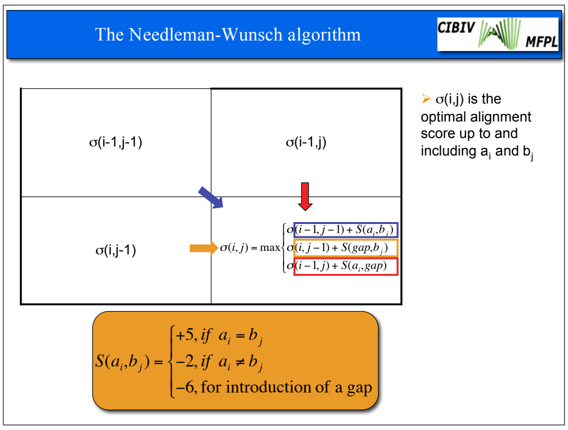
		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0										
1	T									
2	T									
3	C									
4	A									
5	T									
6	A									

Scoring function

$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\-2, if\ \ a_i\neq b_j\\-6, \text{for introduction of a gap}\end{cases}$$

Objective function

$$\sigma(i,j)=\max\begin{cases}\sigma(i-1,j-1)+S(a_i,b_j)\\\sigma(i,j-1)+S(gap,b_j)\\\sigma(i-1,j)+S(a_i,gap)\end{cases}$$

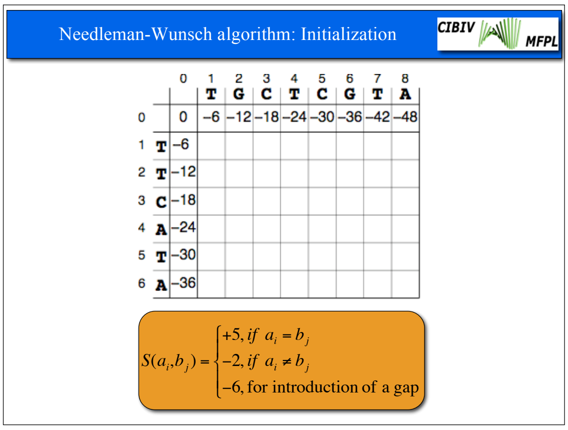


The Needleman-Wunsch algorithm

$$\sigma(i,j)=\max\begin{cases}\boxed{\sigma(i-1,j-1)+S(a_i,b_j)}\\\boxed{\sigma(i,j-1)+S(gap,b_j)}\\\boxed{\sigma(i-1,j)+S(a_i,gap)}\end{cases}$$

- σ(i,j) is the optimal alignment score up to and including a_i and b_j

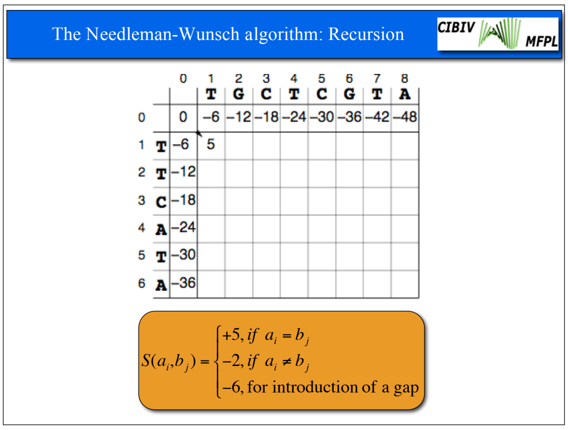
$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\-2, if\ \ a_i\neq b_j\\-6, \text{for introduction of a gap}\end{cases}$$



Needleman-Wunsch algorithm: Initialization

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6								
2	**T**	-12								
3	**C**	-18								
4	**A**	-24								
5	**T**	-30								
6	**A**	-36								

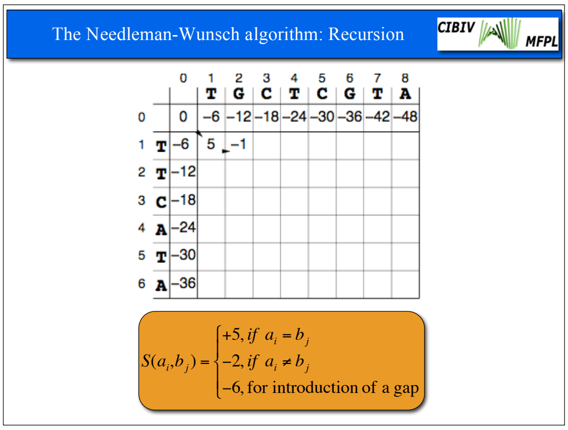
$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$



The Needleman-Wunsch algorithm: Recursion

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6	5							
2	**T**	-12								
3	**C**	-18								
4	**A**	-24								
5	**T**	-30								
6	**A**	-36								

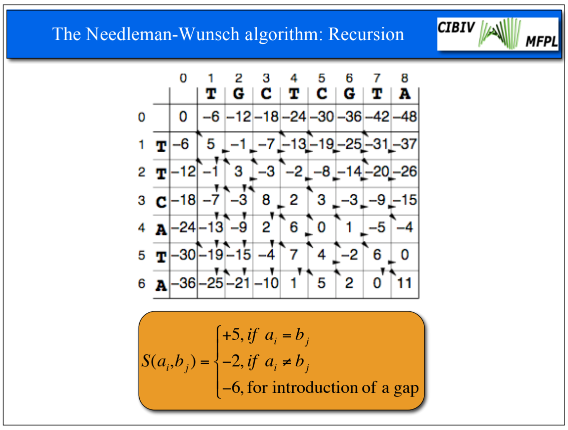
$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$



The Needleman-Wunsch algorithm: Recursion

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6	5	-1						
2	**T**	-12								
3	**C**	-18								
4	**A**	-24								
5	**T**	-30								
6	**A**	-36								

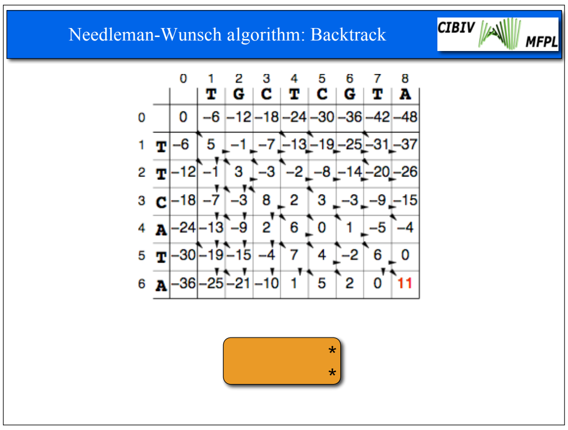
$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$



