Instytut Informatyki i Matematyki Komputerowej UJ,
opracowanie: mgr Ewa Matczyńska, dr Jacek Śmietański

Przyrównywanie sekwencji

1. Porównywanie sekwencji – wprowadzenie

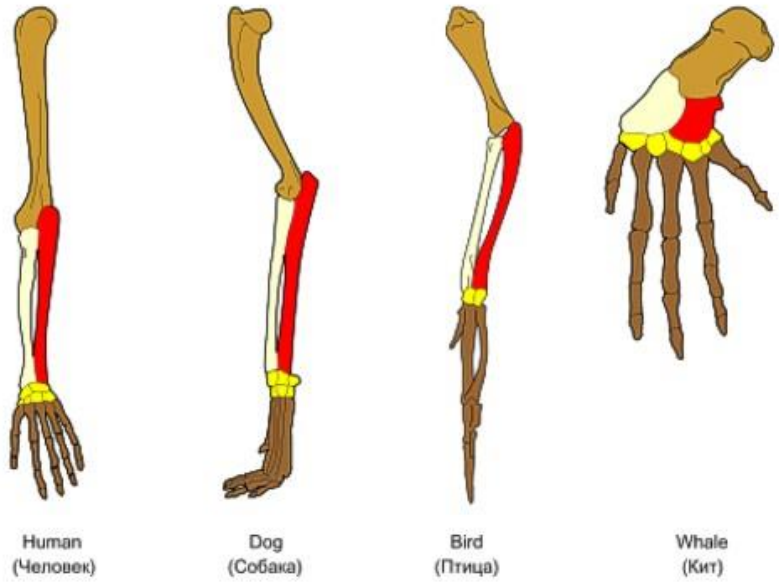
Sekwencje porównujemy po to, aby zidentyfikować miejsca podobne, które mogą świadczyć o powiązaniu ewolucyjnym czy funkcjonalnym obu sekwencji. Można również badać samopodobieństwo sekwencji.

```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************     :  *.: :
```

Rysunek 1: Dopasowanie dwóch sekwencji motywów palca cynkowego u człowieka (ang. zinc finger domain), który jest rodzajem domeny białkowej występującej w białkach wiążących DNA i bierze bezpośredni udział w związaniu cząsteczki kwasu nukleinowego przez białko.

W szczególności mówimy, że sekwencje są homologiczne jeśli wywodzą się od wspólnego przodka, tzn. że różnice między nimi są wynikiem różnicowania się w toku ewolucji.

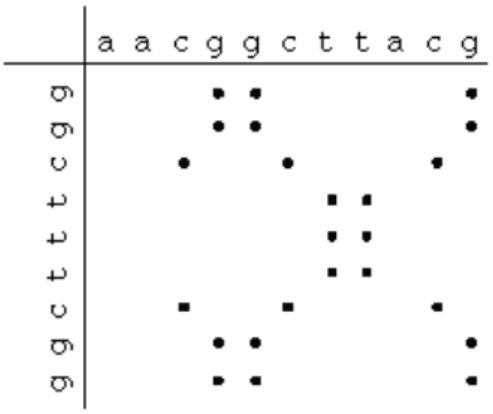


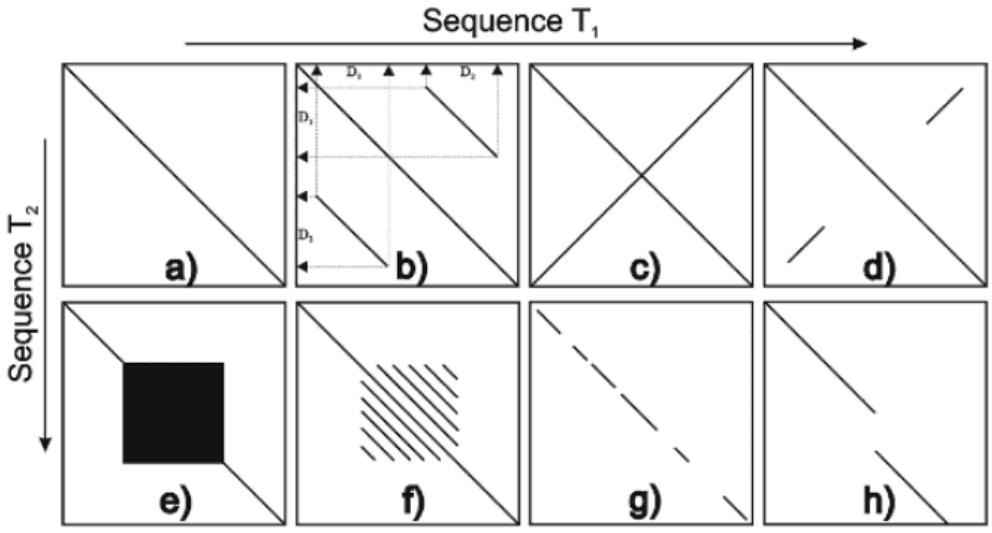
Rysunek 2: Pojęcie homologii w biologii nie dotyczy tylko sekwencji, jest dużo szersze, przykład homologii kończyn u ssaków (jeden z argumentów Darwina za teorią ewolucji).

