

Metody bioinformatyki (MBI)

Wykład 1 - Bioinformatyka jako dziedzina informatyki.
Badanie podobieństw sekwencji.

Robert Nowak

2026L

Cel i zakres przedmiotu

umiejętność rozwiązywania problemów pojawiających się przy przetwarzaniu danych biologicznych i medycznych

- ▶ omówienie problemów biologicznych i medycznych, które wymagają stosowania zaawansowanych algorytmów
- ▶ omówienie niezbędnych podstaw biologii molekularnej
- ▶ demonstracja charakterystycznych algorytmów

Zakładana znajomość:

- ▶ podstawowych algorytmów i struktur danych
- ▶ programowania

Nie jest wymagana znajomość biologii i medycyny

Podstawowe dane o przedmiocie

Wykład: środa 14¹⁵ – 16⁰⁰

Prowadzący:

prof. dr hab. inż. Robert Nowak,
dr hab. inż. Tomasz Gambin, prof. uczelni
dr inż. Wiktor Kuśmirek

Strona przedmiotu:

<http://staff.elka.pw.edu.pl/~rnowak2/dyd/mbi>

Konsultacje:

patrz repo.pw.edu.pl

Tematyka wykładów

- T. Gambin** – Opis od strony medycyny, algorytmy stosowane w genetyce medycznej.
- W. Kuśmirek** – Sekwencjonowanie 3 generacji.
- R.Nowak** – Algorytmika, programowanie, inżynieria genetyczna.

Literatura do części informatycznej

- ▶ Jin Xiong, Podstawy bioinformatyki, PWN, 2011
- ▶ P.Higgs, T.Attwood, Bioinformatyka i ewolucja molekularna, PWN, 2008
- ▶ V. Makinen, D. Belazzougui, F. Cunial, A. Tomescu, Genome-Scale Algorithm design, Cambridge 2015
- ▶ Wing-Kin Sung, Algorithms for next-generation sequencing, CRC Press 2017
- ▶ R.Durbin, S.Eddy, A.Krogh, G.Mithison, Biological sequence analysis. Cambridge 2007
- ▶ Jeremy Ramsden, Bioinformatics, an introduction. Springer 2023

Tematyka wykładów - część informatyczna

- ▶ bioinformatyka jako dziedzina informatyki;
- ▶ programowanie dynamiczne, uliniowanie dwu sekwencji;
- ▶ tworzenie macierzy podobieństwa;
- ▶ bazy danych sekwencji biologicznych, algorytm BLAST;
- ▶ badanie podobieństw wielu sekwencji;
- ▶ tworzenie drzew filogenetycznych;
- ▶ wyszukiwanie motywów w sekwencjach;
- ▶ assembly DNA, algorytmy oparte o graf de Brujna;
- ▶ ukryte modele Markowa dla sekwencji;
- ▶ markery genetyczne, badanie pokrewieństw;
- ▶ badanie rozkładu haplotypów, algorytm EM;
- ▶ grupowanie i redukcja wymiarów, algorytm PCA;
- ▶ biologia syntetyczna, obliczanie struktur drugorzędowych.

Zaliczenie przedmiotu

- ▶ egzamin 0 – 20pkt
- ▶ 4 ćwiczenia 0 – 5 pkt każde
 - ▶ można je wykonać samodzielnie lub pod opieką prowadzącego
- albo projekt 0 – 20pkt
 - ▶ projekt i implementacja oprogramowania

37 – 40 pkt.	ocena 5
33 – 36 pkt.	ocena $4\frac{1}{2}$
29 – 32 pkt.	ocena 4
25 – 28 pkt.	ocena $3\frac{1}{2}$
21 – 24 pkt.	ocena 3
0 – 40 pkt.	ocena 2

Nie jest wymagana obecność na wykładzie i nie będzie ona sprawdzana.