The Needleman-Wunsch algorithm: Recursion

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6	5	-1	-7	-13	-19	-25	-31	-37
2	**T**	-12	-1	3	-3	-2	-8	-14	-20	-26
3	**C**	-18	-7	-3	8	2	3	-3	-9	-15
4	**A**	-24	-13	-9	2	6	0	1	-5	-4
5	**T**	-30	-19	-15	-4	7	4	-2	6	0
6	**A**	-36	-25	-21	-10	1	5	2	0	11

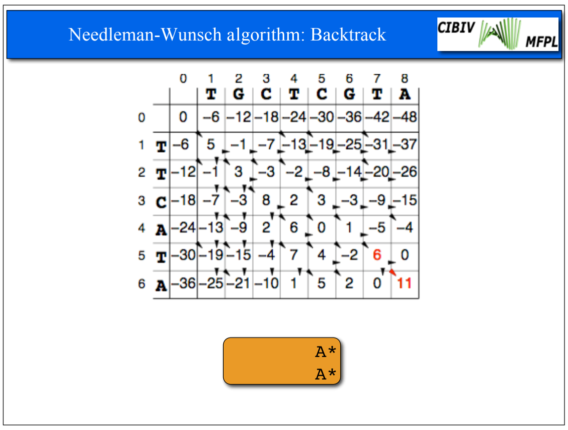
$$S(a_i,b_j)=\begin{cases}+5, if\ \ a_i=b_j\\ -2, if\ \ a_i\neq b_j\\ -6, \text{for introduction of a gap}\end{cases}$$



Needleman-Wunsch algorithm: Backtrack

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6	5	-1	-7	-13	-19	-25	-31	-37
2	**T**	-12	-1	3	-3	-2	-8	-14	-20	-26
3	**C**	-18	-7	-3	8	2	3	-3	-9	-15
4	**A**	-24	-13	-9	2	6	0	1	-5	-4
5	**T**	-30	-19	-15	-4	7	4	-2	6	0
6	**A**	-36	-25	-21	-10	1	5	2	0	11

```
*
*
```

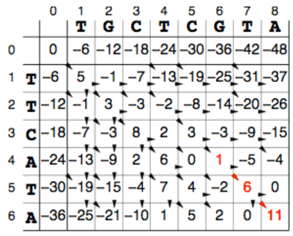


Needleman-Wunsch algorithm: Backtrack

		0	1	2	3	4	5	6	7	8
			T	**G**	**C**	**T**	**C**	**G**	**T**	**A**
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	**T**	-6	5	-1	-7	-13	-19	-25	-31	-37
2	**T**	-12	-1	3	-3	-2	-8	-14	-20	-26
3	**C**	-18	-7	-3	8	2	3	-3	-9	-15
4	**A**	-24	-13	-9	2	6	0	1	-5	-4
5	**T**	-30	-19	-15	-4	7	4	-2	6	0
6	**A**	-36	-25	-21	-10	1	5	2	0	11

```
A*
A*
```

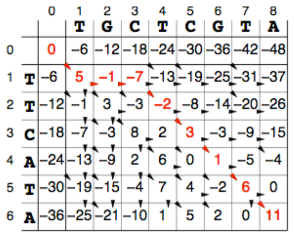
Needleman-Wunsch algorithm: Backtrack



		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
0		0	–6	–12	–18	–24	–30	–36	–42	–48
1	T	–6	5	–1	–7	–13	–19	–25	–31	–37
2	T	–12	–1	3	–3	–2	–8	–14	–20	–26
3	C	–18	–7	–3	8	2	3	–3	–9	–15
4	A	–24	–13	–9	2	6	0	1	–5	–4
5	T	–30	–19	–15	–4	7	4	–2	6	0
6	A	–36	–25	–21	–10	1	5	2	0	11

```
TA*
TA*
```

Needleman-Wunsch algorithm: Backtrack

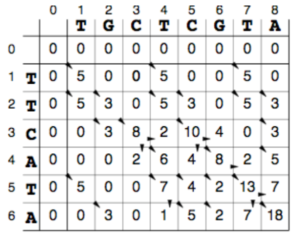


		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
0		0	–6	–12	–18	–24	–30	–36	–42	–48
1	T	–6	5	–1	–7	–13	–19	–25	–31	–37
2	T	–12	–1	3	–3	–2	–8	–14	–20	–26
3	C	–18	–7	–3	8	2	3	–3	–9	–15
4	A	–24	–13	–9	2	6	0	1	–5	–4
5	T	–30	–19	–15	–4	7	4	–2	6	0
6	A	–36	–25	–21	–10	1	5	2	0	11

```
*TGCTCGTA*
*T--TCATA*
```

Alignment Score: 11

Smith-Waterman pairwise local alignment

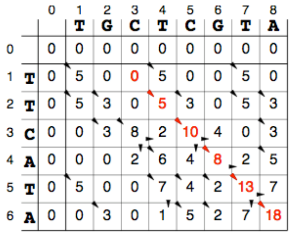


		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
0		0	0	0	0	0	0	0	0	0
1	T	0	5	0	0	5	0	0	5	0
2	T	0	5	3	0	5	3	0	5	3
3	C	0	0	3	8	2	10	4	0	3
4	A	0	0	0	2	6	4	8	2	5
5	T	0	5	0	0	7	4	2	13	7
6	A	0	0	3	0	1	5	2	7	18

$$S(a_i,b_j) = \begin{cases} +5, if\ \ a_i = b_j \\ -2, if\ \ a_i \neq b_j \\ -6, \text{for introduction of a gap} \end{cases}$$

$$\sigma(i,j) = \max \begin{cases} \sigma(i-1,j-1) + S(a_i,b_j) \\ \sigma(i,j-1) + S(gap) \\ \sigma(i-1,j) + S(gap) \\ 0 \end{cases}$$

