

Laboratorium ze Wstępu do bioinformatyki

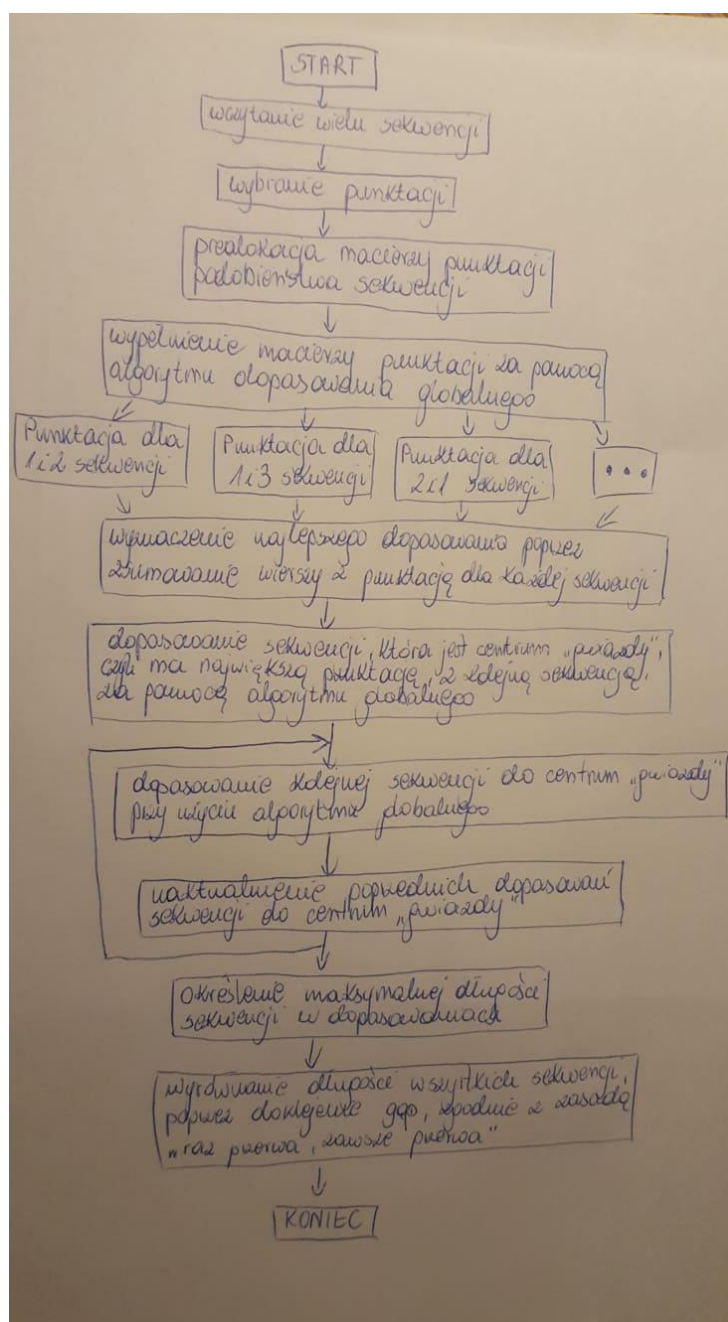
Typ dokumentu:	Zadanie 4	Data oddania zadania: 16.05.2019
Tytuł:	Progresywne dopasowanie wielosekwencyjne (progressive MSA – multiple sequence alignment)	