4 ćwiczenia

- ▶ zespoły dwuosobowe
- ▶ instrukcje instalacji, pakiety dla Ubuntu (docker)
- ▶ ćwiczenie jest oceniane w skali 0-5 pkt

utworzenie zespołów	od 15 do 31 marca
cw 1, assembling de novo	08.04.2026, 18-20, online
cw 1, dokumentacja	do 15.04.2026
cw 2, adnotacja DNA	22.04.2026, 18-20, online
cw 2, dokumentacja	do 29.04.2026
cw 3, resekwencjonowanie DNA	06.05.2026, 18-20, online
cw 3, dokumentacja	do 13.05.2026
cw 4, analiza genomu człowieka	20.05.2026, 18-20, online
cw 4, dokumentacja	do 27.05.2026

Projekt

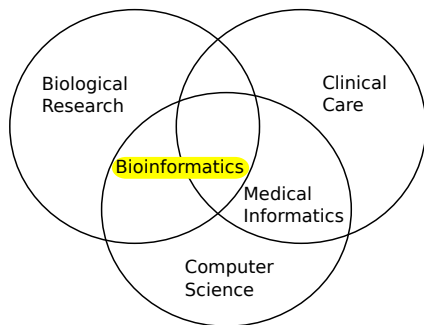
utworzenie aplikacji lub biblioteki

- ▶ zespoły dwuosobowe,
- ▶ zgłoszenie zespołu do 15 marca (i wysłanie tematu),
- ▶ uzyskanie akceptacji tematu projektu do 31 marca,
- ▶ działający prototyp, do 30 kwietnia,
- ▶ aplikacja i dokumentacja, do 10 czerwca.

Miejsce bioinformatyki wśród innych dziedzin wiedzy

Dziedzina na styku:

- ▶ matematyki
- ▶ informatyki
- ▶ statystyki
- ▶ biologii molekularnej
- ▶ medycyny

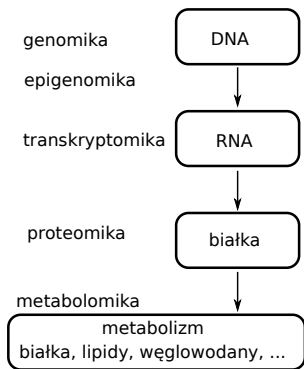


Główny cel: **zrozumieć procesy biologiczne**

Bioinformatyka - ważne daty

data	wydarzenie
1970	pierwszy raz użyto słowa 'bioinformatyka'
1970	algorytm uliniowania 2 sekwencji
1971	PDB, Protein Data Bank
1978	algorytm PAM znajdowania macierzy podobieństwa
1982	GenBank
1990	algorytm BLAST
1997	algorytm PSI-BLAST
2003	sekwencja genomu człowieka
2010	pierwszy genom syntetyczny
2012	system CRISPR-Cas9, edycja DNA
2020	AlphaFold2 poprawnie przewiduje str. przestrzenną białka

Bioinformatyka - rodzaj analizowanych danych



- ▶ genomika - analiza genomu:
 - ▶ genomika strukturalna,
 - ▶ genomika funkcjonalna,
 - ▶ genomika porównawcza;
- ▶ epigenomika – analiza modyfikacji chromatyny;
- ▶ transkryptomika – analiza transkryptomów;
- ▶ proteomika – analiza białek;
- ▶ metabolomika – analiza przemian chemicznych zachodzących w organizmach;
- ▶ **metagenomika** – analiza materiału izolowanego z nisz ekologicznych.

Bioinformatyka - rodzaj narzędzi

- ▶ zarządzanie danymi (data management), np. bazy danych, ontologie;
- ▶ obliczenia (data computation), np. algorytmy do pozyskiwania danych, asemblacja;
- ▶ odkrywanie wiedzy z danych (data-mining, biological discovers);
- ▶ modelowanie i symulacje (modeling/simulation).

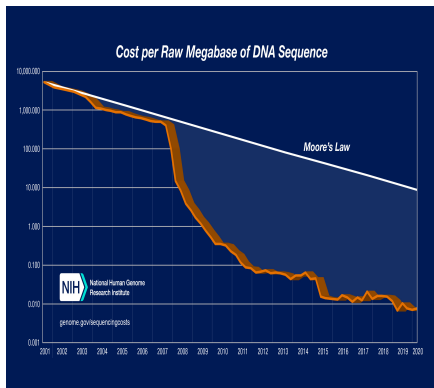
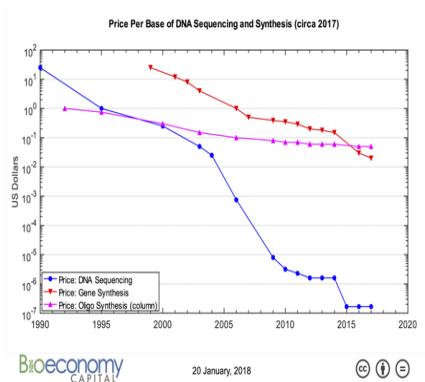
Dlaczego warto coś wiedzieć o bioinformatyce?