Smith-Waterman pairwise local alignment



		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
0		0	0	0	0	0	0	0	0	0
1	T	0	5	0	0	5	0	0	5	0
2	T	0	5	3	0	5	3	0	5	3
3	C	0	0	3	8	2	10	4	0	3
4	A	0	0	0	2	6	4	8	2	5
5	T	0	5	0	0	7	4	2	13	7
6	A	0	0	3	0	1	5	2	7	18

```
*TCGTA*
*TCATA*
```

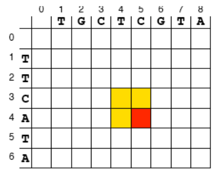
Alignment Score: 18

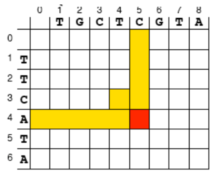
Affine Gap costs

$$g(l) = g_o + l * g_e$$

$$\sigma(i,j) = \max \begin{cases} \sigma(i-1,j-1) + S(a_i,b_j) \\ \sigma(i,j-1) + S(gap,b_j) \\ \sigma(i-1,j) + S(a_i,gap) \end{cases}$$

$$\sigma(i,j) = \max \begin{cases} \sigma(i-1,j-1) + S(a_i,b_j) \\ \max_{k=0}^{i-1}(\sigma(k,j) + g(i-k)), gap\ in\ B \\ \max_{k=0}^{j-1}(\sigma(i,k) + g(j-k)), gap\ in\ A \end{cases}$$





Alternative Scoring Functions

Blosum62:

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W

PAM250:

C Cys	12																			
S Ser	0	2																		
T Thr	-2	1	3																	
P Pro	-3	1	0	6																
A Ala	-2	1	1	1	2															
G Gly	-3	1	0	-1	1	5														
N Asn	-4	1	0	-1	0	0	2													
D Asp	-5	0	0	-1	0	1	2	4												
E Glu	-5	0	0	-1	0	0	1	3	4											
Q Gln	-5	-1	-1	0	0	-1	1	2	2	4										
H His	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F Phe	-4	-3	-3	-5	-5	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W Trp	-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

Many others...

Exact vs. Heuristic searches

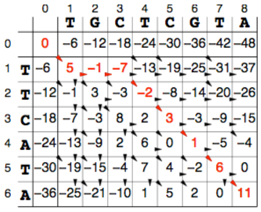
Both, Needleman-Wunsch and Smith-Waterman alignment methods are **exact** methods since they guarantee a globally optimal solution for the optimization problem!

Drawback: Computational expensive, i.e. O(nm) in time and memory

Exact vs. Heuristic searches

Solutions:

- ➢ omit regions from the grid, that cannot contribute to the optimal alignment (reduction of the search space, by remaining exact)



		0	1	2	3	4	5	6	7	8
			T	G	C	T	C	G	T	A
0		0	−6	−12	−18	−24	−30	−36	−42	−48
1	T	−6	5	−1	−7	−13	−19	−25	−31	−37
2	T	−12	−1	3	−3	−2	−8	−14	−20	−26
3	C	−18	−7	−3	8	2	3	−3	−9	−15
4	A	−24	−13	−9	2	6	0	1	−5	−4
5	T	−30	−19	−15	−4	7	4	−2	6	0
6	A	−36	−25	−21	−10	1	5	2	0	11

Exact vs. Heuristic searches

Solutions:

- ➢ use of heuristics (more rigorous reduction of the search space, sacrificing the guaranteed optimal solution for search speed)

Hashing

- Lookup method for finding an alignment

```
Pos:    1  2  3  4  5  6  7  8  9 10 11
Seq 1: k  c  s  p  t  a  .  .  .  .  .
Seq 2: .  .  .  .  .  a  c  s  p  r  k
```

Amino acid	Pos in Seq 1	Pos in Seq 2	Offset
k	1	11	10
c	2	7	-5
s	3	8	-5
p	4	9	-5
t	5	-	-
a	6	6	0
r	-	10	-

Hashing

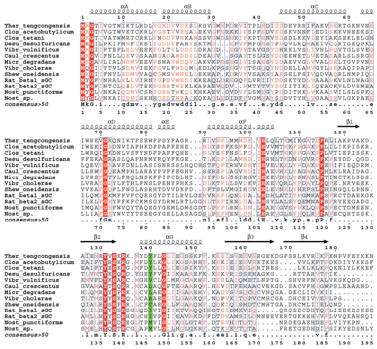
- Lookup method for finding an alignment

```
Pos:    1  2  3  4  5  6  7  8  9 10 11
Seq 1: k  c  s  p  t  a  .  .  .  .  .
Seq 2: .  .  .  .  .  a  c  s  p  r  k
```

Amino acid	Pos in Seq 1	Pos in Seq 2	Offset
k	1	11	10
c	2	7	-5
s	3	8	-5
p	4	9	-5
t	5	-	-
a	6	6	0
r	-	10	-

```
Resulting alignment: Seq 1: k  c  s  p  t  a
                     Seq 2: a  c  s  p  r  k
```

What we are really looking for:

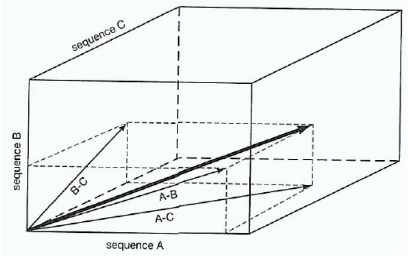


How to construct Multiple Sequence Alignments?

Optimal Solution:
Extend Needleman-Wunsch or Smith-Waterman to multiple sequences

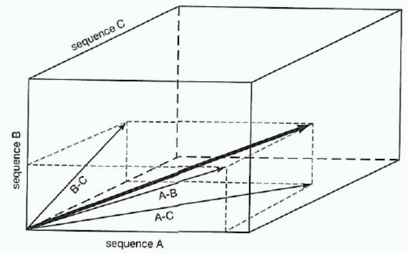
How to construct Multiple Sequence Alignments?

Optimal Solution:
Extend Needleman-Wunsch or Smith-Waterman to multiple sequences



How to construct Multiple Sequence Alignments?

