Alignments

Robert Kofler

Alignment

What is an alignment

Arranging sequences (DNA, RNA, protein) to identify regions of similarity. Alignments are usually represented as rows within a matrix. I will only introduce pairwise alignments (two sequences) but note that also multiple sequences may be aligned.

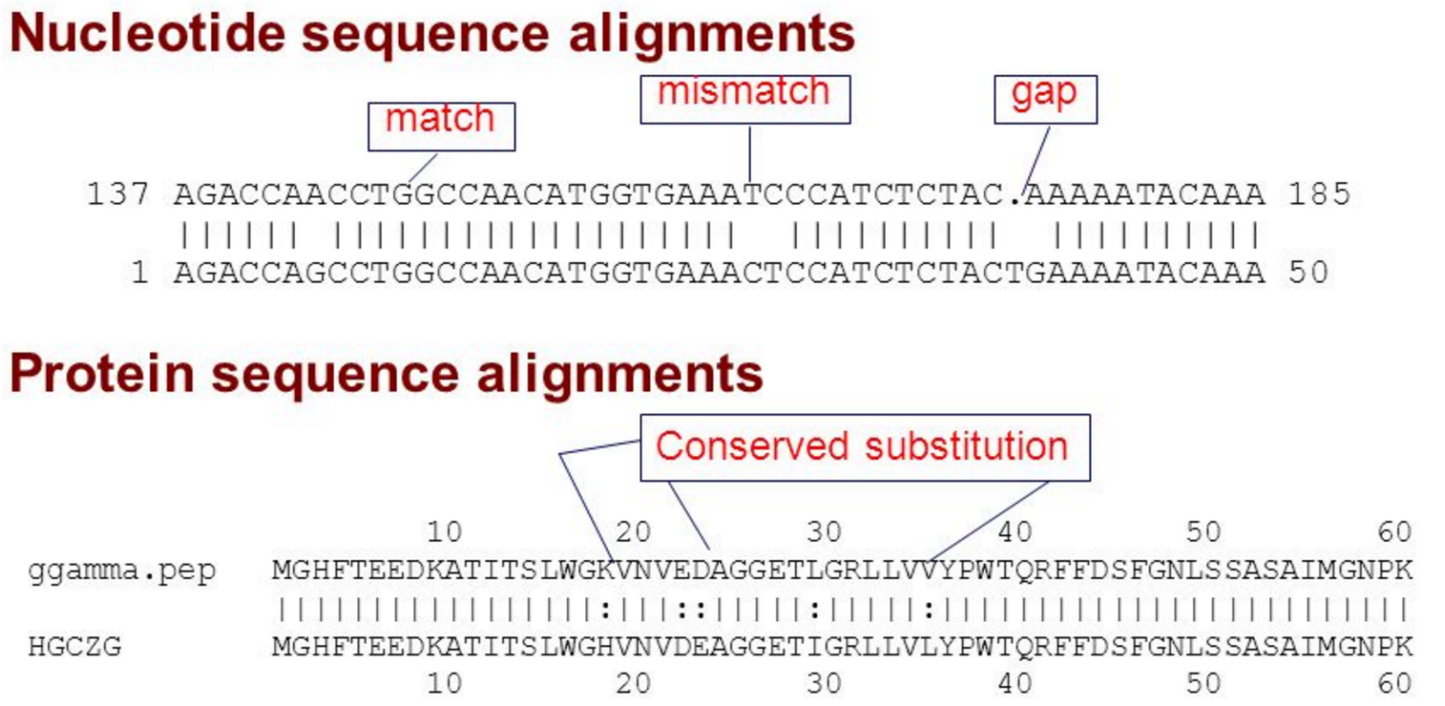


Figure 1:

The basic problem: for example given the two sequences

```
A: CCGCATGCTCC
B: GCTTACC
```

How to align them? This is a true art, and there are no perfect solutions.

Why would we even be interested in aligning two sequences?

- Evolutionary relationship (divergence)
- Structural relationship
- Identify SNPs (or other polymorphism)
- Find origin of read
- Assemble into larger fragments
- Any other ideas?