- ▶ wielkie nakłady na rozwój tej dziedziny
- ▶ potencjalnie duży obszar zastosowań (medycyna, farmacja, sądownictwo, ubezpieczenia)
- ▶ potrzeba wykorzystywania wyrafinowanych algorytmów

Specyfika aplikacji bioinformatycznych

- ▶ duże ilości danych
- ▶ złożone algorytmy
- ▶ wydajność jest niezwykle istotna
- ▶ szerokie wykorzystanie super-komputerów oraz obliczeń rozproszonych (np. grid-computing)

Dlaczego bioinformatyka, np. sekwencjonowanie



- ▶ 10^5 Genomics England, 1+ Million Genomes in EU (1+MG),
 $\sim 10^6$ All of Us in America, $\sim 10^5$ Genome Asia
- ▶ jeden genom to plik 15GB

Badanie podobieństw sekwencji

Oznaczenia

Σ - alfabet, $\Sigma_{DNA} = \{A, C, G, T\}$

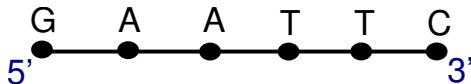
s - sekwencja nukleotydów $s = s_1 s_2 \dots s_n$, gdzie

n - długość sekwencji s , oznaczana $|s|$,

s_i - symbol (o indeksie i) sekwencji s , $s_i \in \Sigma$

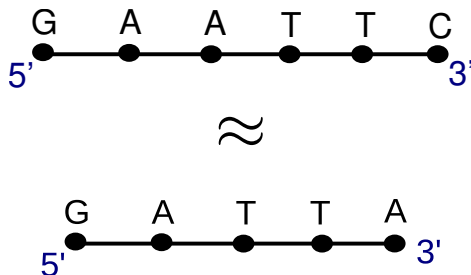
Ciągi pisane od 5' końca.

Przykład:



Rysunek: Sekwencja DNA, długość $|s| = 6$

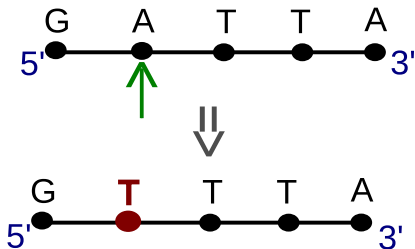
Podobieństwo sekwencji



Operacje zmieniające sekwencje:

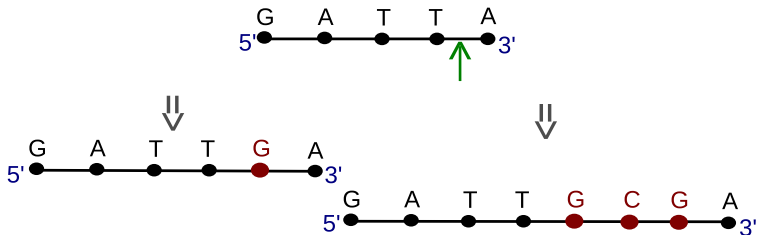
- ▶ mutacje
- ▶ wstawienia (insercje): pojedyncze symbole, łańcuchy
- ▶ usunięcia (delecje): pojedyncze symbole, łańcuchy

Mutacje (mutation)

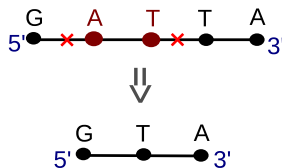
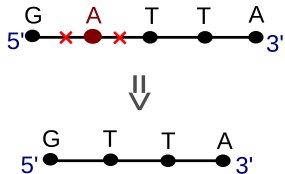


- ▶ prawdopodobieństwo mutacji $A \rightarrow T$
- ▶ prawdopodobieństwo mutacji $A \rightarrow C$
- ▶ ...

