

Optimal alignment of two sequences I

Bioinformatics

Miha Gazvoda

January 17, 2017

1 Instructions

Write a program that for a pair of amino acid sequences calculates:

- the edit distance and the similarity, and
- the best local and global alignments.

(You don't have to use gap penalty functions.)

Data should be read from text files. Data includes: 2 sequences, a similarity matrix, and a distance matrix (both matrices of size 5 × 5, since there are 4 nucleotides and space).

The output of the program should contain the optimal alignment, its edit distance and similarity.

2 Implementation

I implemented tasks in Python programming language. Specifically, in the Jupyter Notebook. I implemented functions which are the same (with different arguments) for edit distance or similarity.

2.1 Optimal alignment

2.1.1 Functions

d(mode, a,b) calculates based on *mode* similarity or edit distance between two characters (or space) *a* and *b*.

preparation(amino1, amino2, edit_m, direction_m, distance_mode) calculates edit/similarity matrix and direction matrix.

from_where(mode,directions,i,j,a,b,c) based on mode and position function calculates edit/similarity value of (i,j) elements of matrix from elements $(i-1,j-1)$, $(i-1,j)$, and $(i,j-1)$.

optimal_alignment(direction_m,i,j,str1,str2) builds corresponding alignments of the input amino acid sequences.

2.1.2 Code

Then I just initialized matrices for distance and directions and called functions *preparation()* and *optimal_alignment()*.

2.2 Local alignment

2.2.1 Functions

local_preparation(amino1, amino2, edit_m, direction_m, distance_mode) returns similarity and direction matrix from local alignment.

local_from_where(mode,direction_m,i,j,a,b,c) calculates value in (i, j) matrix position and values in direction matrix.

local_alignment(direction_m,i,j,str1,str2) calculates local alignment of two sequences.

2.2.2 Code

I initialized matrices for distance and directions and called functions *local_preparation()* and *local_alignment()*.