Ideal alignment

Reflects evolutionary history.

(Thanks to Carolin Kosiol for these two pics)

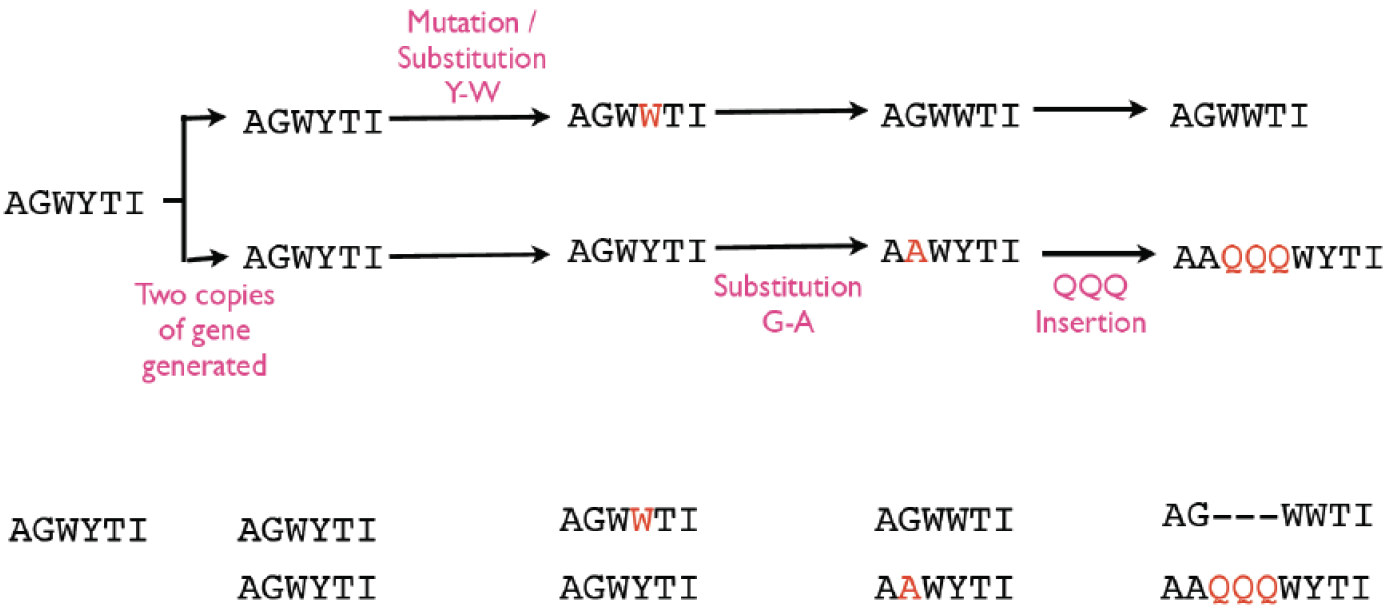
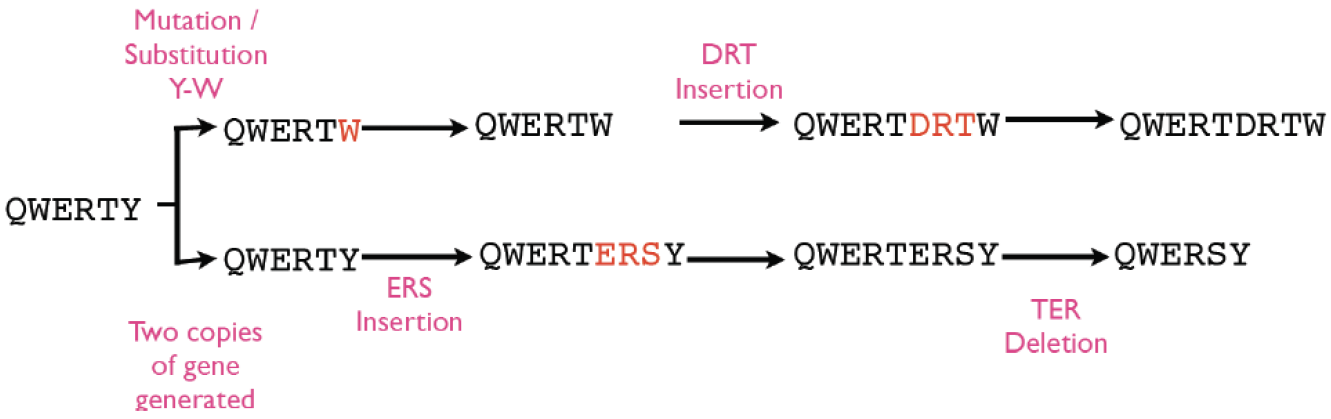