Wstawienia (insercje), usunięcia (delecje)

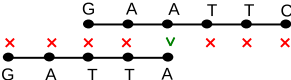
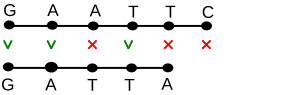
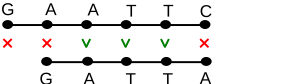


- prawdopodobieństwo wstawienia lub usunięcia sekwencji w funkcji długości tej sekwencji



Znajdowanie najdłuższego wspólnego podciągu (wyszukiwanie wzorca)

- ▶ nie uwzględnia wstawiania i usuwania nukleotydów
- ▶ nagroda za zgodność = 1
- ▶ kara za niezgodność = -1

offset = -2		score = -6
offset = 0		score = 0
offset = 1		score = 0

Macierz podobieństwa

	A	G	C	T
A	1	-1	-1	-1
G	-1	1	-1	-1
C	-1	-1	1	-1
T	-1	-1	-1	1

$d = -1$

Przykładowe wyniki:

	A	G	C	T
A	10	-1	-3	-4
G	-1	7	-5	-3
C	-3	-5	9	0
T	-4	-3	0	8

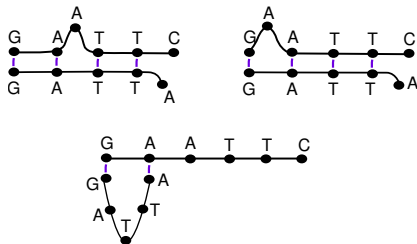
$d = -5$

offset -2	--GAATTC GATTA---	$S_1 = -6$	$S_2 = -22$
offset 0	GAATTC GATTA-	$S_1 = 0$	$S_2 = 12$
offset 1	GAATTC -GATTA	$S_1 = 0$	$S_2 = 17$

Uwzględnienie przerw - wszystkie możliwe połączenia

Liczba możliwych połączeń łańcuchów s i t (o długości n)

$$C(s, t) = \binom{|s| + |t|}{|s|} \simeq \frac{(2n)!}{(n!)^2} \simeq \frac{2^{2n}}{\sqrt{n}} \text{ gdy } |s| \simeq |t| \simeq n$$



n	liczba połączeń
10	$3 * 10^5$
20	$2.5 * 10^{11}$
30	$2 * 10^{17}$
50	$1.8 * 10^{29}$
100	$1.6 * 10^{59}$

Rozwiązanie problemu

Miara dopasowania jest addytywna, więc można konstruować optymalne rozwiązanie na podstawie wyników wcześniejszych obliczeń.

czyli

rekurencyjnie zdefiniować problem

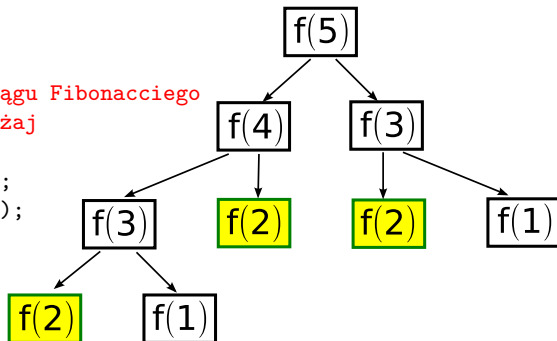
a następnie go rozwiązać

- ▶ metodą dziel i zwyciężaj
- ▶ metodą programowania dynamicznego

Metoda dziel i zwyciężaj : ciąg Fibonacciego

//obliczanie wyrazów ciągu Fibonacciego
//metodą dziel i zwyciężaj

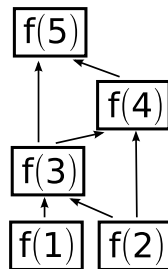
```
int f(int n) {  
    if(n <= 2) return 1;  
    return f(n-1)+f(n-2);  
}
```