Optimal Solution:
Extend Needleman-Wunsch or Smith-Waterman to multiple sequences



But $O(n^m)$ in time and memory:
Computationally not feasible... 4 sequences of length 1000 -> 1TB RAM

A new objective function: Sum of Pairs

```
Seq1: AGA--CTA
Seq2: G-A--CTT
Seq3: AGAAACTT
```

A new objective function: Sum of Pairs

```
Seq1: AGA--CTA
Seq2: G-A--CTT
Seq3: AGAAACTT
```

```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
```

$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\-2, if\ a_i\neq b_j\\-6, \text{for introduction of a gap}\end{cases}$$

```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
```

Score: +5 **Score: +11** **Score: 0**

A new objective function: Sum of Pairs

```
Seq1: AGA--CTA
Seq2: G-A--CTT
Seq3: AGAAACTT
```

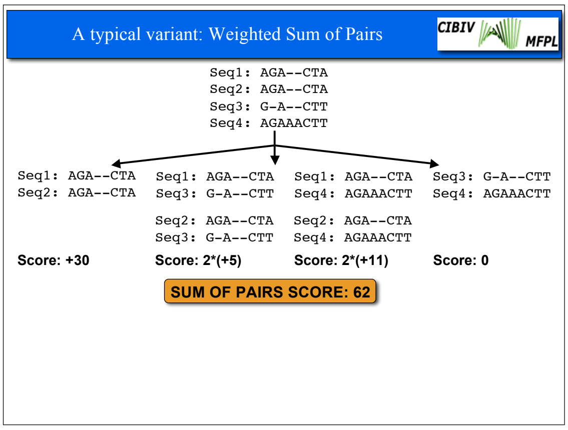
```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
```

$$S(a_i,b_j)=\begin{cases}+5, if\ a_i=b_j\\-2, if\ a_i\neq b_j\\-6, \text{for introduction of a gap}\end{cases}$$

```
Seq1: AGA--CTA      Seq1: AGA--CTA      Seq2: G-A--CTT
Seq2: G-A--CTT      Seq3: AGAAACTT      Seq3: AGAAACTT
```

Score: +5 **Score: +11** **Score: 0**

SUM OF PAIRS SCORE: 16



A typical variant: Weighted Sum of Pairs

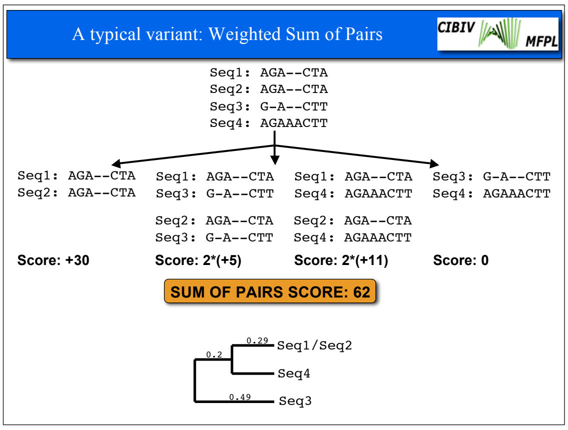
```
Seq1: AGA--CTA
Seq2: AGA--CTA
Seq3: G-A--CTT
Seq4: AGAAACTT
```

```
Seq1: AGA--CTA   Seq1: AGA--CTA   Seq1: AGA--CTA   Seq3: G-A--CTT
Seq2: AGA--CTA   Seq3: G-A--CTT   Seq4: AGAAACTT   Seq4: AGAAACTT

                 Seq2: AGA--CTA   Seq2: AGA--CTA
                 Seq3: G-A--CTT   Seq4: AGAAACTT
```

Score: +30 **Score: 2*(+5)** **Score: 2*(+11)** **Score: 0**

SUM OF PAIRS SCORE: 62



A typical variant: Weighted Sum of Pairs

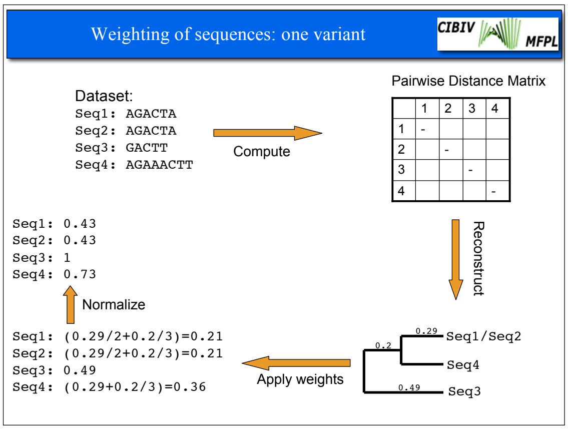
```
Seq1: AGA--CTA
Seq2: AGA--CTA
Seq3: G-A--CTT
Seq4: AGAAACTT
```

```
Seq1: AGA--CTA   Seq1: AGA--CTA   Seq1: AGA--CTA   Seq3: G-A--CTT
Seq2: AGA--CTA   Seq3: G-A--CTT   Seq4: AGAAACTT   Seq4: AGAAACTT

                 Seq2: AGA--CTA   Seq2: AGA--CTA
                 Seq3: G-A--CTT   Seq4: AGAAACTT
```

Score: +30 **Score: 2*(+5)** **Score: 2*(+11)** **Score: 0**

SUM OF PAIRS SCORE: 62



Weighting of sequences: one variant

Dataset:

```
Seq1: AGACTA
Seq2: AGACTA
Seq3: GACTT
Seq4: AGAAACTT
```

Compute

Pairwise Distance Matrix

	1	2	3	4
1	-			
2		-		
3			-	
4				-

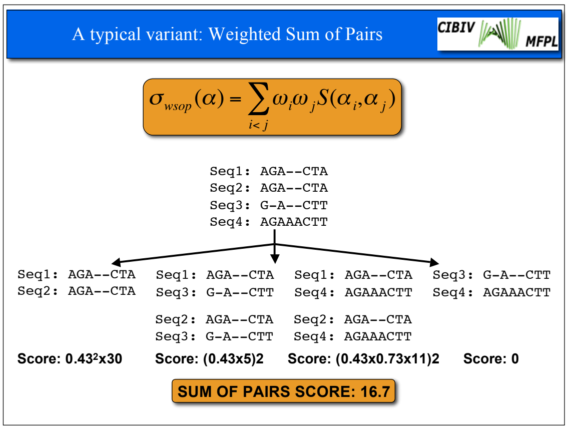
Reconstruct

Apply weights

```
Seq1: (0.29/2+0.2/3)=0.21
Seq2: (0.29/2+0.2/3)=0.21
Seq3: 0.49
Seq4: (0.29+0.2/3)=0.36
```