Autor: Aleksandra Brela, 229636

1. Odnosnik do repozytorium z kodem źródłowym:

<https://github.com/Abrela/MSA>

2. Schemat blokowy algorytmu dopasowania wielu sekwencji – MSA:



3. Analiza złożoności obliczeniowej algorytmu –

a. Funkcja multipleSequenceAlignment()

```
1: function [alignmentOut, maxOut, starSeqOut, starHeaderOut] = multipleSequenceAlignment(fastaStruct,
2: match, mismatch, gap)
3: dimension = length(fastaStruct); przypisanie
4: MSAScore = zeros(dimension); przypisanie, prealokacja
5: for i = 1:dimension inkrementacja, iteracje
6: for j = 1:dimension inkrementacja, iteracje
7: if i==j warunek, porównanie
8: MSAScore(i,j) = 0; przypisanie
9: else
10: [ mNW, matrixSize, bias, vertical, horizontal ]= ...
11: matrixScore(fastaStruct(i).sequence, fastaStruct(j).sequence, match, mismatch, gap); subfunkcja
12: [pathOut, identityOut, gapsOut, s1Out, s2Out, lengthAlignmentOut, scoreOut] = matchPath(matrixSize, bias,
13: vertical, horizontal, fastaStruct(i).sequence, fastaStruct(j).sequence, match, mismatch, gap, mNW );
subfunkcja
14: MSAScore(i,j) = scoreOut; przypisanie
15: end
16: end
17: end
18: scoreRow = []; przypisanie
19: for i = 1:dimension inkrementacja, iteracje
20: row = MSAScore(i, :); przypisanie
21: scoreRow1 = sum(row); przypisanie
22: scoreRow = [scoreRow; scoreRow1]; przypisanie
23: end
24: maxScoreRow = max(scoreRow); przypisanie
25: [coordinateRowX, coordinateRowY] = find(maxScoreRow == scoreRow); przypisanie
26: seq = fastaStruct(coordinateRowX).sequence; przypisanie
27: for j = 1:length(scoreRow) inkrementacja, iteracje
28: if isequal(seq, fastaStruct(j).sequence) warunek, porównanie
29: seq = fastaStruct(j).sequence; przypisanie
30: else
31: [ mNW, matrixSize, bias, vertical, horizontal ]= ...
32: matrixScore(seq, fastaStruct(j).sequence, match, mismatch, gap); subfunkcja
33: [pathOut, identityOut, gapsOut, s1Out, s2Out, lengthAlignmentOut, scoreOut] = ...
34: matchPath(matrixSize, bias, vertical, horizontal, seq, fastaStruct(j).sequence, match, mismatch, gap, mNW );
subfunkcja
35: fastaStruct(j).sequence = s2Out; podstawienie
36: end
37: end
38: A = []; przypisanie
39: for i = 1:length(scoreRow) inkrementacja, iteracje
40: a = length(fastaStruct(i).sequence); przypisanie
41: A=[A; a]; przypisanie
42: end
43: for i = 1:(length(scoreRow)) inkrementacja, iteracje
44: while length(fastaStruct(i).sequence) ~= max(A) warunek, porównanie
45: if length(fastaStruct(i).sequence) < max(A) warunek, porównanie
46: fastaStruct(i).sequence = strcat(fastaStruct(i).sequence, '-'); podstawienie
47: end
48: end
49: end
50: alignmentOut = fastaStruct; przypisanie
51: maxOut = max(A); przypisanie
52: starSeqOut = fastaStruct(coordinateRowX).sequence; przypisanie
53: starHeaderOut = fastaStruct(coordinateRowX).header; przypisanie
54: end
```

4. Przykład otrzymanego dopasowania dla czterech sekwencji powiązanych ewolucyjnie:

Wybierz rodzaj wczytywania plików - wpisz file lub ncbi: file

Liczba sekwencji: 4

Wpisz nazwę pliku lub identyfikator: AsiaticElephant.txt

Wpisz nazwę pliku lub identyfikator: Rhino.txt

Wpisz nazwę pliku lub identyfikator: Mastodon.txt

Wpisz nazwę pliku lub identyfikator: AfricanaElephant.txt

Podaj punktację za dopasowanie: 1

Podaj punktację za niedopasowanie: 0

Podaj punktację za przerwę: -1

```
>AsiaticElephant gi|92471      GAAATTTTCGGCTCACTACTAGGAGCGTGCCTAATTACCCAAATCCTAACAGGATTATTCC      228
>Rhino gi|12903|emb|X5628      GAAATTTTGGCTCTCTACTAGGAATCTGCCTAATCCTACAAATCCTAACCGGACTATTTC      228
***** *****
>AsiaticElephant gi|92471      GAAATTTTCGGCTCACTACTAGGAGCGTGCCTAATTACCCAAATCCTAACAGGATTATTCC      228
>NorthAmericanMastodon gi      GAAATTTTCGGCTCACTACTAGGAGCATGCCTAATTACCCAAATCCTAACAGGATTATTCC      228
*****
>AsiaticElephant gi|92471      GAAATTTTCGGCTCACTACTAGGAGCGTGCCTAATTACCCAAATCCTAACAGGATTATTCC      228
>CC935997.1 ele-ncrm5 ele      -AAGC--CAT-TC-CTACAAGTAGCC--CCT--TGGCACT--TGCTT---GGATTATTTC-      228
**      *      ** ***** **      *      *      *      *      *****
```