Wielokrotnie wykonuje te same obliczenia

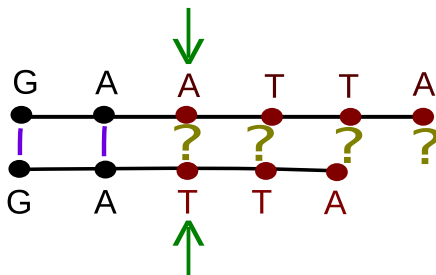
Metoda programowania dynamicznego : ciąg Fibonacciego

```
//obliczanie wyrazów ciągu Fibonacciego  
// metodą programowania dynamicznego  
int f(int n) {  
    boost::scoped_array<int> tab = new int[n];  
    tab[0] = 1;  
    tab[1] = 1;  
    for(int i=2; i<n;++i)  
        tab[i] = tab[i-1] + tab[i-2];  
    return tab[n-1];  
};
```



**oblicza optymalne rozwiązanie metodą wstępującą
(bottom-up)**

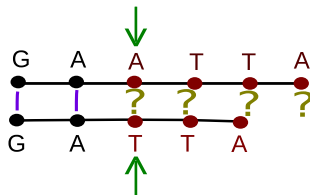
Rekurencyjna definicja problemu



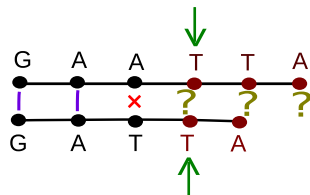
Mając częściową strukturę możemy:

- ▶ dodać parę
- ▶ dodać przerwę na jednej nici
- ▶ dodać przerwę na drugiej nici

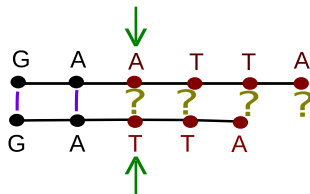
Definicja - dodawanie pary



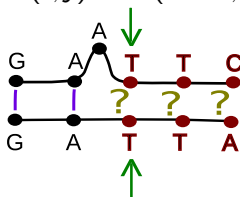
$$F(i, j) = F(i-1, j-1) + e(s_i, t_j), \begin{cases} e \text{ zawiera nagrody za zgodność} \\ e \text{ zawiera kary za niezgodność} \end{cases}$$



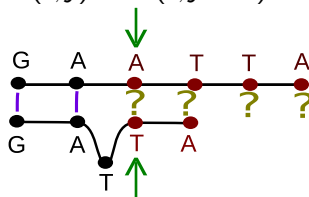
Definicja - dodawanie przerwy



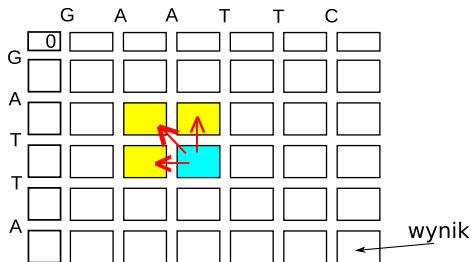
$$F(i, j) = F(i - 1, j) + d$$



$$F(i, j) = F(i, j - 1) + d$$



Rekurencyjna definicja problemu



Algorytm Needlemana-Wunscha

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + e(s_i, t_j) // \text{połączenie} \\ F(i-1, j) + d // \text{przerwa na łańcuchu s} \\ F(i, j-1) + d // \text{przerwa na łańcuchu t} \end{cases}$$

Algorytm Needlemana-Wunscha

- ▶ znajduje $\max M(s, t)$
- ▶ złożoność $O(n^2)$

Założenia:

- ▶ znana macierz podobieństwa E
- ▶ kara za „przerwę” $\gamma(g) = |g| * d$
- ▶ wykorzystuje obliczenia dla krótszych łańcuchów

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + e(s_i, t_j) & // \text{połączenie} \\ F(i-1, j) + d & // \text{przerwa na łańcuchu } s \\ F(i, j-1) + d & // \text{przerwa na łańcuchu } t \end{cases}$$

Algorytm Needlemana-Wunscha (przykład)

Przykładowa macierz kar i nagród

	A	G	C	T
A	10	-1	-3	-4
G	-1	7	-5	-3
C	-3	-5	9	0
T	-4	-3	0	8

Kara za przerwę:

$$d = -5$$

Kara za wiszący nukleotyd:

$$d = -5$$

Algorytm Needlemana-Wunscha (przykład)