Normalize

```
Seq1: 0.43
Seq2: 0.43
Seq3: 1
Seq4: 0.73
```



A typical variant: Weighted Sum of Pairs

$$\sigma_{wsop}(\alpha) = \sum_{i<j} \omega_i \omega_j S(\alpha_i, \alpha_j)$$

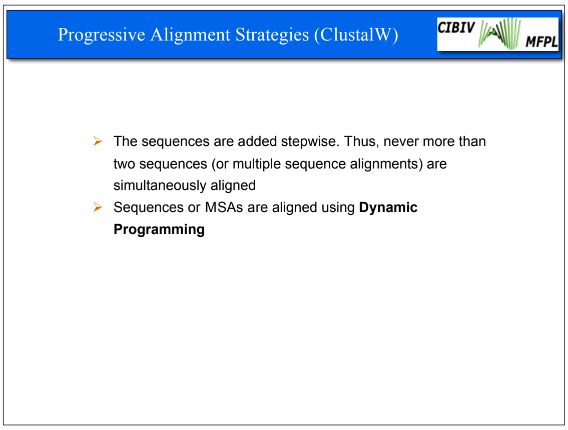
```
Seq1: AGA--CTA
Seq2: AGA--CTA
Seq3: G-A--CTT
Seq4: AGAAACTT
```

```
Seq1: AGA--CTA   Seq1: AGA--CTA   Seq1: AGA--CTA   Seq3: G-A--CTT
Seq2: AGA--CTA   Seq3: G-A--CTT   Seq4: AGAAACTT   Seq4: AGAAACTT

                 Seq2: AGA--CTA   Seq2: AGA--CTA
                 Seq3: G-A--CTT   Seq4: AGAAACTT
```

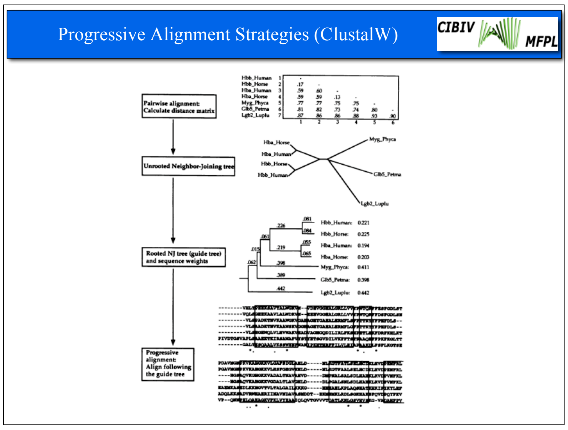
Score: 0.43^2x30 **Score: (0.43x5)2** **Score: (0.43x0.73x11)2** **Score: 0**

SUM OF PAIRS SCORE: 16.7



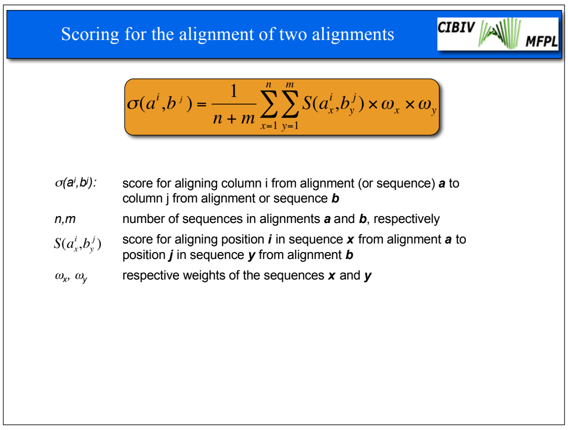
Progressive Alignment Strategies (ClustalW)

- The sequences are added stepwise. Thus, never more than two sequences (or multiple sequence alignments) are simultaneously aligned
- Sequences or MSAs are aligned using **Dynamic Programming**



Progressive Alignment Strategies (ClustalW)

		1	2	3	4	5	6
Hbb_Human	1	-					
Hbb_Horse	2	.17	-				
Hba_Human	3	.59	.60	-			
Hba_Horse	4	.59	.59	.13	-		
Myg_Phyca	5	.77	.77	.75	.75	-	
Glb5_Petma	6	.81	.82	.73	.74	.80	-
Lgb2_Luplu	7	.87	.86	.86	.88	.93	.90



Scoring for the alignment of two alignments

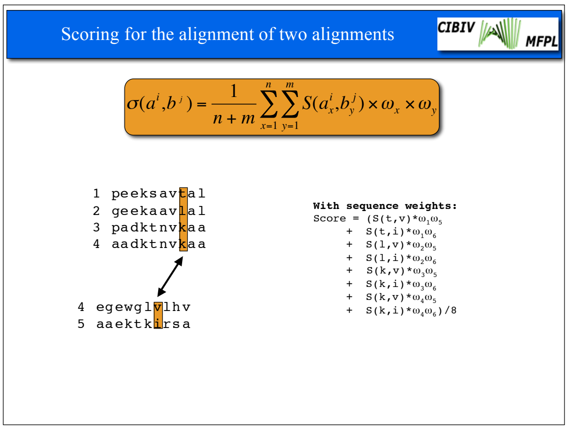
$$\sigma(a^i,b^j) = \frac{1}{n+m}\sum_{x=1}^{n}\sum_{y=1}^{m} S(a_x^i,b_y^j)\times\omega_x\times\omega_y$$

$\sigma(a^i,b^j)$: score for aligning column i from alignment (or sequence) ***a*** to column j from alignment or sequence ***b***

n,m number of sequences in alignments ***a*** and ***b***, respectively

$S(a_x^i,b_y^j)$ score for aligning position ***i*** in sequence ***x*** from alignment ***a*** to position ***j*** in sequence ***y*** from alignment ***b***