Możliwa jest również sytuacja, kiedy sekwencje wykazują duże podobieństwo mimo braku wspólnego przodka. Najczęściej ma to miejsce wtedy, kiedy pełnią one podobną funkcję np. wiążą się do tego samego białka lub helisy DNA (np. motyw palca cynkowego powyżej), wtedy ich podobna struktura niejako wymusza pewne podobieństwo w sekwencji. Oczywiście podobieństwo może również być przypadkowe, wtedy musimy się zastanowić czy otrzymane podobieństwo jest statystycznie istotne. Tak więc, wyszukiwanie podobieństw w sekwencjach może służyć np. wyszukiwania prawdopodobnej funkcji genów, do porównywania genów, białek między gatunkami - obserwowaniu działania ewolucji.

2. Macierze kropkowe – *dot plot*

Jedną z pierwszych metod porównywania sekwencji był tzw. *dot plot*. Tworzymy macierz, której kolumny odpowiadają kolejnym resztom pierwszej sekwencji, a wiersze kolejnym resztom drugiej sekwencji. Bardzo prosta idea polega na uzupełnianiu kropkami macierzy, tam gdzie mamy zgodność. Ta metoda nie nadaje się do analizy dużej liczby sekwencji, ale można dzięki niej szybko zobrazować charakterystyczne cechy dopasowania. Często porównuje się sekwencję samą ze sobą, np. aby zobaczyć czy są jakieś fragmenty które się powtarzają.





Zadanie 1.

- Gepard (GEnome PAir - Rapid Dotter) – to narzędzie do generacji dot-plotów: http://cube.univie.ac.at/gepard
- W sekcji download - uruchom program przez Java Web Start (może być low memory version) [żeby zadziałało w pracowni, zapewne trzeba pogmerać w ustawieniach zabezpieczeń]
- Załaduj jako sekwencję 1 i 2 - sekwencję nukleotydową ludzkiego genu TPO (ID: 7173)
- Przyglądnij się rejonom samopodobieństwa sekwencji używając powiększenia.

Alternatywnie możesz przetestować inne podobne narzędzia, np. pracujące on-line:

- dotmatcher: http://www.bioinformatics.nl/cgi-bin/emboss/dotmatcher
- dotlet: https://dotlet.vital-it.ch/

3. Dopasowanie globalne i lokalne

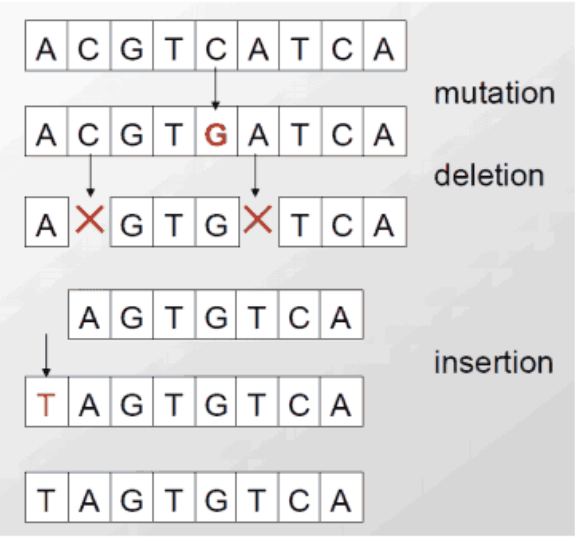
Rozróżniamy dopasowanie globalne i lokalne :

- **globalne** - dopasowuje sekwencje na całej długości, użyteczne jeśli sekwencje są w miarę podobnych rozmiarów - np. dopasowanie domen białkowych;
- **lokalne** - dopasowuje pewne lokalne regiony sekwencji, wyszukuje lokalne podobieństwa, możemy nim dopasowywać sekwencje, które znacznie różnią się długością np. wyszukujemy określonej sekwencji genu na całym chromosomie, bądź pewnego motywu białkowego w całej sekwencji białka.

```
Global  FTFTALILLAVAV
        F--TAL-LLA-AV

Local   FTFTALILL-AVAV
        --FTAL-LLAAV--
```

Dla przypomnienia, jakie zmiany mogą nam zajść w łańcuchu DNA lub aminokwasowym ?