	G	A	A	T	T	C	
G	0	-5	-10	-15	-20	-25	-30
A	-5						
A	-10						
T	-15						
T	-20						
A	-25						

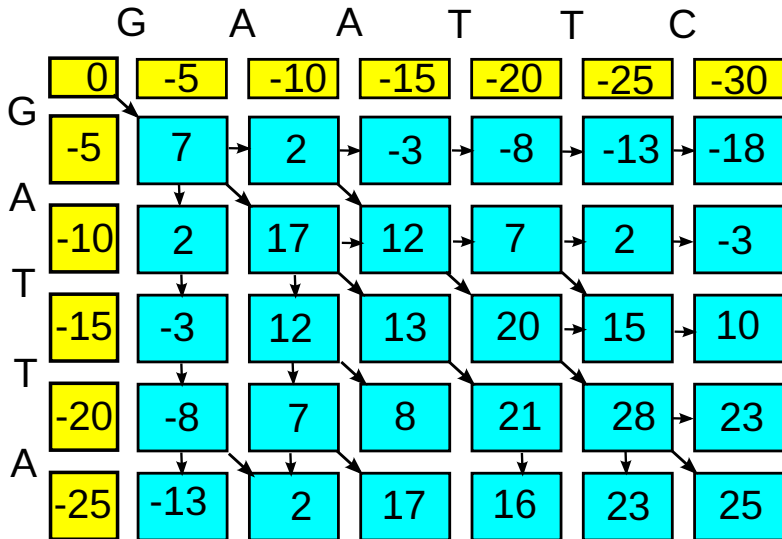
Algorytm Needlemana-Wunscha (przykład)

	G	A	A	T	T	C	
G	0	-5	-10	-15	-20	-25	-30
A	-5	7					
A	-10						
T	-15						
T	-20						
A	-25						

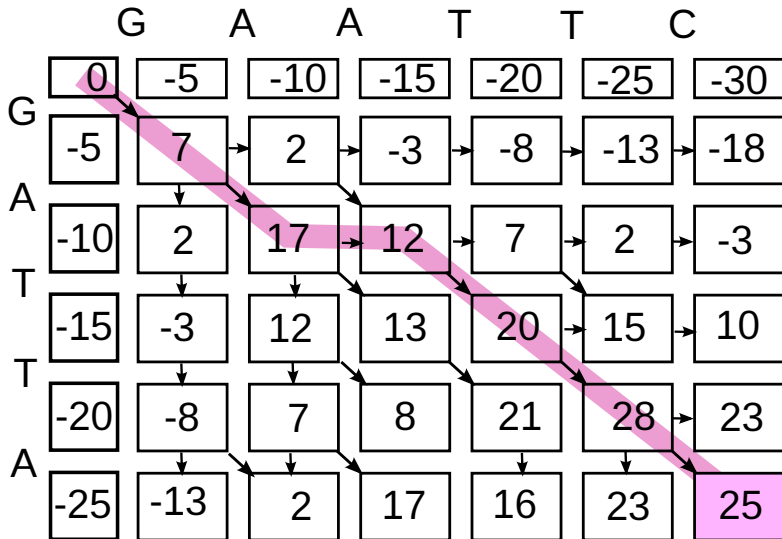
Algorytm Needlemana-Wunscha (przykład)

	G	A	A	T	T	C	
G	0	-5	-10	-15	-20	-25	-30
A	-5	7	2	-3	-8	-13	-18
T	-10						
T	-15						
A	-20						
A	-25						

Algorytm Needlemana-Wunscha (przykład)



Algorytm Needlemana-Wunscha (przykład)



Algorytm Needlemana-Wunscha (przykład)

2 rozwiązania:

G	A	A	T	T	C		G	A	A	T	T	C
						,						.
G	A	-	T	T	A		G	-	A	T	T	A

Parametry:

- ▶ złożoność czasowa: $O(m*n)$
- ▶ złożoność pamięciowa: $O(m*n)$

Na stronie przedmiotu dostępne są programy demonstracyjne pokazujące kolejne kroki obliczeń.

Dziękuję