Figure 2:

How likely is it that an alignment algorithm produces this alignment?



Which alignment (X, Y or Z) shows only residues related by substitution events in the same column?

QWERTDRTW	QWERTDRTW	QWERTDRT-W
QWER---SY	Q---WERSY	QWER----SY
X	Y	Z

Figure 3:

Finding optimal alignment

Generating all possible combinations of alignments is computitionally demanding. The number of alignments increases exponentially with length, so even with the help of the computer its quite impossible to explore all possible alignments.

One vs one base

```
A: T
B: C

# Alignments: 3
A: T     -T      T-
B: C     C-      -C
```

Two vs one base

```
A: TC
B: C

# Alignments: 5
A: T-C     -TC      TC-     TC      TC
B: -C-     C--      --C     C-      -C
```

Two vs two bases

```
A: TC
B: GC

# Alignments: 11
A: T--C     --TC      TC--
B: -GC-     GC--      --GC

A: -T-C     T-C-      -TC-
B: G-C-     -G-C      G--C

A: -TC      TC-      -TC      TC-      TC
B: GC-      -GC      G-C      G-C      GC
```

Best alignment?

Assuming it is feasible to generate all possible pairwise alignments, how to find the best?

```
# Scoring system, for example:
# mismtach -1
# match 1
# gap -2

# Alignment 1      Score: -8
A: -T-C
B: G-C-

# Alignment 2      Score: -5
A: TC-
B: G-C
```

```
# Alignment 3      Score: 0
A: TC
B: GC
```

Task find the best alignment with the following parameters

```
# mismtach -3
# match 1
# gap -1
```

Take away: Number of alignments is exponentially growing with lengths; So an efficient solution is needed for aligning sequences.

Needleman Wunsch Algorithm: Global alignment

Dynamic programming turned out to be one of the most efficient ways to construct an alignment. Basic idea: the best alignment can be gradually built, always only adding one base to the alignment. This algorithm is computitionally efficient O(mn); Nucleotide sequences as well as amino acid sequences may be aligned (actually anything with a scoring matrix may be aligned).

Example (thanks to Wikipedia)

Global alignment is from end to end:

```
GCATGCT
GATTACA
```

1.) Choose the scores

- Match score (two bases are identical): 1
- Mismatch penalty (two bases are not identical): -1
- Gap penalty (indel): -1

Note that a scoring matrix may be used instead (e.g. for amino acid substitutions)

2.) Construct the grid

		G	C	A	T	G	C	U
G								
A								
T								
T								
A								
C								
A								

Figure 4:

3.) Fill in the gap columns

We start with a zero in the top/left cell than subtract the mismatch penalty in the gap columns

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

Figure 5:

4.) Fill in the rest of the matrix

		G
	0	-1
G	-1	?

Figure 6:

$$M_{i,j} = \text{maximum of} \begin{cases} M_{i-1,j-1} + S_{i,j} \\ M_{i,j-1} + w \\ M_{i-1,j} + w \end{cases}$$

Figure 7:

We have three options

- move down: -1 -1 = -2
- move right: -1 -1 = -2
- move diagonal: 0 + 1 = 1
- So what are we doing?
- Don't forget to remember the path we took! How could we remember the path we took?
- How could we move to fill in the matrix? Remember we always need three cells (left,top,diagonal)

	Seq1	G	C	A	T	G	C	T
Seq2	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	1
A	-5	-3	-3	-1	0	0	0	0
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Figure 8:

5.) Constructing the alignment: Backtracking (traceback)

	Seq1	G	C	A	T	G	C	T
Seq2	0	←	←	←	←	←	←	←
G	↑	↖	←↖	←↖	←↖	←↖	←↖	←↖
A	↑	↑↖	↖	↖	←↖	←↖	←↖	←↖
T	↑	↑↖	↑↖	↑↖	↖	←↖	←↖	←↖
T	↑	↑↖	↑↖	↑↖	↑↖	↖	←↖	↖
A	↑	↑↖	↑↖	↖	↑↖	↑↖	↖	↑↖
C	↑	↑↖	↖	↑↖	↑↖	↑↖	↖	←↖
A	↑	↑↖	↑↖	↖	↑←↖	↑↖	↑↖	↖

Figure 9:

To find the best alignment move from bottom-right corner to the top-left. In our case there are two equaly good paths.

```
GCAT-GCT
| ||  |
G-ATTACA
```

Task:

- Whats the score of the above best alignment (1,-1,-1)
- Whats the score of the following alignment:

```
GCATGCT
| ||
GATTACA
```

- How would the score of these two alignments change when we change the gap penalty to -2

Super task:

Open Excel (or Google Sheets) and align the following two sequences (1,-1,-1) using Needleman Wunsch; Generate the matrix and perform the backtracking; Any creative ideas how to find the correct path during backtracking?

```
A: CCCGCATGCTCC
B: AGCTGTA
```

Smith Waterman Algorithm: Local alignment

Only part of the sequences need to align. Find the best subalignment. The Smith Waterman algorithm is very similar to the Needleman Wunsch algorithm. Amino acids and nucleotide sequences may be aligned using SW-algorithm.

Local vs global

```
A: CCCGCATGCTCC
B: AAGCATAA

#GLOBAL: for example
CCCGCATGCTCC
-AAGCATAA---


#LOCAL: for example
GCAT
GCAT
```

1) Pick a scoring scheme

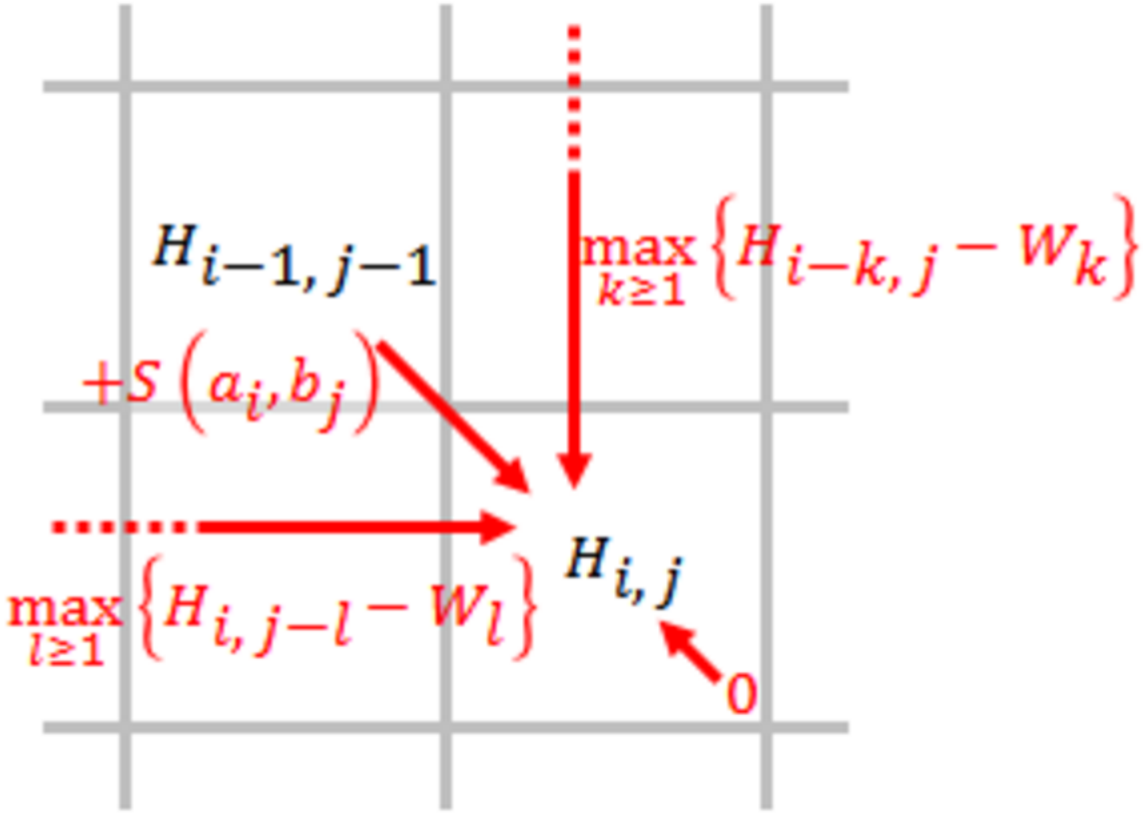
E.g substitution matrix (for aa), and gap penalties