4. Algorytm LCS

Ponieważ algorytmy wyznaczające dopasowanie globalne i lokalne są algorytmami dynamicznymi, jako rozgrzewkę przypomnimy sobie algorytm najdłuższego wspólnego podciągu (LCS - *longest common subsequence*). Pamiętajmy, że ważną cechą problemów do których możemy używać algorytmów dynamicznych jest własność optymalnej podstruktury tzn. jego optymalne rozwiązanie jest funkcją optymalnych rozwiązań podproblemów.

Zadanie 2.
Jaki jest sens algorytmu LCS dla sekwencji DNA i białkowych - jakiego typu mutacji nie bierze on pod uwagę?

Algorytm:
Mamy dwa ciągi *V* i *W* długości odpowiednio *n* i *m*. Konstruujemy tablicę *H* rozmiaru *n* * *m*, gdzie pozycja *H[j][j]* oznacza LCS dla odpowiednio *V[1..i]* i *W[1..j]*. Ostatecznie w polu *H[n][m]* otrzymamy LCS dla całego *V* i *W*. W danym kroku wyliczamy *H[j][j]* na podstawie wcześniejszych obliczonych LCS dla podproblemów :
- jeśli $V[i] = W[j]$ wtedy dołączamy zgodny znak do wyniku:

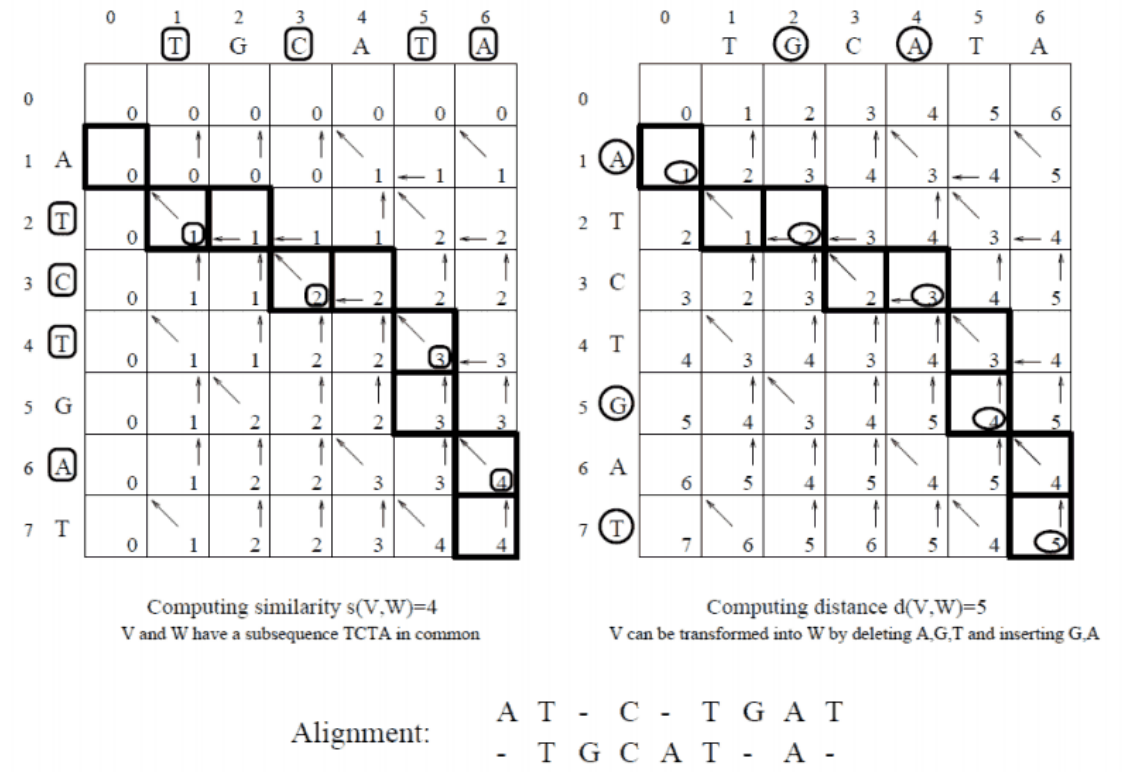
$$H[i][j] = H[i-1][j-1] + 1$$

- jeśli $V[i] \neq W[j]$ wtedy sprawdzamy, który LCS dla podproblemów odpowiednio (*V[1..i]; W[1..j-1]*) i (*V [1.i-1];W[1..j]*) był dłuższy:

$$H[i][j] = max\{ H[i-1][j]; H[i][j-1] \}$$

Dodatkowo możemy określić ilość zmian - insercji, delecji potrzebnych , aby przeprowadzić *V* w *W*, (*H(V;W)* oznacza długość *LCS(V;W)*):

$$d(V;W) = n + m - 2H(V;W)$$



Rysunek 3: Rysunek przebiegu LCS dla dwóch przykładowych sekwencji oraz powstałe dopasowanie (przykład z książki „An Introduction to Bioinformatics Algorithms", Jones, Pevzner, MIT Press 2004).

