

Algorithms in Bioinformatics

Andrey Prjibelski (andrewprzh@gmail.com),
Ira Vasilinetc (vasilinetc.ira@gmail.com)

20.02.2014

1 Alignment

Task 1

Find Levenshtein distance between two very long sequences if they have distance less than $k = 100$. Running time should be less than 1 minute for strings of length 1 million.

Input:

FASTA file with two sequences.

Output:

Levenshtein distance between these sequences if distance is less than k or “not similar” in other case.

Example:

Input:

```
>seq1
acgtacgt
>seq2
aagtacgt
```

Output:

1

Task 2

Find an optimal global alignment of two very long sequences if they have Levenshtein distance less than $k = 100$. If several alignments with equal scores exist, then output one of them. Running time should be less than 1 minute for strings of length 1 million.

Input:

FASTA file with two sequences.

Output:

An alignment of these sequences if Levenshtein distance is less than k or “not similar” in other case.

Example:

Input:

```
>seq1
acgtacgt
>seq2
agtacgt
```

Output:

```
acgtacgt
a-gtacgt
```

Task 3

Find a multiple alignment of a collection of 4 sequences. Use given scoring matrix for mismatches penalty and gap penalty. If several alignments with

equal scores exist, then output one of them. Running time should not exceed 1 minute for 4 sequences of length 50.

Input:

FASTA file with several sequences, file with an integer representing gap penalty and scoring matrix. Matrix goes in ACGT-order one row per line.

Output:

Optimal alignment of all sequences from the collection.

Example:

Input file 1:

```
>seq1
acgtacgt
>seq2
agtacgt
>seq3
cgtttacgt
>seq4
cgttacat
```

Input file 2:

```
1
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
```

Output:

```
acgt-acgt
a-gt-acgt
-cgtttacgt
-cgttacat
```

Task 4

Let \tilde{a} be the reverse-complement sequence of a . Find the maximum substring s_1 in the sequence s such that \tilde{s}_1 is contained in s and does not overlap with s_1 . Running time should be less than 1 minute for strings of length 1 million.

Input:

FASTA file with sequence s .

Output:

FASTA file with subsequence s_1 .

Example:

Input file:

```
>seq
ACGTTTACGT
```

Output:

```
ACGT
```

Task 5

Given two strings s and t . Find substring s' of s that maximize an alignment score with respect to t and output optimal alignment of s' against t . If multiple such alignments exist, then you may output any one. Running time should not exceed 1 minute for sequences of length less than 1000.

Input:

FASTA file with two sequences s and t .

Output:

An optimal alignment score of s' and t , followed by an optimal alignment s' against t . If multiple such alignments exist, then you may output any one.

Example:

Input:

```
>seq1
acgtacgt
>seq2
acgt
```

Output:

```
0
acgt
acgt
```