ω_x, ω_y respective weights of the sequences ***x*** and ***y***



Scoring for the alignment of two alignments

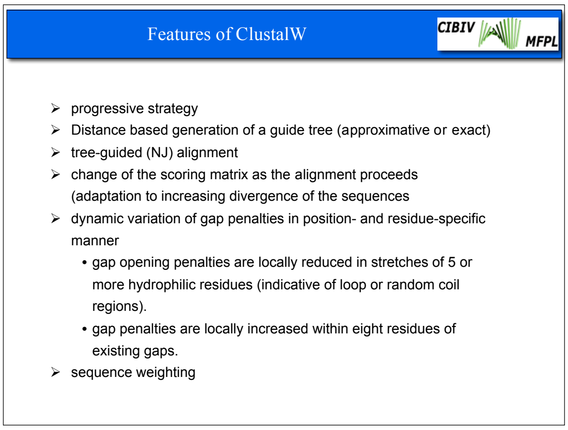
$$\sigma(a^i,b^j) = \frac{1}{n+m}\sum_{x=1}^{n}\sum_{y=1}^{m} S(a_x^i,b_y^j)\times\omega_x\times\omega_y$$

```
  1 peeksavtal
  2 geekaavlal
  3 padktnvkaa
  4 aadktnvkaa

4 egewglvlhv
5 aaektkirsa
```

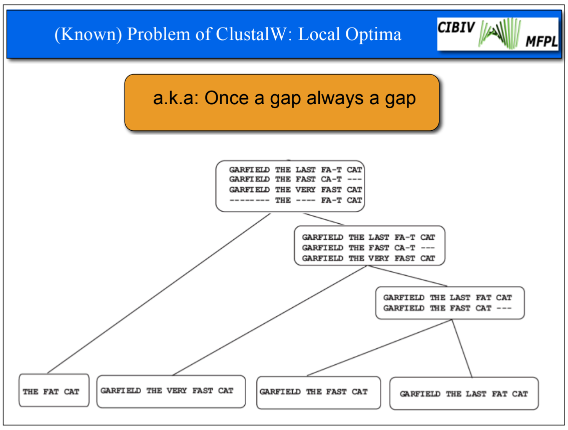
With sequence weights:

$$\begin{aligned}\text{Score} = (&S(t,v)*\omega_1\omega_5\\ +\ &S(t,i)*\omega_1\omega_6\\ +\ &S(l,v)*\omega_2\omega_5\\ +\ &S(l,i)*\omega_2\omega_6\\ +\ &S(k,v)*\omega_3\omega_5\\ +\ &S(k,i)*\omega_3\omega_6\\ +\ &S(k,v)*\omega_4\omega_5\\ +\ &S(k,i)*\omega_4\omega_6)/8\end{aligned}$$



Features of ClustalW

- progressive strategy
- Distance based generation of a guide tree (approximative or exact)
- tree-guided (NJ) alignment
- change of the scoring matrix as the alignment proceeds (adaptation to increasing divergence of the sequences
- dynamic variation of gap penalties in position- and residue-specific manner
  - gap opening penalties are locally reduced in stretches of 5 or more hydrophilic residues (indicative of loop or random coil regions).
  - gap penalties are locally increased within eight residues of existing gaps.
- sequence weighting



(Known) Problem of ClustalW: Local Optima

a.k.a: Once a gap always a gap

```
GARFIELD THE LAST FA-T CAT
GARFIELD THE FAST CA-T ---
GARFIELD THE VERY FAST CAT
-------- THE ---- FA-T CAT
```

```
GARFIELD THE LAST FA-T CAT
GARFIELD THE FAST CA-T ---
GARFIELD THE VERY FAST CAT
```

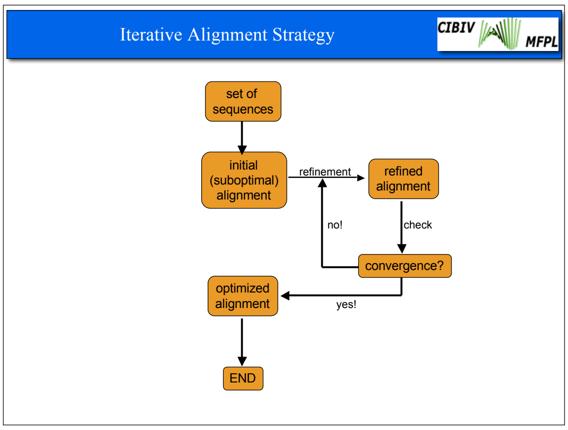
```
GARFIELD THE LAST FAT CAT
GARFIELD THE FAST CAT ---
```

```
THE FAT CAT
```

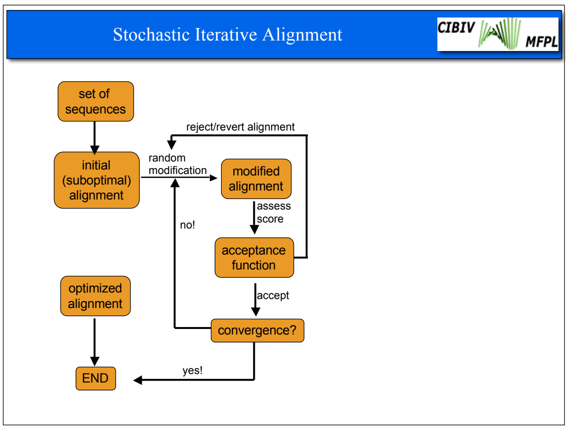
```
GARFIELD THE VERY FAST CAT
```

```
GARFIELD THE FAST CAT
```

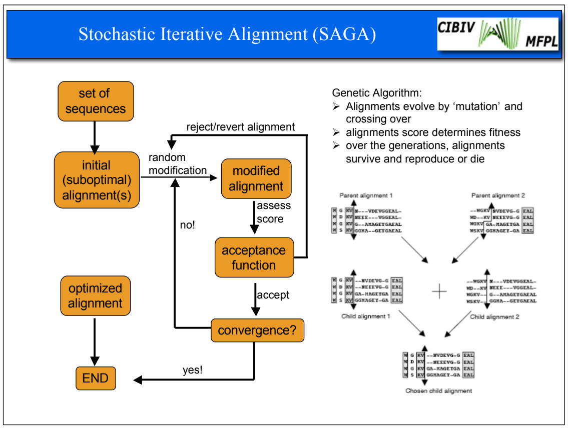
```
GARFIELD THE LAST FAT CAT
```



Iterative Alignment Strategy

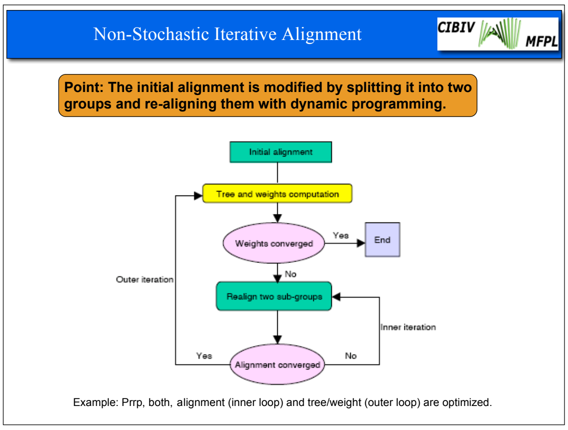


Stochastic Iterative Alignment



Stochastic Iterative Alignment (SAGA)