2) Set the first row and the first column to zero

(Remember with Needleman Wunsch we used the gap penalty to initialize the first row and the first column)

3) Fill in the rest of the matrix

$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \\ \max_{l \geq 1}\{H_{i,j-l} - W_l\}, \\ 0 \end{cases} \qquad (1 \leq i \leq n, 1 \leq j \leq m)$$



4) Use Backtracking to find the alignment

Start at the highest value in the matrix and stop the alignment at the first zero value. (Remember with Needleman Wunsch backtracking starts at lower right corner and ends in top left corner)

Nice visualization of the algorithm https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm#/media/File:Smith-Waterman-Algorithm-Example-En.gif

Super Task

Open Excel (or Google Sheets) and align the following two sequences (1,-1,-1) using Smith Waterman; Generate the matrix and perform the backtracking; Any creative ideas how to find the correct path during backtracking?

```
A: CCCGCATGCTCC
B: AGCTGTA
```

Substitution matrices for amino acids and nucleotide sequences

So far we used a super simplistic scoring matrix:

```
match     =  1
mismatch  = -1
```

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

Figure 10:

Especially for aa, substitution matrices are in wide use

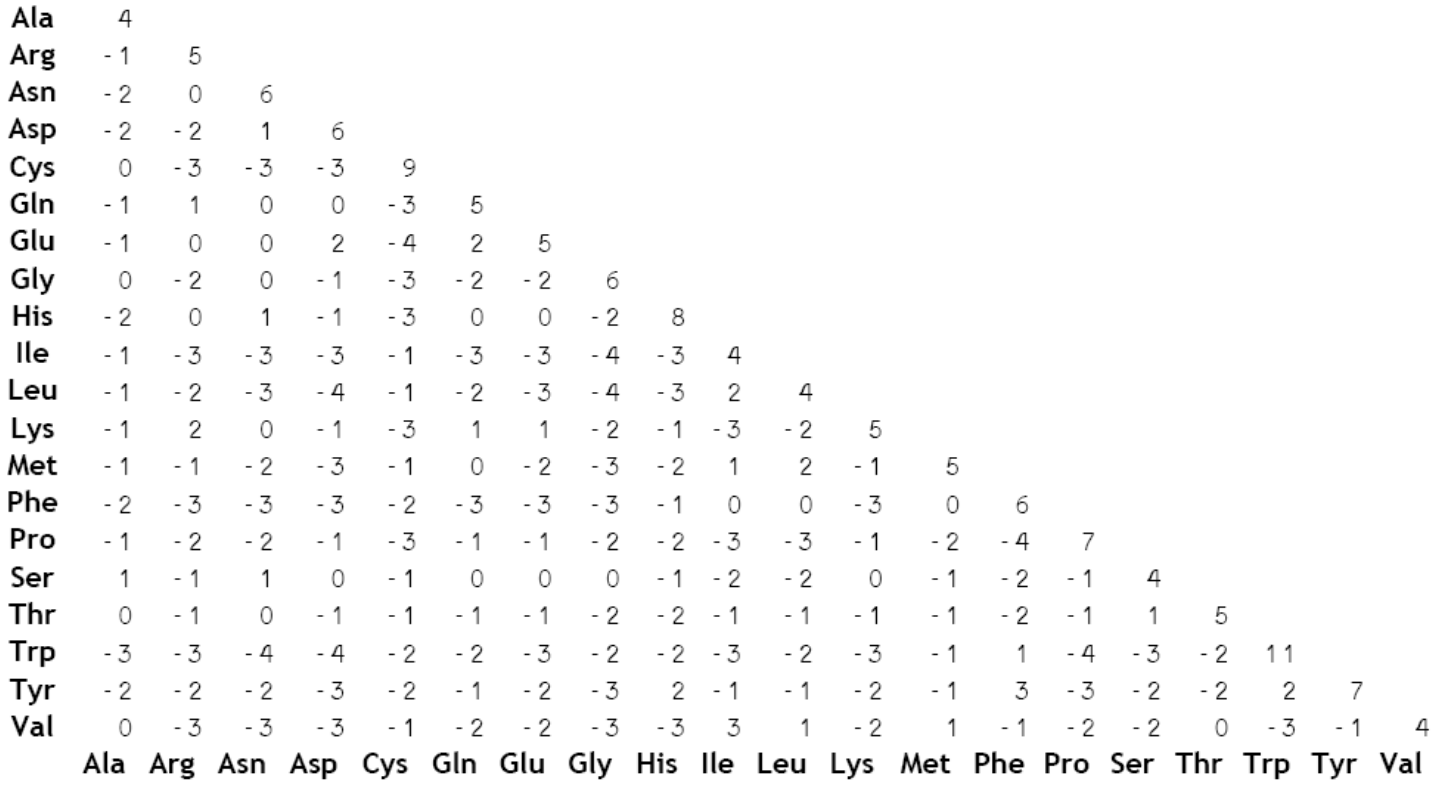

Ala 4
Arg -1 5
Asn -2 0 6
Asp -2 -2 1 6
Cys 0 -3 -3 -3 9
Gln -1 1 0 0 -3 5
Glu -1 0 0 2 -4 2 5
Gly 0 -2 0 -1 -3 -2 -2 6
His -2 0 1 -1 -3 0 0 -2 8
Ile -1 -3 -3 -3 -1 -3 -3 -4 -3 4
