

Bionformatyka

Wykonane przez:

Denys Hromniuk 141232

Pavlo Ravliv 135412

Wybrany problem:

Klasyczny problem SBH z błędami negatywnymi obu typów.

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:

Aspiration criteria - kryterium ktore jest evaluowane jako **True** w przypadku jesli pod czas danej iteracji dostajemy lepsze rozwiazanie od najlepszego wczesniejszego i znajduje sie na liscie Tabu

Jako warunek stopu wybralismy liczbe iteracji

0. W zerowym kroku generujemy początkowe rozwiązanie za pomocą algorytmu zachłannego

1. Generujemy otoczenie \$\$\$. Otoczenie \$\$\$ jest generowane za pomocą funkcji :

```
fun generateNeighbourhood(solution) : neighbourhood[]
```

funkcja zwraca otoczenie dla danego rozwiązania. Myślimy o generowaniu otoczenia w sposób wymiany poszczególnych nukleotydów lub podciągów nukleotydów

2. Z posród otoczenie wybieramy rozwiązanie dopuszczalne (nie tabu albo spełnie aspiration criteria) i sprawdzamy czy jest lepsze od wcześniej otrzymanego rozwiązania czyli sprawdzamy `if(tabu && !aspirationCriteria)` to odrzucamy to rozwiązanie od razu `if(tabu && aspirationCriteria)` to robimy wyjątek jeśli nowe wygenerowane rozwiązanie jest lepsze od wcześniej otrzymanego rozwiązania
3. Sprawdzamy czy warunek stopu jest ewaluowany jako `True` jeśli jest `True` to kończymy obliczanie w przeciwnym przypadku przechodzimy do kroku 4
4. Aktualizujemy naszą listę tabu i przechodzimy do kroku 1