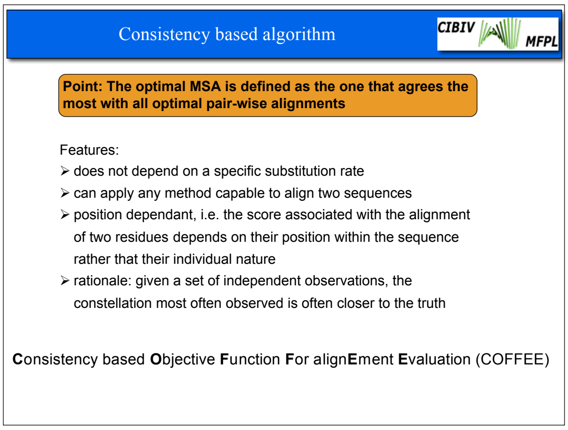
Genetic Algorithm:
- Alignments evolve by 'mutation' and crossing over
- alignments score determines fitness
- over the generations, alignments survive and reproduce or die



Non-Stochastic Iterative Alignment

Point: The initial alignment is modified by splitting it into two groups and re-aligning them with dynamic programming.

Example: Prrp, both, alignment (inner loop) and tree/weight (outer loop) are optimized.



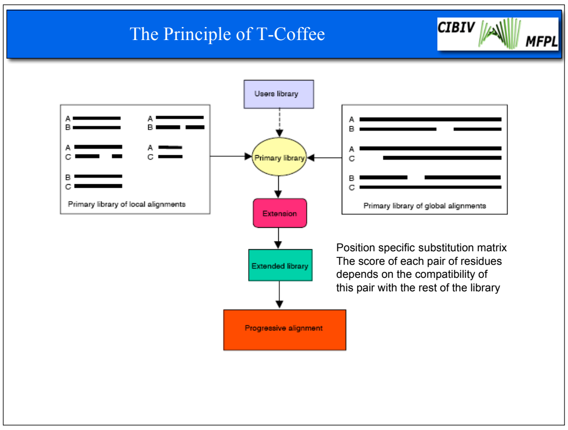
Consistency based algorithm

Point: The optimal MSA is defined as the one that agrees the most with all optimal pair-wise alignments

Features:

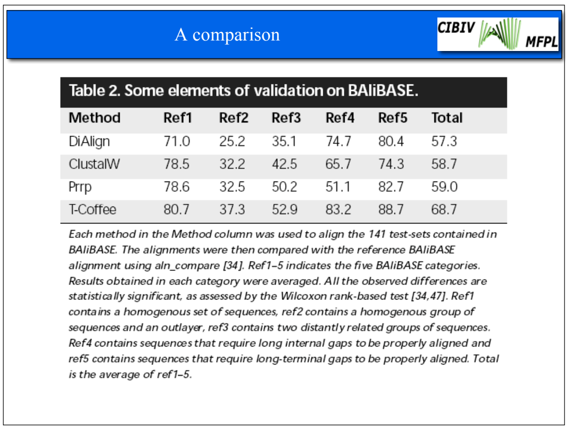
- does not depend on a specific substitution rate
- can apply any method capable to align two sequences
- position dependant, i.e. the score associated with the alignment of two residues depends on their position within the sequence rather that their individual nature
- rationale: given a set of independent observations, the constellation most often observed is often closer to the truth

Consistency based **O**bjective **F**unction **F**or align**E**ment **E**valuation (COFFEE)



The Principle of T-Coffee

Position specific substitution matrix
The score of each pair of residues depends on the compatibility of this pair with the rest of the library

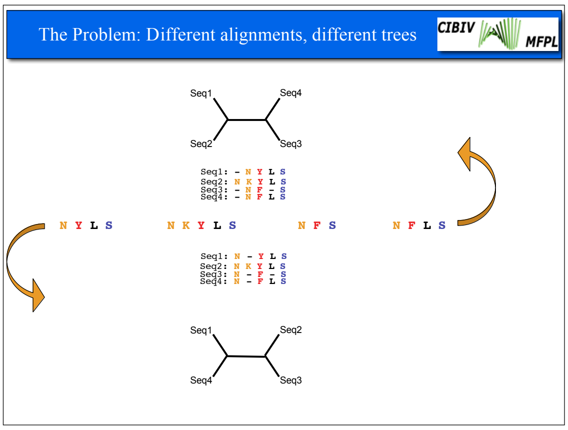


A comparison

Table 2. Some elements of validation on BAliBASE.

Method	Ref1	Ref2	Ref3	Ref4	Ref5	Total
DiAlign	71.0	25.2	35.1	74.7	80.4	57.3
ClustalW	78.5	32.2	42.5	65.7	74.3	58.7
Prrp	78.6	32.5	50.2	51.1	82.7	59.0
T-Coffee	80.7	37.3	52.9	83.2	88.7	68.7

Each method in the Method column was used to align the 141 test-sets contained in BAliBASE. The alignments were then compared with the reference BAliBASE alignment using aln_compare [34]. Ref1–5 indicates the five BAliBASE categories. Results obtained in each category were averaged. All the observed differences are statistically significant, as assessed by the Wilcoxon rank-based test [34,47]. Ref1 contains a homogenous set of sequences, ref2 contains a homogenous group of sequences and an outlayer, ref3 contains two distantly related groups of sequences. Ref4 contains sequences that require long internal gaps to be properly aligned and ref5 contains sequences that require long-terminal gaps to be properly aligned. Total is the average of ref1–5.