5. Algorytm Needlemana-Wunscha (dopasowanie globalne)

Algorytm LCS jest w pewnym sensie użyteczny dla zagadnienia dopasowania sekwencji, natomiast bierze on pod uwagę tylko możliwość indeli, czyli insercji lub delecji nukleotydów lub aminokwasów, przy czym dopasowanie nie jest w żadnej sposób za nie „karane". Można pomyśleć nad takim wariantem algorytmu, który punktowałby również podstawienia, czyli zamianę nukleotydu czy aminokwasu na inny oraz w pewien sposób obniżał punktację dopasowania, jeśli nastąpi indel, ponieważ pojawienie się mutacji punktowej - zamiany nukleotydu czy aminokwasu na inny jest w naturze dużo bardziej prawdopodobne niż indel.

Tego typu algorytm zaproponowali panowie Needleman i Wunsch w 1970 roku.

$$H[i][j] = \max \begin{cases} H[i-1][j-1] + s(V[i], W[j]), & gdy\ dopasowanie\ lub\ substytucja \\ H[i-1][j] + g, & gdy\ przerwa\ wV \\ H[i][j-1] + g, & gdy\ przerwa\ wW \end{cases}$$

gdzie macierz H jest macierzą punktacji, g jest liniową karą za przerwę. Dla nukleotydów macierz punktacji to np. +1 za dopasowanie, -2 za niedopasowanie, natomiast dla aminokwasów sytuacja jest bardziej skomplikowana i wykorzystuje się macierze substytucji.

Możliwe są też inne typy kary za przerwę, np. najczęściej używany jest afiniczny model kary za przerwę:

$$Gap_{penalty} = Gap_{opening} + k * Gap_{extension}$$

Wiemy, że otwarcie przerwy, czyli możliwość insercji lub delecji jest dość rzadka, natomiast jeżeli już przerwa się pojawi, to jest duże prawdopodobieństwo, że wstawi się lub zniknie wiele nukleotydów na raz. Dlatego lepiej jest gdy mamy mniej przerw, ale są one dłuższe, niż więcej krótkich. Stąd nakładamy dużą karę za otwarcie przerwy, które jest mało prawdopodobne, a potem dokładamy dużo mniejszą karę za wydłużenie przerwy.

Zadanie 3: (4 pkt)

Rozwiązanie zadanie prześlij mailem do niedzieli, **5.11.2019** włącznie, na adres:
jacek.smietanski@ii.uj.edu.pl
Temat wiadomości proszę opatrzyć przedrostkiem **[Bio] Lab 04**. Rozwiązaniem ma być **tylko jeden plik** – skrypt zgodny z Pythonem w wersji 3.x**,** zawierający wszystkie niezbędne funkcje oraz procedurę wykonawczą. Proszę o nazwanie pliku wg schematu**: Imie.Nazwisko.04.py**.

Stwórz własną implementację algorytmu Needlemana-Wunscha dopasowania globalnego par sekwencji w modelu liniowym (w tym zadaniu zabronione jest korzystanie z implementacji dostępnej w biopythonie i innych zewnętrznych bibliotekach).

Dla uproszczenia można przyjąć, że rozważane będą wyłącznie sekwencje aminokwasowe (wykorzystaj macierz substytucji, np. BLOSUM62), kara za przerwę: -7 pkt.

Gdy istnieje więcej niż jedno optymalne dopasowanie, program powinien wypisać wszystkie.

Przetestuj swój algorytm porównując sekwencje hemoglobiny beta u człowieka (GI:40886941) i szczura (GI:34849618) z bazy Protein (użyj Bio.Entrez).

Możesz wykorzystać dostępne w biopythonie implementacje macierzy substytucji:

```
from Bio.SubsMat import MatrixInfo as mi
mi.blosum62
```

Macierz możesz zaimplementować jako listę list (przykład tworzenia niżej) albo (lepiej) skorzystać z zewnętrznej biblioteki numerycznej NumPy.

Przykład: utworzenie macierzy 2 x 3

```
n, m = 2, 3
H = [[None] * m for i in range(n)]
```

I z wykorzystaniem NumPy:

```
import numpy as np
H = np.zeros((n, m))
```