

Bioinformatyka

Wykonane przez:

Denys Hromniuk 141232

Pavlo Ravliv 135412

Wybrany problem:

Problem SBH z informacją o położeniu oraz wszystkimi rodzajami błędów.

Algorytm zachłanny

Dostajemy odczytywane DNA,

dzielimy na oligonukleotydy o zadanej długości -> szuflujemy je

Działanie algorytmu zachłannego skoro wiemy, który oligonukleotyd jest początkowy to od niego zaczynamy

```
function greedy(firstOligonucleotide ,  
allOligonucleotides[],resultLength): result  
  dodajemy do result nasz firstOligonucleotide  
  currentOligoNucleotide = firstNucleotide  
  while currentOligoNucleotide oraz len(result) <= resultLength ma  
  sasiedztwo  
    wybierz luk o najwyzszej wadze i dodaj do result  
    currentOligoNucleotide = nextOligoNucleotide
```

Algorytm Heurystyczny:

Jako warunek stopu wybraliśmy jeśli wynik się nie polepsza przez N iteracji

0. W zerowym kroku generujemy początkowe rozwiązanie za pomocą algorytmu zachłannego While not warunek stop
1. Dla wszystkich osobników z populacji krzyżujemy jeden osobnik z pozostałymi.

Krzyżowanie(solution1,solution2) :

Polega na:

L -> [A | B C D] <- P

$L \rightarrow [C | B D A] \leftarrow P$

Dzielimy solution1 i solution2 na dwie czesci i zostawiamy czesc L z solution1 oraz czesc P z solution2 i dostajemy potomka1 (solution1(L),solution2(P)) oraz drugi pomtek2(solution1(P),solution2(L))

2. W dwóch wybranych potomkach pewnego losowego rozwiązania robimy mutacje polegające na losowej wymianie dwóch oligonukleotydów
3. Po krosowaniu zostawiamy populacje w dopuszczalnych granicach i osobniki z niskimi wartościami są zastępowane przez potomków osobników z wysokimi wartościami
4. Powtarzamy proces generowania nowych osobników i mutacji potomków do tej pory aż warunek stopu nie zostanie spełniony