

Implementacja algorytmów NeedlemanaWunscha i Smitha-Watermana

Zależności:

- Numpy 1.12.1

W module N_W znajduje się funkcja (jeśli korzystasz z python 2.7 to w module N_W 2.7)
`alignment(seqs, go, ge, s, ret_max=50, linear_memory = False, local = False, time_limit = False)`

gdzie :

`seqs` - tablica sekwencji, które chcemy dopasowywać

`s` - prosta punktacja za dopasowanie/niedopasowanie lub macierzą substytucji w postaci słownika, prosta punktacja jest podana w postaci tablicy `[x,y]` gdzie `x` jest punktacją za dopasowanie a `y` za niedopasowanie.

`go` - kara za otwarcie przerwy

`ge` - kara za rozszerzenie przerwy

`ret_max` - maksymalna liczba zwracanych dopasowań

`linear_memory` – jeśli `True` zmniejsza złożoność pamięciową do liniowej kosztem złożoności obliczeniowej (domyślnie `False`)

`local` – jeśli `True` to poszukujemy najlepszego lokalnego dopasowania w przeciwnym wypadku poszukujemy najlepszego dopasowania globalnego (domyślnie `False`)

`time_limit` – limit czasowy na wykonanie algorytmu podany w sekundach lub `False`, gdy nie przewidujemy limitu. Jeżeli limit czasu jest przekroczony, to program zwraca wartość -1 i wypisuje o tym komunikat.

W programie jest zaimplementowanych 5 algorytmów

- wielosekwencyjne globalne dopasowanie liniowe
- wielosekwencyjne lokalne dopasowanie liniowe
- dwusekwencyjne globalne dopasowanie afiniczne
- dwusekwencyjne lokalne dopasowanie afiniczne
- dwusekwencyjne dopasowanie liniowe o liniowej złożoności pamięciowej

Więc, pewne wywołania funkcji, które nie pasują do tych algorytmów nie zostaną zrealizowane np. gdy `seqs` będzie zawierać 3 sekwencje i `go` będzie różny od zera to nie ma zaimplementowanego algorytmu, który który by obsługiwał tą sytuację. Wtedy funkcja zwróci `None`

Zwracane dopasowanie jest postaci `(value, tab_dopasowań)`

`value` jest wartością dopasowania

`dopasowań` jest tablicą która zawiera znalezione dopasowania.

W przypadku dopasowania globalnego jest ono tablicą `[seq1_dop, ..., seqn_dop]` gdzie poszczególne `seq1_dop, ..., seqn_dop` stanowią obliczone dopasowania dla odpowiednich sekwencji

W przypadku dopasowania globalnego jest ono parą `((start1, ..., startn), [seq1_dop, ..., seqn_dop])`

gdzie poszczególne `seq1_dop, ..., seqn_dop` stanowią obliczone dopasowania lokalne dla odpowiednich sekwencji a `(start1, ..., startn)` oznaczają indeksy dla poszczególnych sekwencji które mówią gdzie zaczyna się dopasowanie lokalne np. przy lokalnym dopasowaniu dwóch sekwencji para `(2,0)` mówi nam, że pierwszą dopasowaną parą jest para `(seq1[2], seq2[0])`

Pliki, które na początku zaczynają się `N_W_test` nie są potrzebne do uruchomienia programu.

Znajdują się w nich porównanie zaimplementowanej w tym projekcie funkcji do funkcji z bio pythona pairwise oraz sprawdzenie ich szybkości. Z tego powodu potrzebny jest biopython do ich uruchomienia. W testach wykorzystywane są białka MHC które są zapisane w pliku `MHC_test`.

Białka te pochodzą od człowieka, szympansa, myszy i małpy. Więcej informacji na temat tego białka można znaleźć na Wikipedii https://pl.wikipedia.org/wiki/G%C5%82%C3%B3wny_uk%C5%82ad_zgodno%C5%9Bci_tkankowej