EMBL-EBI

Help - EMBOSS-Align

INTRODUCTION

LAUNCH APPLICATION

Nucleotide Tutorial

Protein Tutorial

This tool is used to compare 2 sequences.

- When you want an alignment that covers the whole length of both sequences, use needle.
- When you are trying to find the best region of similarity between two sequences, use water.

Program Descriptions **Parameters**

- **Needleman-Wunsch** global alignments.
- **Smith-Waterman** local alignments.

- **Input Sequences**.
- **Scoring Matrices**.
- **Default Gap Penalties**.
- **References and Authors**.

Output Formats

- **Needleman-Wunsch** global alignments.
- **Smith-Waterman** local alignments.

PROGRAM: NEEDLE

- Description

  This program uses the Needleman-Wunsch global alignment algorithm to find the optimum alignment (including gaps) of two sequences when considering their entire length.

  The Needleman-Wunsch algorithm is a member of the class of algorithms that can calculate the best score and alignment in the order of mn steps, (where 'n' and 'm' are the lengths of the two sequences). These dynamic programming algorithms were first developed for protein sequence comparison by Needleman and Wunsch, though similar methods were independently devised during the late 1960's and early 1970's for use in the fields of speech processing and computer science.

  What is the optimal alignment? Dynamic programming methods ensure the optimal global alignment by exploring all possible alignments and choosing the best. It does this by reading in a scoring matrix that contains values for every possible residue or nucleotide match. Needle finds an alignment with the maximum possible score where the score of an alignment is equal to the sum of the matches taken from the scoring matrix.

  An important problem is the treatment of gaps, i.e., spaces inserted to optimise the alignment score. A penalty is subtracted from the score for each gap opened (the 'gap open' penalty) and a penalty is subtracted from the score for the total number of gap spaces multiplied by a cost (the 'gap extension' penalty).

  Typically, the cost of extending a gap is set to be 5-10 times lower than the cost for opening a gap.

- Needle

  This is a true implementation of the Needleman-Wunsch algorithm and so produces a full path matrix. It therefore cannot be used with genome sized sequences unless you've a lot of memory and a lot of time.

  Needle is for aligning two sequences over their entire length. This works best with closely related sequences. If you use needle to align very distantly-related sequences, it will produce a result but much of the alignment may have little or no biological significance.

  A true Needleman Wunsch implementation like **needle** needs memory proportional to the product of the sequence lengths. For two sequences of length 10,000,000 and 1,000 it therefore needs memory proportional to 10,000,000,000 characters. Two arrays of this size are produced, one of integers and one of floats so multiply that figure by 8 to get the memory usage in bytes. That doesn't include other overheads. Therefore only use water and needle for accurate alignment of reasonably short sequences.

PROGRAM: WATER

- Description

  Water uses the Smith-Waterman algorithm (modified for speed enhancements) to calculate the local alignment.

  The Smith-Waterman algorithm is a member of the class of algorithms that can calculate the best score and local alignment in the order of mn steps, (where 'n' and 'm' are the lengths of the two sequences). These dynamic programming algorithms were first developed for protein sequence comparison by Smith and Waterman, though similar methods were independently devised during the late 1960's and early 1970's for use in the fields of speech processing and computer science.

  A local alignment searches for regions of local similarity between two sequences and need not include the entire length of the sequences. Local alignment methods are very useful for scanning databases or other circumstances when you wish to find matches between small regions of sequences, for example between protein domains.

  Dynamic programming methods ensure the optimal local alignment by exploring all possible alignments and choosing the best. It does this by reading in a scoring matrix that contains values for every possible residue or nucleotide match. **Water** finds an alignment with the maximum possible score where the score of an alignment is equal to the sum of the matches taken from the scoring matrix.

  An important problem is the treatment of gaps, i.e., spaces inserted to optimise the alignment score. A penalty is subtracted from the score for each gap opened (the 'gap open' penalty) and a penalty is subtracted from the score for the total number of gap spaces multiplied by a cost (the 'gap extension' penalty).

  Typically, the cost of extending a gap is set to be 5-10 times lower than the cost for opening a gap.

  Water is a true implementation of the Smith-Waterman algorithm and so produces a full path matrix. **It therefore cannot be used with genome sized sequences** unless you have a **lot** of memory and a lot of time.

  Local alignment methods only report the best matching areas between two sequences - there may be a large number of alternative local alignments that do not score as highly. If two proteins share more than one common region, for example one has a single copy of a particular domain while the other has two copies, it may be possible to "miss" the second and subsequent alignments. You will be able to see this situation if you have done a dotplot and your local alignment does not show all the features you expected to see.

  Water is for aligning the best matching subsequences of two sequences. It does not necessarily align whole sequences against each other; you should use **needle** if you wish to align closely related sequences along their whole lengths.