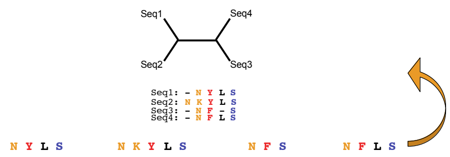


The Problem: Different alignments, different trees

```
Seq1: - N Y L S
Seq2: N K Y L S
Seq3: - N F - S
Seq4: - N F L S
```

```
Seq1: N - Y L S
Seq2: N K Y L S
Seq3: N - F - S
Seq4: N - F L S
```

The Problem: Different alignments, different trees



```
Seq1: - N Y L S
Seq2: N K Y L S
Seq3: - N F - S
Seq4: - N F L S
```

```
N Y L S        N K Y L S        N F S        N F L S
```

The alignment strategy may have more impact on the reconstructed tree than does the type of tree building method.
Morrison and Ellis (1997) Mol. Biol. Evol. 14:428-441

Focussing on stable parts of the alignment

Gblocks (Castresana (2000) Mol. Biol. Evol. 17:540-552
Objective:
Define a set of conserved blocks from an alignment to be used in phylogeny reconstruction

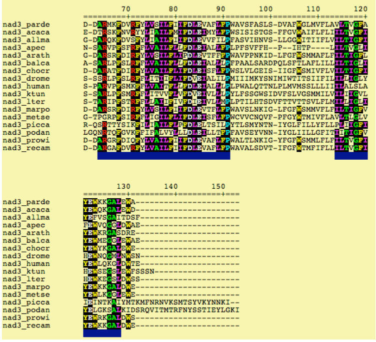
Approach:

1) Classification of Columns
   - ➢ non-conserved : <n/2 + 1 identical residues, or a gap
   - ➢ conserved : ≥n/2 + 1 and < 85% identical residues
   - ➢ highly conserved :>85% identical residues

2) discard contiguous stretches of non-conserved positions (default l = 8)
3) from remaining blocks: remove flanking positions until blocks begin and end with highly conserved positions, i.e. selected blocks are anchored by positions that can be aligned with high confidence
4) discard blocks with l < 15
5) remove all positions with gaps together with adjacent positions until a conserved position is reached
6) discard blocks with l < 10

Note: all given values are the program defaults as given in the original publication

Focussing on stable parts of the alignment



```
                    70        80        90       100       110       120
             =========+=========+=========+=========+=========+=========+
nad3_parde   D-DARMKFDVRFYLVSILFIIFDLEVAFLFPWAVSFASLS-DVAFWGLMVFLAVLTVGFA
nad3_acaca   E-DTRSKFNVRYYLIAILFMIFDLEIMYLFPWSISISTGS-FFGVWAIFLFLIILTVGFI
nad3_allma   G-DARQKFDVSFYLIAILFIIFDLEVVFILPFASVIHNVS-LLGGWITIIFLVILTIGFI
nad3_apec    N-SARVPFSFRFFLVAILFLLFDLEIALLFPLPFSVFFH--P--IHTP----LILTVGLI
nad3_arath   G-DARSRFDIRFYLVSILFLIPDLEVTFFFPWAVPPNKID-LFGFWSMMAFLFILTIGFL
nad3_balca   A-SARLPFSLRFFLVAILFLLFDLEIALLLPFPAALSARDPQLSFTLAFLILLILTIGLI
nad3_chocr   D-DARATFDIRFYLVAILFLIFDLEISFLFPWSLVLGEIS-IIGFWSMIVFLVILTIGFI
nad3_drome   S-SSRLPFSLRFFLITIIFLIFDVEIALILPMIIIMKYSNIMIWTITSIIFILILLIGLY
nad3_human   S-PARVPFSMKFFLVAITFLLFDLEIALLLPLPWALQTTNLPLMVMSSLLLIIILALSLA
nad3_ktun    S-SARLPFSMRFFLITVVFLVFDVEIVLLLPYLFSSGWSIDVFSLVGSMMILVILIIGVL
nad3_lter    S-TARIPFSTRFFLLAIIFIVFDIEIVLLMPLPTILHTSDVFTTVTTSVLFLMILLIGLI
nad3_marpo   D-DARSRFDIRFYLVSILFIIFDLEVTFLFPWAVSLNKIG-LFGFWSMMVFLFILTIGFV
nad3_metse   G-TPGRPFSIRFFLIGILFLIFDLEISFLFPWCVVCNQVF-PFGYWTMIVFLAVLTLGLV
nad3_picca   R-QSRTTYSIKFILIAILFLPFDLELTSILPYTLSMYNTN-IYGLFILLYFLLPLIIGFI
nad3_podan   LGQNRTQFGVKFFIFALVYLLLDLEILLTFPFAVSEYVNN-IYGLIILLGFITIITIGFV
nad3_prowi   D-DARGRFDIQFYLVAILFIIFDLEVAFLFPWALTLNKIG-YFGFWSMMLFLFILTVGFI
nad3_recam   D-DARGAFDVRFYLVAILFIIFDLEVAFLFPWAVALSDVT-IFGFWTMFIFLLILTVGFI

                   130       140       150
             =========+=========+=========+===
nad3_parde   YEWKKGALEWA----------------------
nad3_acaca   YEWQKGALEWD----------------------
nad3_allma   YEFVSGAITDSF---------------------
nad3_apec    FEWVQGGLDWAE---------------------
nad3_arath   YEWKRGASDRE----------------------
nad3_balca   YEWMEGGLEWAE---------------------
nad3_chocr   YEWYKGALEWE----------------------
nad3_drome   HEWNQGMLNWSN---------------------
nad3_human   YEWLQKGLDWTE---------------------
nad3_ktun    HEWSEGSLEWFSSSN------------------
nad3_lter    HEWKEGSLDWSS---------------------
nad3_marpo   YEWKKGALDWE----------------------
nad3_metse   YEWLKGGLEWE----------------------
nad3_picca   IEINTKAIYMTKMFNRNVKSMTSYVKYNNKI--
nad3_podan   YELGKSALKIDSRQVITMTRFNYSSTIEYLGKI
nad3_prowi   YEWRKGALDWS----------------------
nad3_recam   YEWKKGALDWE----------------------
```