Leu -1 -2 -3 -4 -1 -2 -3 -4 -3 2 4
Lys -1 2 0 -1 -3 1 1 -2 -1 -3 -2 5
Met -1 -1 -2 -3 -1 0 -2 -3 -2 1 2 -1 5
Phe -2 -3 -3 -3 -2 -3 -3 -3 -1 0 0 -3 0 6
Pro -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4 7
Ser 1 -1 1 0 -1 0 0 0 -1 -2 -2 0 -1 -2 -1 4
Thr 0 -1 0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1 1 5
Trp -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1 1 -4 -3 -2 11
Tyr -2 -2 -2 -3 -2 -1 -2 -3 2 -1 -1 -2 -1 3 -3 -2 -2 2 7
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

Figure 11:

BLAST

Uses a heuristic, searching for small words (k-mer) and than performing dynamic programming in regions with many k-mer matches.

Major question: is my alignment significant?

Any match, especially of short sequences may be entirely due to chance and thus not necessarily reflect a meaningful biological relationship. How to find if a match of two sequences is due to chance?

BLAST provides an e-value. As a rule of thump many researchers take an e-value lower than $e^{-10} as significiant. But the lower the better. Also note that the value depends on the length of the sequences. For small fragments it is impossible to obtain very low values.

$E(S) = Kmne^{-\lambda S}$

- S score
- K and λ scaling parameters dependent on search space and scoring scheme
- m,n size of the sequences
- probability of finding at least one match with our score $p = 1 - e^{-E(S)}$

Links

- http://bioinformaticnotes.com/Needle-Water/

Remember: all alignments are wrong!