  A true Smith Waterman implementation like **water** needs memory proportional to the product of the sequence lengths. For two sequences of length 10,000,000 and 1,000 it therefore needs memory proportional to 10,000,000,000 characters. Two arrays of this size are produced, one of integers and one of floating point (real) numbers so multiply that figure by 8 to get the memory usage in bytes. That doesn't include other overheads. Therefore only use water and needle for accurate alignment of reasonably short sequences.

- INPUT SEQUENCES

  You can cut and paste or type a sequence into the large text window. We recommend the submission of sequences less than 10,000 characters in length. A free text (raw) sequence is simply a block of characters representing a DNA/RNA or Protein sequence. You may also paste a sequence in GCG, FASTA, EMBL, GenBank, PIR, NBRF, Phylip or Swiss-Prot format. Partially formatted sequences will not be accepted. Copying and pasting directly from word processors may yield unpredictable results as hidden/control characters may be present. Adding a return to the end of the sequence may help certain applications understand the input. In fasta format, do not leave a space at the start of a sequence header before the ">" symbol, or end this line with a full stop. Some examples of common sequence formats may be seen here.

- UPLOAD A FILE

  You may upload a file from your computer which contains a valid sequence in any format (GCG, FASTA, EMBL, GenBank, PIR, NBRF, Phylipor Swiss-Prot) using this option. Please note that this option only works with Netscape Browsers or Internet Explorer version 5 or later. Some word processors may yield unpredictable results as hidden/control characters may be present in the files. It is best to save files with the Unix format option to avoid hidden windows characters. Some examples of common sequence formats may be seen here.

- DEFAULT SUBSTITUTION SCORING MATRICES

  For protein sequences EBLOSUM62 is used for the substitution matrix. For nucleotide sequences, EDNAMAT is used. Others can be specified for more or less sensitive alignments. More about matrices.

- MOLECULE TYPE

  In order to choose from the correct set of matrices the molecule type (DNA or PROTEIN) must be chosen. More about types of molecules

- DEFAULT GAP PENALTIES

	Protein		**DNA**	
program	Gap Open	Gap Extend	Gap Open	Gap Extend
water	10.0	0.5	10.0	0.5
needle	10.0	0.5	10.0	0.5

  More about gaps.

- EXAMPLE

  The following sequences were used.

Sequence 1

```
NGPSTKDFGKISESREFDNQNGPSTKDFGKISESREFDNQ
```

Sequence 2

```
QNQLERSFGKINMRLEDALVQNQLERSFGKINMRLEDALV
```

Needle Results(Global Alignment)

```
########################################
# Program:  needle
# Rundate:  Tue Jul 23 12:21:46 2002
# Report_file: /ebi/extserv/old-work/236801027423305.needle.res
########################################
#=======================================
#
# Aligned_sequences: 2
# 1:
# 2:
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 41
# Identity:      11/41 (26.8%)
# Similarity:    15/41 (36.6%)
# Gaps:           2/41 ( 4.9%)
# Score: 22.0
#
#
#=======================================

                     1 NGPSTKDFGKISESREFDNQNGPSTKDFGKISESREFDNQ      40
                       .....:.||||:...|.........:.||||:...| |..
                     1 QNQLERSFGKINMRLEDALVQNQLERSFGKINMRLE-DALV     40


#---------------------------------------
#---------------------------------------
```

Water Results(Local Alignment)

```
########################################
# Program:  water
```

```
# Rundate:  Tue Jul 23 12:21:59 2002
# Report_file: /ebi/extserv/old-work/237041027423316.water.res
########################################
#======================================
#
# Aligned_sequences: 2
# 1:
# 2:
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 25
# Identity:       9/25 (36.0%)
# Similarity:    12/25 (48.0%)
# Gaps:           0/25 ( 0.0%)
# Score: 32.0
#
#
#======================================

                     8 FGKISESREFDNQNGPSTKDFGKIS     32
                       ||||:...|.........:.||||:
                     8 FGKINMRLEDALVQNQLERSFGKIN     32


#---------------------------------------
#---------------------------------------
```

The **%id** is the percentage of identical matches between the two sequences over the reported aligned region.

The **%similarity** is the percentage of matches between the two sequences over the reported aligned region where the scoring matrix value is greater or equal to 0.0.

The **Overall %id** and **Overall %similarity** are calculated in a similar manner for the number of matches over the length of the longest of the two sequences.

- REFERENCES

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.
abstract

Kruskal J.B. (1983)
An overview of sequence comparison.
In D. Sankoff and J. B. Kruskal, (ed.), Time warps, string edits and macromolecules: the theory and practice of sequence comparison, pp. 1-44 Addison Wesley.

Smith T.F., Waterman M.S. (1981)
Identification of common molecular subsequences.
Journal of Molecular Biology 147: 195-197.

abstract

- AUTHORS

  EMBOSS::needle and EMBOSS::water were written by Alan Bleasby (ableasby@hgmp.mrc.ac.uk)

  This page was produced by EBI Support and is maintained by DocHelp@epo.org